EcoStruxure Control Expert

Operating Modes

Original instructions

33003101.27
12/2023

www.se.com

Legal Information

The information provided in this document contains general descriptions, technical characteristics and/or recommendations related to products/solutions.

This document is not intended as a substitute for a detailed study or operational and site-specific development or schematic plan. It is not to be used for determining suitability or reliability of the products/solutions for specific user applications. It is the duty of any such user to perform or have any professional expert of its choice (integrator, specifier or the like) perform the appropriate and comprehensive risk analysis, evaluation and testing of the products/solutions with respect to the relevant specific application or use thereof.

The Schneider Electric brand and any trademarks of Schneider Electric SE and its subsidiaries referred to in this document are the property of Schneider Electric SE or its subsidiaries. All other brands may be trademarks of their respective owner.

This document and its content are protected under applicable copyright laws and provided for informative use only. No part of this document may be reproduced or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), for any purpose, without the prior written permission of Schneider Electric.

Schneider Electric does not grant any right or license for commercial use of the document or its content, except for a non-exclusive and personal license to consult it on an "as is" basis.

Schneider Electric reserves the right to make changes or updates with respect to or in the content of this document or the format thereof, at any time without notice.

To the extent permitted by applicable law, no responsibility or liability is assumed by Schneider Electric and its subsidiaries for any errors or omissions in the informational content of this document, as well as any non-intended use or misuse of the content thereof.

Table of Contents

Safety Information 31
Before You Begin 32
Start-up and Test 33
Operation and Adjustments 34
About the Book 35
General 45
General Presentation of Control Expert 46
Capabilities of Control Expert 46
User Interface 51
Programming Language Editors 55
Function Block Diagram FBD 55
Ladder Diagram (LD) Language 58
General Information about SFC Sequence Language 60
Instruction List IL 64
Structured Text ST 65
Connecting a terminal to a PLC 68
Methodology for connecting from a PC to a PLC 68
Global project management 71
Access security management 71
Access security management 71
Project management 72
Access to Control Expert Software 72
Services in Offline Mode 74
Services in Online Mode 83
Allowed Online Modifications 85
Connection and Disconnection 91
Access a Controller Over a Network 93
Overview of Types of Connection Possible With Schneider Controllers 98
Access Communication Parameters 102
Project comparison 104
Conversion of Legacy Applications/Projects 106

Importing an LL984 Application 107
Transfer the Project Between EcoStruxure Control Expert and the Controller 117
Update of Upload Information 120
Sending a command to the PLC 122
Project Backup Management for Modicon M340 122
Project backup for Premium 125
Save/Restore Data Between a File and the Controller 127
Compatibility Rules for Restore Using a *.DTX File 132
Transferring Current Values 139
Memory usage 139
Data Memory Protect 146
Function Pack for Memory 151
Memory Card Access for Modicon M340 153
FTP and Memory Card for Modicon M340 155
Project Management with DTMs 156
Project browser 161
Introduction to the project browser 161
Introduction to the project browser 161
Structural view and functional view of the project 164
The Zoom In and Zoom Out commands 166
The commands Contract all and Expand all 168
The Goto command 173
User Directory 176
Project Properties 177
Application Protection 179
Safety-Related Area Password Protection 186
Firmware Protection 190
Program Unit, Section and Subroutine Protection 192
Data Storage/Web Protection 195
Loss of Password 197
Introduction to the various directories of the project browser 203
Project Directory 203
Configuration Directory 204
Derived data types (DDT) directory 206

Derived FB (DFB) types directory 207
Variables Directory 208
Motion Directory 210
Communication Directory 212
Ethernet Network Directory 213
Programs Directory 213
Animation Tables Directory 217
Operator Screens Directory 220
Documentation Directory 222
Conversion Report Directory 223
Summaries of the utilities associated with the structure view and function view 225
Summary of the services associated with the structural view 225
Functional modules 237
Introduction to the functional view 237
Functional modules 237
Project directory of the functional view 239
Functional Module Directory 240
Functional Module Program Directory 241
Functional Module Animation Tables Directory 242
Functional Module Operator Screens Directory 243
Summary of Services for the Functional Views 244
The functional modules and their associated utilities 251
Properties of a functional module 251
Creating a functional module 252
Programming a functional module 253
Protecting program elements of the functional module 255
Debugging a functional module 255
Detach or Delete a Functional Module 256
Functional module export 259
Functional module import 259
Creating, Deleting, Locating, Dragging-and-Dropping an Animation Table in a Functional Module 259
Creating, Deleting, Locating and Dragging-and-Dropping an Operator Screen in a Functional Module 262

FDT Container 267
FDT Container 267
Introduction to the Control Expert FDT Container 267
FDT Container User Rights 268
DTM Hardware Catalog 270
Introduction to the Hardware Catalog 270
Description of DTM Hardware Catalog 271
Adding a DTM to the Control Expert Hardware Catalog 275
Adding an EDS File to the Hardware Catalog 276
Removing an EDS File from the Hardware Catalog 278
Export / Import EDS Library 280
DTM Browser 283
DTM Browser 283
Ethernet Ready Equipment 288
Ready, Built, Connected and Installed DTMs 289
DTM Browser Contextual Menus 293
Field Bus Discovery Service 301
Store Device Configurations 305
Renaming DTMs 306
Profibus DTMs 308
DTM Properties Tabs 309
DTM Properties Dialog 309
Type Library Manager 314
Libset (Set of Libraries) 314
Overview of the Type Library Manager 317
Type Library Manager 318
Description of the Type Library Manager 321
Loading an object from the library into the project 327
Transferring an object from the project into the library 330
Creating a new library and family 332
Creating an installable family 334
Updating a family of a library 336
Consistency Check Assistant 338
Library Version Management 341
Using the type library of an older Unity Pro/Control Expert Version 345

Data Editor 348
Description of the Data Editor 349
Data Editor Access 349
Description of Data Editor tabs 351
Creating derived data types (DDT) 356
Access to derived data types (DDT) 356
Creation of a derived data type (DDT) 357
Administration of derived data types and their elements 368
Creating Nested Derived Data Types 370
Save Derived Data Types (DDT) 375
Creating data types of user function blocks (DFB) 378
Creation and Management of Data Types and Function Blocks (DFB) 378
Managing of instances of data belonging to the family of function blocks (EF) 380
Creating / Deleting a variable instance of function blocks (EFB/DFB) 380
Displaying function block variable instances in the editor 383
Changing the attribute for variable instances in a function block 386
Managing variable instances belonging to the EDT/DDT/IODDT/Device DDT families 390
Creation of EDT instances 390
Creation of DDT instances 394
Creating an IODDT instance 400
Managing a Device DDT instance 403
Managing an IODDT, DDT or an Array type instance 407
Creation of multiple variable instances 413
Displaying the variable instances in the editor 415
Changing attributes of variable instances 418
Creation of a Reference Type Instance 422
General Functions 425
Configure the Data Editor Columns 425
Description of the Data Editor: Variable type selection 429
Filtering of Data 433
Exporting a subset of variables 437
Analyze DDT and DFB Data Types 438
Purge of Unused Instances 439
Local printing of data 441

Save the context of the Data Editor 442
Communication 444
Presentation of the communication editors 445
Presentation of the communication editors of the project browser 445
Network Configuration 448
Network Configuration Principle Using Control Expert 448
Creating a Logic Network 449
Configuring a Logic Network 451
Associating a Logic Network with Network Hardware 452
Configuration of X-Way Routing Premium Stations 456
Configuration 457
Configuration of Multi-Network Services 458
Configuring an X-Way Router Module 459
Examples of X-Way Routing Stations 463
Examples of Partial Routing 466
Programming 471
Accessing Tasks, Program Units and Sections 472
Access to Tasks 472
Create and Configuration of a Task 472
Displaying and Modifying Task Properties 474
Accessing Program Units 475
Creating Program Units 475
Properties Dialog Box for Program Units 477
Program Unit Data Scope 482
Program Unit Data Editor Access 485
Description of Program Unit Data Editor Tabs 486
Program Unit Variables 491
Program Unit Management 499
Accessing Sections 502
Creating an FBD, LD, IL, or ST Section 502
Properties Dialog Box for FBD, LD, IL, or ST Sections 505
Creating SFC sections 510
Properties Dialog Box for SFC Sections 513
Creating action, transition and macro sections 518
Action, transition and macro sections properties dialog box 519

Create a Sub-Program Section (SR) 522
Property Dialog of the Sub-Program Sections 524
Creating a Timer Event Section 526
Properties dialog box for Timer Event Sections 529
Create an I/O Event Section 532
Property Dialog of I/O Event Sections 534
Opening a Section 537
Deleting sections 538
Execution order of sections 538
Creating LL984 Segments and Networks 539
Modifying LL984 Segment Properties 543
Modifying LL984 Network Properties 544
Scheduling LL984 Segments and Network 546
Deleting, Cutting, Copying and Pasting LL984 Segments and Network 547
Control Expert Software Options 549
Project Settings 549
General Description 549
General Project Settings 551
Variables 569
Program 573
Maintain Output Links on Disabled EF (EN=0) 581
Configuration Settings for Operator Screens 591
Options 595
General Description 595
General Options 596
Data and Languages 598
Connection Options 600
The Operator Screens Option 602
Converters 604
Customize Dialog Box 610
Customize Dialog Box 610
Common Function for graphical language 615
Using Bookmarks in Graphical Languages Sections 615
Display grid in a FBD/LD/SFC section 621

Zoom in a FBD/LD/SFC section 621
Tracking links in a FBD/LD/SFC section 624
Print Current Section in a FBD/LD/SFC section 625
Undo and redo changes in a FBD/LD section 626
Using connectors in a FBD/LD section 627
Inspect windows in a FBD/LD section 632
Tooltips for Variables in a FBD/LD/SFC Section 636
Replace Variables in a FBD/LD Section 638
Initialize search in a FBD/LD section 643
Initialize Animation Table in a FBD/LD section 644
Go to in a FBD/LD section 645
Go to dialog box in a FBD/LD section 647
FBD Editor 651
Creating a program with FBD programming language 651
Structure of an FBD program (function block language) 651
Creating an FBD Program 653
Syntax and Semantics Check During Programming 654
Navigating with the keyboard 655
Selecting objects 656
Deleting, cutting, copying, pasting, duplicating, and moving objects 658
Copying, pasting and adjusting pin variables 663
Inserting Rows and Columns 664
Deleting Rows and Columns 665
Displaying the properties 666
Refining DFBs and subroutines 667
Editing FFBs 669
General information about calling an FFB 669
Calling an FFB via the FFB Input Assistant. 671
Calling an FFB via the data selection 674
Assign actual parameters 679
Using Public Variables 691
Expanding Functions 692
Negating FFB Pins 693
EN and ENO Show and Hide 694
Modifying the execution order 695

Replacing an FFB 699
Properties dialog box for elementary functions, procedures and sub-routine blocks. 701
Properties dialog box for elementary and derived function blocks (FBs) 703
Returning from a Subroutine or DFB 706
Returning from a Subroutine or DFB 706
Return object properties dialog box 708
Calling a Subroutine 708
Calling a subroutine 709
Jumps within the Current Section 713
Jump within the current section 713
Jump object properties dialog box 716
Definition of the Jump Target (Jump Labels) 718
Definitions of Jump Labels 718
Jump label objects properties dialog box 721
Editing Links 723
Links 723
Placing a Link 724
Editing Links 729
Entering Comments 737
Entering comments 737
Online functions 739
Online functions 739
FBD Reference Data Type 739
Reference Data Type in FBD 739
Export/Import 740
Export/Import FBD Sections 741
Customize FBD Editor 741
Customize FBD Editor Colors 741
FBD Favorite Bar 743
LD Editor 745
Creating a program with LD programming language 745
Structure of an LD program (ladder diagram) 746
Creating an LD program 748

Syntax and Semantics Check during programming ... 752
View for variables ... 753
Navigating with the Keyboard ... 758
Selecting objects ... 760
Deleting, cutting, copying, pasting, duplicating, and moving objects ... 763
Copying, and pasting variables ... 767
Insert Mode ... 768
Using Mnemonics ... 769
Displaying the properties ... 770
Refining DFBs and subroutines ... 771
Inserting, Deleting, and Copying Rows ... 773
Columns Management ... 774
Editing Contacts ... 775
Select contacts ... 776
Placing contacts ... 777
Property dialog box for contacts ... 781
Editing Coils ... 783
Selection of coils ... 784
Placing coils ... 786
Property dialog box for coils ... 791
Editing compare blocks ... 794
Placing compare blocks ... 795
Compare block properties dialog box ... 798
Editing operation blocks ... 800
Placing Operate Blocks ... 800
Operate block properties dialog box ... 804
Editing FFBs ... 806
General information on calling an FFB ... 807
Calling an FFB via the FFB input assistant ... 808
Calling an FFB via the data selection ... 812
Connection of FFBs to the Left Power Rail ... 816
Assigning Actual Parameters ... 817
Using Public Variables ... 830
Expanding Functions ... 832
Negating FFB Pins ... 834

EN and ENO Show and Hide 835
Properties dialog box for elementary functions, procedures and sub-routine blocks. 835
Properties dialog box for elementary and derived function blocks (FBs) 838
Returning from a Subroutine or DFB 840
Returning from a subroutine or DFB 840
Return object properties dialog box 842
Jumps within the Current Section 843
Jumps within the current section 843
Jump object properties dialog box 846
Definition of the Jump Target (Jump Labels) 848
Definitions of jump targets (jump labels) 848
Jump label properties dialog box 851
Editing Links 853
Select links 853
Combining Links 855
Placing links 857
Editing links 872
Entering Comments 880
Entering comments 880
Online functions 882
Online Functions 882
LD Reference Data Type 882
Reference Data Type in LD 882
Export/Import 883
Export/Import LD Sections 884
Customize LD Editor 884
Customize LD Editor Colors 884
LD Instruction Bar 886
LD Favorite Bar 887
SFC Editor 890
Creating a program with the SFC programming language 890
Structure of an SFC Program (Sequential Function Chart) 891
Creating an SFC Program 893

Syntax and Semantics Check during Programming 894
Navigating with the keyboard 895
Selecting Objects 897
Undo and Redo Changes 899
Delete, Cut, Copy, Paste, and Move Objects 900
Inserting Rows and Columns 905
Deleting Rows and Columns 907
Displaying the object properties 908
Initialize Search 909
Initialize Animation Table 909
Refining steps, macro steps and transitions 911
Go to 912
Go to dialog 914
List and display jumps 918
Go to Step 919
General Information about Steps 920
Selecting steps 920
Placing steps 922
Editing "normal" steps 923
Defining the properties of steps 924
Steps properties dialog box 931
Editing macro steps 936
Defining the properties of macro steps 937
Macro Steps Properties Dialog Box 944
Editing input steps 947
Defining the properties of InSteps 947
InSteps properties dialog box 949
Editing Output Steps 953
Defining the properties of OutSteps 954
OutStep properties dialog box 955
Editing actions 957
Assigning actions to a step 958
Display modes for actions 967
Editing Transitions 968
Placing transitions 968

Defining transition properties 970
Properties dialog box for transitions 975
Editing Jumps 977
Placing jumps 978
Defining jump properties 979
Jump properties dialog box 981
Editing Branches and Joints 983
Selecting Branches and Joints 984
Placing branches and joints 985
Modifying the Properties of Branches and Joints 986
Branches properties dialog box 989
Joints properties dialog box 990
Inserting Alternative Strings and Simultaneous Strings 991
Selecting Alternative Sequences and Parallel Sequences 991
Placing alternative sequences and parallel sequences 992
Inserting Step-Transitions and Transition-Step-Sequences 995
Selecting Step-Transition and Transition-Step Sequences 995
Placing step-transition and transition-step sequences 996
Editing Links 999
Links 1000
Placing a Link 1001
Editing links 1005
Entering Comments 1010
Entering comments 1010
Online functions 1012
Online functions 1012
Export/Import 1012
Export/Import SFC Sections 1012
Customize SFC Editor Colors 1012
Customize SFC Editor Colors 1013
Common Function for Textual Language 1016
Selecting Text 1016
Deleting, cutting, copying, pasting and moving text 1019
Undoing and redoing changes 1021
Match case 1022

Zoom (font size) 1023
Using Bookmarks 1024
Go to 1026
Go to dialog box 1028
Searching text 1030
Searching and replacing text 1032
Call a subroutine 1034
Refine DFBs and subroutines 1035
Inspect windows 1037
Enter comments 1041
Printing the Current Section 1042
Initialize search 1043
Initialize Animation table 1044
Displaying the section properties 1046
Fullscreen 1046
Customize IL and ST Editor Colors 1047
IL Editor 1051
Structure of an IL program 1051
Creating an IL Program 1053
Syntax and Semantics Check during Programming 1054
Navigating with the keyboard 1059
Data input 1060
Calling an FFB 1066
Using Public Variables 1077
Online functions 1079
Reference Data Type in IL 1079
Export/Import IL Sections 1080
ST Editor 1082
Structure of an ST program 1082
Creating an ST Program 1084
Syntax and Semantics Check during Programming 1085
Navigating with the keyboard 1089
Data input 1090
Calling an FFB 1097
Using public variables 1108

Calling a statement 1110
Online functions 1117
Reference Data Type in ST 1118
Export/Import IL Sections 1118
LL984 Editor 1120
Creating a Program with LL984 Programming Language 1120
Project Settings for LL984 Programs 1120
Structure of an LL984 Program 1121
Creating an LL984 Program 1124
Syntax and Semantics Check During Programming 1126
Network Display 1129
Navigating with the Keyboard 1130
Selecting Objects 1132
Deleting, Cutting, Copying, Pasting and Moving Objects 1134
Displaying the Properties 1137
Inserting and Deleting Rows/Columns 1137
Editing Contacts 1139
Selecting Contacts 1139
Placing Contacts 1141
Properties Dialog for Contacts 1144
Editing Coils 1146
Selecting Coils 1147
Placing Coils 1149
Property Dialog for Coils 1152
Editing Function Blocks 1154
Editing Function Blocks 1154
Using Mnemonics 1157
Editing Links 1158
Selecting Links 1158
Combining Links 1159
Placing Links 1160
Editing Links 1162
Editing Equation Network Blocks 1163
Editing Equation Network Blocks 1163
Editing the ST Assignment Expression 1165

Assigning Variables to LL984 Objects ... 1170
Assigning Variables to LL984 Objects ... 1170
Calling Subroutines ... 1173
Calling Subroutines ... 1173
Entering Comments ... 1175
Entering Comments ... 1175
Customize LL984 Editor Colors ... 1177
Customize LL984 Editor Colors ... 1177
Programming Languages Data Selection Dialog Box ... 1181
Instance selection dialog box ... 1181
FFB Selection Dialog box ... 1186
FFB Type Selection Dialog box ... 1190
Data selection dialog box ... 1192
Function Input Assistant ... 1199
Data Properties ... 1202
Generate variables ... 1204
I/O Objects tab ... 1207
Introduction to the I/O Object Tab ... 1207
I/O Objects Tab for a Processor ... 1209
I/O Objects Tab for a Module ... 1211
Details of Areas in the I/O Objects Tab ... 1213
I/O Objects Tab for a Bus Device ... 1219
How to Use the I/O Objects Tab ... 1219
Memory Tab ... 1225
Memory Tab ... 1225
Type Library Browser ... 1231
Access to the Type Library Browser ... 1231
Description of the Type Library Browser ... 1233
Configuration of Columns of the Type Library Browser ... 1235
Drag & Drop of FFBs in a program section ... 1237
Search/Replace Tool ... 1239
Overview of the Search / Replace tool ... 1239
General Features for Search/Replace ... 1241
Searching for Data ... 1242
Replacing Data ... 1243

Search Toolbar 1244
Data Search Dialog 1246
Cross-References (Search Results) 1247
Cross-References (Replace) 1250
Tracing Variables / Addresses 1252
User's Function Blocks 1254
DFB Type 1254
DFB Type 1254
Creating a DFB Type 1256
Configuration of a DFB Type 1257
Programming a DFB type 1261
Deleting a DFB type element 1263
Archiving a DFB Type 1264
How to Reinforce Protection of a DFB Type 1265
Deleting a DFB Type 1267
How to access the attributes of a DFB type 1268
Authorized DFB Type Modifications 1269
DFB Instance 1273
DFB Instance 1273
Creating a DFB Instance 1274
Creating several DFB Instances 1276
Deleting a DFB instance 1276
Operations on DFB Instances 1276
Modifying the Attributes of a DFB Instance 1279
Authorized DFB Instance Modifications 1283
Searching / Animating DFB Instances 1284
How to Create Nested DFBs 1286
How to Create Nested DFBs 1286
Project Configuration 1287
Introduction to the configuration 1288
Proposed Services 1288
Services offered in offline mode 1293
Services Offered In Online Mode 1294
Access to project configuration 1295
Bus Editors 1301

Rack configuration on local bus 1301
Organization of the Racks 1301
How to configure the racks 1306
Configuration of the power supply modules..... 1308
How to configure the power supply modules 1308
Configuration of the processor..... 1310
Choose and Replace the Controller..... 1311
Configuration of the modules in the PLC station 1314
How to configure the modules in the PLC station 1315
Configuration of devices on the field bus..... 1319
Configuration of the CANopen Slaves for Modicon M340 1319
How to access and create a Fipio field bus..... 1322
Creating and Accessing RIO\DIO Field Buses..... 1326
How to configure devices on the RIO/DIO bus..... 1332
Consumption management 1337
Power consumption budget 1337
Budget on the number of application-specific channels 1341
Bus editors in online mode 1344
Online Information 1344
Modules editor 1348
Configuration of Modicon M340 processors 1348
Configuration of Modicon M340 Controllers..... 1348
Configuration of Premium processors..... 1361
Configuration of Premium\Atrium processors 1361
Access to PCMCIA memory cards of Premium processors..... 1368
Access to control loop configuration for Premium processors 1369
Configuration of the Quantum processors..... 1371
Configuration of Quantum Controllers..... 1371
Configuration of Modbus ports on Quantum 1377
Configuration of the input\output modules for a Premium\Atrium, Quantum or Modicon M340 station..... 1380
Access to the Modicon, Premium or Atrium I/O Module Configuration Editor..... 1380
Access the Quantum Controller Input/Output Module Configuration Editor..... 1382

General functionalities of the editor 1385
Analysis of the configuration 1385
Export\Import of the configuration 1386
Printing the configuration 1386
Debugging and adjustment 1389
Debugging the program 1390
Safety precautions 1390
Modification in RUN 1391
Switch to STOP of a Controller in Debug mode 1391
Setting a Task to STOP 1392
Task Deactivation 1393
Deactivation of Inputs/Outputs 1394
Use of the Step by Step Mode and Breakpoint 1395
Unwanted Disconnection in Debug Mode for Premium Controllers 1395
Debug Screen Via a Fipway, Fipio or Uni-Telway Network with a PCMCIA Card 1396
Behavior of Forced Variables Between Modsoft/NxT/Concept and EcoStruxure Control Expert 1396
Program Animation 1397
Introduction to animation 1397
Section in textual languages (Structured Text or Instruction List) 1401
Section in a Ladder Diagram language 1402
Section in Function Block Diagram (FBD) language 1405
Section in Sequential Function Chart Language (SFC) 1407
Display of Prepositioned steps, after an on-line modification 1408
DFB instances (user function blocks) 1409
Representation of forced variables 1412
Debugging in Ladder Language 1413
Inserting a breakpoint in Ladder Diagram language (LD) 1413
Step by step execution mode in Ladder Diagram language (LD) 1416
Inserting a Watchpoint in Ladder Diagram Language 1419
Debugging in Textual Languages (Structured Text, Instruction List) 1421
Insertion of a breakpoint in textual languages (Structured Text or Instruction List) 1422

Step by step execution mode in textual languages (Structured Text or Instruction List) 1424
Insertion of a Watchpoint in Textual Languages (Structured Text or Instruction List) 1427
Debugging in Function Block Diagram (FBD) Language 1430
Inserting a breakpoint in Function Block Diagram language 1430
Step by step execution mode in Function Block Diagram language (FBD) 1433
Inserting a watchpoint in Function Block Diagram language (FBD) 1436
Debugging User Function Block (DFB) Instances 1439
Inserting a breakpoint in a DFB instance (user function block) 1440
Step by step execution mode in a DFB instance (user function block) 1441
Inserting a watchpoint in a user function block (DFB) 1442
Debugging in Sequential Function Chart Language (SFC) 1443
Insert a Breakpoint in Sequential Function Chart Language (SFC) 1443
Step by Step Execution Mode in Sequential Function Chart Language (SFC) 1445
Inserting a watchpoint in Sequential Function Chart language 1449
Information on Debugging 1450
Tracing task execution 1450
Task states during debugging 1452
Disconnection During the Debug Phase 1452
Operation of the Alarm Relay in Debug Mode 1453
PLC debug screen 1453
Overview of the CPU debugging screen 1454
Description of the Task tab: Task checking field 1455
Description of the Task tab: Operating mode field 1457
Description of tab: Realtime Clock 1460
Description of the Tab: Information 1461
Viewing and adjusting variables 1466
Introduction to variable animation 1466
Displaying variables in debug mode 1466
Variable Display Window 1467
Introduction to the Variable Output Window 1468

Variable Display Boxes 1470
Introduction to variable display boxes 1471
Animation Tables 1473
Permanent and Temporary Animation Tables 1473
Introduction to Animation Tables 1475
Managing Animation Tables 1476
Multiple Modifications of Animation Table Variables 1479
Animation Table Properties 1480
Basic functions 1481
Animation of Strings in the Animation Table 1485
Modification Mode 1487
Force Mode 1492
Multiple modification and forcing of Boolean variables 1497
Animation Table of Forced Bits 1500
Instruction Editor (LL984) 1501
Instruction Editor 1501
Modifying values: 1504
Modifying values: 1504
Machine-Process Simulator 1508
Introduction 1508
Diagnostics 1510
Introduction to diagnostics tools 1511
Diagnostics with EcoStruxure Control Expert 1511
Diagnostics screens 1515
Module and processor diagnostics screens 1515
Viewing Diagnostics 1516
Lists of Diagnostics 1517
Diagnostics Viewer 1522
Reminder of the Diagnostics Associated with the Viewer 1522
Operation of the Diagnostics in EcoStruxure Control Expert 1524
Description of the **Diagnostics viewer** 1525
How to find causes 1528
Configuring the display 1530
Managing Error Messages 1532
Diagnostics Viewer Commands 1534

DiagViewer Properties 1536
Archiving Error Messages 1539
System diagnostics 1543
System Diagnostic 1543
Project diagnostics 1548
How to Create a User Diagnostic DFB 1548
Project diagnostic 1550
Operator screen 1554
General information 1555
Introduction 1555
Creation of Runtime Screens 1556
First steps 1558
Working principles 1558
How to access the operator screens 1558
Menus and tools in the runtime screen editor 1559
Menus in the operator screen editor 1560
Edit toolbar 1563
Configuring options from the runtime screen editor 1566
General configuration of operator screens 1567
Editing runtime screens 1568
Editing screens from the browser 1568
Editing operator screens 1568
Operator screen editor functions 1570
Screen Family Editor Functions 1574
Operator screen properties 1576
Properties of Operator Screen Families 1578
Creating an object 1580
Introduction of a graphic screen object 1580
How to create standard objects 1580
How to insert an image in a screen 1582
How to create driving objects 1584
Operation of the driving objects 1585
Default Properties screen 1589
Default Properties screen 1589
Modification of an object’s attributes 1591

How to access an object’s attributes 1591
Design tab 1592
Extended colors 1593
Text tab 1594
Animation Tab 1596
Animation type tab 1598
Trend diagrams 1601
Bar chart 1602
Screen browse tab 1604
Control tab 1606
Image Tab 1608
Explicit Exchanges Tab 1610
Information on the objects 1611
The data selection tool 1613
Manipulation of objects in runtime screens 1616
How to select graphic objects 1616
How to move and resize objects 1617
How to delete and copy objects 1619
How to position objects 1620
How to create compound objects 1622
Managing the runtime screens messages editor 1624
Operator screens messages editor 1624
How to manage messages linked to operator screens 1626
Find dialog box 1628
Replace dialog box 1629
Message Renumbering Dialog Box 1630
Message editing and import/export function 1631
Runtime screens in online mode 1635
Basic functions in online mode 1635
How to select an animated object 1636
How to use command objects to modify the variables associated with objects 1637
How to view the variables associated with graphic objects 1638
How to modify the variables associated with objects from an animation table 1639

How to search for cross-references ... 1640
How to manage the display of operator screens ... 1641
How to modify screen control ... 1643
How to obtain information on objects when in connected mode ... 1644
Runtime screens objects library ... 1646
Managing library objects ... 1646
The Objects Library ... 1646
How to manage the list of graphic items in the library ... 1647
Properties of the object library and its elements ... 1650
Handling the Objects ... 1653
Dereferencing in Operator Screens ... 1656
Documentation, Import/export ... 1659
Documentation ... 1660
Creating Project Documentation ... 1660
Documentation Structure ... 1661
Topics in Documentation ... 1663
Creating Project Documentation ... 1672
Print Preview ... 1675
Page Setup ... 1677
Print ... 1678
Printout Setup ... 1681
Import / Export ... 1686
General information on the Import/Export function ... 1686
General Information on Import/Export ... 1686
Import / Export Files ... 1688
Type of dialog box for the export function ... 1689
Type of dialog boxes for the import function ... 1692
Managing Conflicts ... 1694
Importing/Exporting the various elements of Control Expert ... 1696
Export of the input/output configuration ... 1696
Import the Input/Output Configuration ... 1698
Exporting the program ... 1699
Import the Program ... 1701
Exporting DFB types ... 1703
Importing DFB types ... 1704

Exporting DDTs 1705
Importing DDTs 1706
Exporting Variables 1707
Importing Variables 1710
Exporting the communication network configuration 1711
Importing the configuration of a communication network 1712
Exporting operator screens 1713
Import Operator Screens 1715
Exporting animation tables 1716
Importing animation tables 1717
Exporting a functional module 1718
Importing a functional module 1719
Export a Global Project 1720
Importing a global project 1722
Exporting/Importing LL984 Segments/Networks 1725
Wizard for importing the various elements of Control Expert 1727
Using the assistant 1727
Variables tab of the wizard 1731
DFB and DDT Tabs of the Wizard 1733
SR, Sections, Events and Macro steps tabs of the wizard 1734
Wizard animation tables tab 1736
Wizard functional module tabs 1736
Correspondence File 1738
Conversion Wizard 1741
Conversion Wizard 1742
General Description of the Conversion Wizard 1742
Introduction 1742
Access to Conversion Wizard 1744
General Representation 1746
Conversion and Result 1747
Structure Tab of the Conversion Wizard 1748
Introduction 1748
Selecting Application Parts to be Converted 1750
I/O Remapping Tab of the Conversion Wizard 1752
Introduction 1752

Remapping Table ... 1755
Remapping Channels, Groups of Channels or Specific I/O Types ... 1758
Clustering and Remapping of Memory Variables ... 1761
Memory Area Tabs of the Conversion Wizard ... 1762
Concurrent Editing ... 1764
Exclude, Communication -1, Type Restriction ... 1766
CPU Configuration Tab of the Conversion Wizard ... 1770
Introduction ... 1771
Appendices ... 1773
Hyperlinks in Control Expert ... 1774
General ... 1774
Introduction to Hyperlinks ... 1774
Representation and opening of a hyperlink ... 1776
Principles ... 1777
Definitions ... 1777
Overview of operate-modes ... 1778
1-step mode ... 1778
2-step mode ... 1780
Description of the Hyperlink Comment Window ... 1781
Properties ... 1784
List of barred characters ... 1785
Hyperlinks in the project browser ... 1786
Creating a hyperlink in the project browser ... 1786
Editing the properties of a hyperlink in the project browser ... 1787
Opening a hyperlink from the project browser ... 1789
Deleting a hyperlink in the project browser ... 1789
Hyperlink in the data editor ... 1790
Creating a hyperlink in the data editor ... 1790
Editing a hyperlink in the data editor ... 1792
Selecting a hyperlink in the data editor ... 1794
Deleting a hyperlink in the data editor ... 1795
Hyperlinks in the DFB type attributes ... 1796
Creating a Hyperlink from DFB Type Attributes ... 1796
Modifying a Hyperlink from DFB Type Attributes ... 1798
Selecting a Hyperlink from DFB Type Attributes ... 1800

Deleting a Hyperlink from DFB Type Attributes ... 1802
Hyperlinks in animation tables ... 1804
Creating a hyperlink in an animation table ... 1804
Editing a hyperlink from an animation table ... 1806
Selecting a hyperlink from an animation table ... 1807
Deleting a hyperlink from an animation table ... 1808
Hyperlinks in runtime screens ... 1808
Creating a hyperlink from operator screens ... 1809
Editing a hyperlink in the operator screens ... 1810
Selecting a hyperlink in an operator screen ... 1812
Deleting a hyperlink in an operator screen ... 1812
Hyperlinks in the documentation folder ... 1814
Creating a hyperlink in the general information of the project folder ... 1814
Editing a hyperlink in the documentation folder ... 1816
Deleting a hyperlink in the documentation folder ... 1818
Hyperlinks and drag-and-drop ... 1819
Drag/drop ... 1820
User Interface ... 1825
Window ... 1825
Anchored window ... 1827
Menus ... 1830
Menu commands ... 1832
Toolbar ... 1835
Output Windows ... 1841
Status bar ... 1843
Directory Structures ... 1846
Using the Mouse ... 1847
Using the Keyboard ... 1848
Data selection boxes ... 1850
Dialog boxes ... 1851
Keyboard Shortcuts ... 1857
General Keyboard Shortcuts ... 1857
Bus Editor Keyboard Shortcuts ... 1857
Control Expert Main Screen and Project Browser Keyboard Shortcuts ... 1858

Configuring the Project ... 1860
PLC Configuration Screen Keyboard Shortcuts ... 1860
Creating the Project ... 1861
ST/IL Editors Keyboard Shortcuts ... 1861
FBD Editor Keyboard Shortcuts ... 1863
SFC Editor Keyboard Shortcuts ... 1864
LD Editor Keyboard Shortcuts ... 1866
Data Editor Keyboard Shortcuts ... 1868
Type Library Browser Keyboard Shortcuts ... 1869
Type Library Manager Keyboard Shortcuts ... 1870
Operating the Project ... 1870
Search Replace Tool Screen Keyboard Shortcuts ... 1871
Diagnostic Viewer Screen Keyboard Shortcuts ... 1871
PLC Screen Keyboard Shortcuts ... 1871
Animation Screen Keyboard Shortcuts ... 1872
Operator Screen Keyboard Shortcuts ... 1873
Documenting the Project ... 1874
Documentation Screen Keyboard Shortcuts ... 1874
Glossary ... 1875
Index ... 1882

Safety Information

Important Information

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, service, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.



The addition of this symbol to a “Danger” or “Warning” safety label indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.



This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠ DANGER

DANGER indicates a hazardous situation which, if not avoided, **will result in** death or serious injury.

⚠ WARNING

WARNING indicates a hazardous situation which, if not avoided, **could result in** death or serious injury.

⚠ CAUTION

CAUTION indicates a hazardous situation which, if not avoided, **could result** in minor or moderate injury.

NOTICE

NOTICE is used to address practices not related to physical injury.

Please Note

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

Before You Begin

Do not use this product on machinery lacking effective point-of-operation guarding. Lack of effective point-of-operation guarding on a machine can result in serious injury to the operator of that machine.

⚠WARNING
UNGUARDED EQUIPMENT • Do not use this software and related automation equipment on equipment which does not have point-of-operation protection. • Do not reach into machinery during operation. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

This automation equipment and related software is used to control a variety of industrial processes. The type or model of automation equipment suitable for each application will vary depending on factors such as the control function required, degree of protection required, production methods, unusual conditions, government regulations, etc. In some applications, more than one processor may be required, as when backup redundancy is needed.

Only you, the user, machine builder or system integrator can be aware of all the conditions and factors present during setup, operation, and maintenance of the machine and, therefore, can determine the automation equipment and the related safeties and interlocks which can be properly used. When selecting automation and control equipment and related software for a particular application, you should refer to the applicable local and national standards and regulations. The National Safety Council's Accident Prevention Manual (nationally recognized in the United States of America) also provides much useful information.

In some applications, such as packaging machinery, additional operator protection such as point-of-operation guarding must be provided. This is necessary if the operator's hands and

other parts of the body are free to enter the pinch points or other hazardous areas and serious injury can occur. Software products alone cannot protect an operator from injury. For this reason the software cannot be substituted for or take the place of point-of-operation protection.

Ensure that appropriate safeties and mechanical/electrical interlocks related to point-of-operation protection have been installed and are operational before placing the equipment into service. All interlocks and safeties related to point-of-operation protection must be coordinated with the related automation equipment and software programming.

NOTE: Coordination of safeties and mechanical/electrical interlocks for point-of-operation protection is outside the scope of the Function Block Library, System User Guide, or other implementation referenced in this documentation.

Start-up and Test

Before using electrical control and automation equipment for regular operation after installation, the system should be given a start-up test by qualified personnel to verify correct operation of the equipment. It is important that arrangements for such a check are made and that enough time is allowed to perform complete and satisfactory testing.

⚠WARNING

EQUIPMENT OPERATION HAZARD

- Verify that all installation and set up procedures have been completed.
- Before operational tests are performed, remove all blocks or other temporary holding means used for shipment from all component devices.
- Remove tools, meters, and debris from equipment.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Follow all start-up tests recommended in the equipment documentation. Store all equipment documentation for future references.

Software testing must be done in both simulated and real environments.

Verify that the completed system is free from all short circuits and temporary grounds that are not installed according to local regulations (according to the National Electrical Code in the U.S.A, for instance). If high-potential voltage testing is necessary, follow recommendations in equipment documentation to prevent accidental equipment damage.

Before energizing equipment:

- Remove tools, meters, and debris from equipment.

- Close the equipment enclosure door.
- Remove all temporary grounds from incoming power lines.
- Perform all start-up tests recommended by the manufacturer.

Operation and Adjustments

The following precautions are from the NEMA Standards Publication ICS 7.1-1995:

(In case of divergence or contradiction between any translation and the English original, the original text in the English language will prevail.)

- Regardless of the care exercised in the design and manufacture of equipment or in the selection and ratings of components, there are hazards that can be encountered if such equipment is improperly operated.
- It is sometimes possible to misadjust the equipment and thus produce unsatisfactory or unsafe operation. Always use the manufacturer's instructions as a guide for functional adjustments. Personnel who have access to these adjustments should be familiar with the equipment manufacturer's instructions and the machinery used with the electrical equipment.
- Only those operational adjustments required by the operator should be accessible to the operator. Access to other controls should be restricted to prevent unauthorized changes in operating characteristics.

About the Book

Document Scope

The information contained herein describes programming Modicon M340, Modicon M580, Momentum, Premium, Atrium, or Quantum controllers using the IEC 61131-3 conforming programming languages Function Block Diagram (FBD), Ladder Diagram (LD), Sequential Function Chart (SFC), Instruction List (IL) and Structured Text (ST).

Validity Note

The information contained herein has been updated for the release of EcoStruxure™ Control Expert 16.0.

For product compliance and environmental information (RoHS, REACH, PEP, EOLI, etc.), go to www.se.com/ww/en/work/support/green-premium/.

Available Languages of this Document

The information contained herein is available in the following languages:

- English (33003101 (ENG))
- French (33003102 (FRE))
- German (33003103 (GER))
- Spanish (33003104 (SPA))
- Italian (33003696 (ITA))
- Chinese (33003697 (CHS))

Related Documents

Title of documentation	Reference number
EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual	35006144 (ENG) 35006145 (FRE) 35006146 (GER) 35006147 (SPA) 35013361 (ITA) 35013362 (CHS)
EcoStruxure™ Control Expert, System Bits and Words, Reference Manual	EIO0000002135 (ENG) EIO0000002136 (FRE) EIO0000002137 (GER) EIO0000002139 (SPA) EIO0000002138 (ITA) EIO0000002140 (CHS)
EcoStruxure™ Control Expert, Installation Manual	35014793 (ENG) 35014792 (FRE) 35014794 (GER) 35014795 (SPA) 35014796 (ITA) 35012191 (CHS)
EcoStruxure™ Control Expert, LL984 Editor, Reference Manual	EIO0000000549 (ENG) EIO0000000802 (FRE) EIO0000000803 (GER) EIO0000000804 (ITA) EIO0000000805 (SPA) EIO0000000806 (CHS)
EcoStruxure™ Control Expert, Concept Application Converter, User Manual	33002515 (ENG) 33002516 (FRE) 33002517 (GER) 33003676 (ITA) 33002518 (SPA) 33003677 (CHS).
EcoStruxure™ Control Expert, PL7 Application Converter, User Manual	35006148 (ENG) 35006149 (FRE) 35006150 (GER) 35013967 (ITA) 35006151 (SPA) 35012199 (CHS).
Modicon M580, Hardware, Reference Manual	EIO0000001578 (ENG) EIO0000001579 (FRE) EIO0000001580 (GER) EIO0000001582 (ITA) EIO0000001581 (SPA) EIO0000001583 (CHS)
Modicon M580, Change Configuration on the Fly, User Guide	EIO0000001590 (ENG) EIO0000001591 (FRE) EIO0000001592 (GER) EIO0000001594 (ITA) EIO0000001593 (SPA) EIO0000001595 (CHS)

Title of documentation	Reference number
Modicon M580 Hot Standby, System Planning Guide for Frequently Used Architectures	NHA58880 (ENG) NHA58881 (FRE) NHA58882 (GER) NHA58883 (ITA) NHA58884 (SPA) NHA58885 (CHS)
Modicon M580, System Planning Guide for Complex Topologies	NHA58892 (ENG) NHA58893 (FRE) NHA58894 (GER) NHA58895 (ITA) NHA58896 (SPA) NHA58897 (CHS)
Modicon M580, Safety Manual	QGH46982 (ENG) QGH46983 (FRE) QGH46984 (GER) QGH46985 (ITA) QGH46986 (SPA) QGH46987 (CHS)
Modicon M340 for Ethernet, Communications Modules and Processors, User Manual	31007131 (ENG) 31007132 (FRE) 31007133 (GER) 31007494 (ITA) 31007134 (SPA) 31007493 (CHS).
Premium and Atrium using EcoStruxure™ Control Expert, Counter Modules, User Manual	35006216 (ENG) 35006217 (FRE) 35006218 (GER) 35013968 (ITA) 35006219 (SPA) 35012200 (CHS).
Premium and Atrium using EcoStruxure™ Control Expert, Process Control, User Manual	35006242 (ENG) 35006243 (FRE) 35006244 (GER) 35014013 (ITA) 35006245 (SPA) 35012198 (CHS)
Communication Services and Architectures, Reference Manual	35010500 (ENG) 35010501 (FRE) 35006176 (GER) 35013966 (ITA) 35006177 (SPA) 35012196 (CHS).
Premium and Atrium using EcoStruxure™ Control Expert, Fipio Bus, Setup Manual	35008155 (ENG) 35008156 (FRE) 35008157 (GER) 35013953 (ITA) 35008158 (SPA) 35013954 (CHS).

Title of documentation	Reference number
Premium and Atrium using EcoStruxure™ Control Expert, Fipway Network, User Manual	35006183 (ENG) 35006185 (FRE) 35006186 (GER) 35013955 (ITA) 35006187 (SPA) 35013956 (CHS).
Premium and Atrium using EcoStruxure™ Control Expert, Ethernet Network Modules, User Manual	35006192 (ENG) 35006193 (FRE) 35006194 (GER) 31007214 (ITA) 35006195 (SPA) 31007102 (CHS).
Premium and Atrium using EcoStruxure™ Control Expert, Modbus Plus Network, User Manual	35006188 (ENG) 35006189 (FRE) 35006190 (GER) 35013962 (ITA) 35006191 (SPA) 35013963 (CHS).
Communication Drivers, Installation Manual	35006152 (ENG) 35006153 (FRE) 35006154 (GER) 35013950 (ITA) 35006155 (SPA) 35012192 (CHS).
Premium using EcoStruxure™ Control Expert, Motion Function Blocks, Start-up Guide	35010601 (ENG) 35010602 (FRE) 35010603 (GER) 35010600 (ITA) 35010604 (SPA) 35012309 (CHS).
Modicon M340, Processors, Setup Manual	35012676 (ENG) 35012677 (FRE) 35013351 (GER) 35013352 (ITA) 35013353 (SPA) 35013354 (CHS)
Modicon X80, Analog Input/Output Modules, User Manual	35011978 (ENG) 35011980 (FRE) 35011979 (GER) 35011982 (ITA) 35011981 (SPA) 35011983 (CHS)
Modicon M340, CANopen, Setup Manual	35013944 (ENG) 35013945 (FRE) 35013946 (GER) 35013948 (ITA) 35013947 (SPA) 35013949 (CHS)

Title of documentation	Reference number
Quantum using EcoStruxure™ Control Expert, Hardware Reference Manual	35010529 (ENG) 35010530 (FRE) 35010531 (GER) 35013975 (ITA) 35010532 (SPA) 35012184 (CHS)
Quantum using EcoStruxure™ Control Expert, Change Configuration On The Fly, User Guide	S1A48967 (ENG) S1A48968 (FRE) S1A48969 (GER) S1A48970 (ITA) S1A48972 (SPA) S1A48976 (CHS).
Momentum for EcoStruxure™ Control Expert, 171 CBU 78090, 171 CBU 98090, 171 CBU 98091 Processors, User Guide	HRB44124 (ENG)
EcoStruxure™ Control Expert, Standard, Block Library	33002519 (ENG) 33002520 (FRE) 33002521 (GER) 33003678 (ITA) 33002522 (SPA) 33003679 (CHS).
EcoStruxure™ Control Expert, Diagnostics, Block Library	33002523 (ENG) 33002524 (FRE) 33002525 (GER) 33003680 (ITA) 33002526 (SPA) 33003681 (CHS).
EcoStruxure™ Control Expert, Communication, Block Library	33002527 (ENG) 33002528 (FRE) 33002529 (GER) 33003682 (ITA) 33002530 (SPA) 33003683 (CHS).
EcoStruxure™ Control Expert, I/O Management, Block Library	33002531 (ENG) 33002532 (FRE) 33002533 (GER) 33003684 (ITA) 33002534 (SPA) 33003685 (CHS)
EcoStruxure™ Control Expert, System, Block Library	33002539 (ENG), 33002540K01000 (FRE), 33002541K01000 (GER), 33003688K01000 (ITA), 33002542K01000 (SPA), 33003689K01000 (CHS).

Title of documentation	Reference number
EcoStruxure™ Control Expert, UnityLL984, Block Library	EIO0000000550 (ENG) EIO0000000807 (FRE) EIO0000000808 (GER) EIO0000000809 (ITA) EIO0000000810 (SPA) EIO0000000811 (CHS)
System Time Stamping, User Guide	EIO0000001217 (ENG) EIO0000001707 (FRE) EIO0000001708 (GER) EIO0000001710 (ITA) EIO0000001709 (SPA) EIO0000001711 (CHS)
EcoStruxure™ Control Expert, Applicative Time Stamping, User Guide	EIO0000001268 (ENG) EIO0000001702 (FRE) EIO0000001703 (GER) EIO0000001705 (ITA) EIO0000001704 (SPA) EIO0000001706 (CHS)

To find documents online, visit the Schneider Electric download center (www.se.com/ww/en/download/).

Product Related Information

⚠WARNING

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA), or equivalent risk analysis, of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate them.
- Review the implications of communication link interruptions and take actions to mitigate them.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and error conditions) according to your risk assessment, and applicable codes and regulations.
- Apply local accident prevention and safety regulations and guidelines.[1]
- Test each implementation of a system for proper operation before placing it into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

[1] For additional information, refer to NEMA ICS 1.1 (latest edition), *Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control* and to NEMA ICS 7.1 (latest edition), *Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems* or their equivalent governing your particular location.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

- Only use software approved by Schneider Electric for use with this equipment.
- Update your application program every time you change the physical hardware configuration.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

The examples given herein are for information only.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

Adapt examples given herein to the specific functions and requirements of your industrial application before you implement them.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Trademarks

QR Code is a registered trademark of DENSO WAVE INCORPORATED in Japan and other countries.

Information on Non-Inclusive or Insensitive Terminology

As a responsible, inclusive company, Schneider Electric is constantly updating its communications and products that contain non-inclusive or insensitive terminology. However, despite these efforts, our content may still contain terms that are deemed inappropriate by some customers.

Terminology Derived from Standards

The technical terms, terminology, symbols and the corresponding descriptions in the information contained herein, or that appear in or on the products themselves, are generally derived from the terms or definitions of international standards.

In the area of functional safety systems, drives and general automation, this may include, but is not limited to, terms such as *safety*, *safety function*, *safe state*, *fault*, *fault reset*, *malfunction*, *failure*, *error*, *error message*, *dangerous*, etc.

Among others, these standards include:

Standard	Description
IEC 61131-2:2007	Programmable controllers, part 2: Equipment requirements and tests.
ISO 13849-1:2023	Safety of machinery: Safety related parts of control systems. General principles for design.
EN 61496-1:2013	Safety of machinery: Electro-sensitive protective equipment. Part 1: General requirements and tests.
ISO 12100:2010	Safety of machinery - General principles for design - Risk assessment and risk reduction
EN 60204-1:2006	Safety of machinery - Electrical equipment of machines - Part 1: General requirements
ISO 14119:2013	Safety of machinery - Interlocking devices associated with guards - Principles for design and selection
ISO 13850:2015	Safety of machinery - Emergency stop - Principles for design
IEC 62061:2021	Safety of machinery - Functional safety of safety-related electrical, electronic, and electronic programmable control systems
IEC 61508-1:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: General requirements.
IEC 61508-2:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Requirements for electrical/electronic/programmable electronic safety-related systems.
IEC 61508-3:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Software requirements.
IEC 61784-3:2021	Industrial communication networks - Profiles - Part 3: Functional safety fieldbuses - General rules and profile definitions.
2006/42/EC	Machinery Directive
2014/30/EU	Electromagnetic Compatibility Directive
2014/35/EU	Low Voltage Directive

In addition, terms used in the information contained herein may tangentially be used as they are derived from other standards such as:

Standard	Description
IEC 60034 series	Rotating electrical machines
IEC 61800 series	Adjustable speed electrical power drive systems
IEC 61158 series	Digital data communications for measurement and control – Fieldbus for use in industrial control systems

Finally, the term zone of operation may be used in conjunction with the description of specific hazards, and is defined as it is for a hazard zone or danger zone in the Machinery Directive (2006/42/EC) and ISO 12100:2010.

NOTE: The aforementioned standards may or may not apply to the specific products cited in the present documentation. For more information concerning the individual standards applicable to the products described herein, see the characteristics tables for those product references.

General

What's in This Part

General Presentation of Control Expert....46
Programming Language Editors....55
Connecting a terminal to a PLC....68
Global project management....71
Project browser....161
Functional modules....237
FDT Container....267
Type Library Manager....314

Subject of this Part

This part presents Control Expert services and describes the different tools used to manage a project: access rights, save/restore, online/offline mode services, etc.

General Presentation of Control Expert

What's in This Chapter

Capabilities of Control Expert.....46
User Interface.....51

Overview

This chapter describes the general features of a project created using Control Expert.

Capabilities of Control Expert

Hardware Platforms

Control Expert supports the following hardware platforms:

- Modicon M340
- Modicon M580
- Quantum
- Momentum
- Premium
- Atrium

Programming Languages

Control Expert provides the following programming languages for creating the user program:

- Function Block Diagram FBD
- Ladder Diagram (LD) language
- Instruction List IL
- Structured Text ST
- Sequential Control SFC
- Ladder Logic 984 (LL984)

All of these programming languages can be used together in the same project.

All these languages (except LL984) conform to IEC 61131-3.

Block Libraries

The blocks that are included in the delivery of Control Expert extensive block libraries extend from blocks for simple boolean operations, through blocks for strings and array operations to blocks for controlling complex control loops.

For a better overview, the different blocks are arranged in libraries, which are then broken down into families.

The blocks can be used in the programming languages FBD, LD, IL, and ST.

Elements of a Program

A program can be constructed from:

- a master task (MAST)
- a FAST task (not available for Momentum)
- one to 4 AUX tasks (not available for Modicon M340 and Momentum)
- Program Units which are assigned one of the defined tasks (available for Modicon M580 and Modicon M340)
- sections, which are assigned one of the defined tasks
- sections for processing time controlled events (Timerx, not available for Momentum)
- sections for processing hardware controlled events (EVTx, not available for Momentum)
- subroutine sections (SR)

Software Licenses

There is one Control Expert installation setup (.iso file) and the license determines the version that can be launched.

The following software versions are available:

- Control Expert S
- Control Expert L
- Control Expert XL
- Control Expert XL with M580 Safety

The M580 Safety CPUs are included in a **Safety add-on** available for Control Expert L and XL.

Two types of licenses are available to activate Control Expert:

- Node-locked license for single use on a local PC.
- Floating license for multiple uses of an authorized number of PCs in a network connected to the Enterprise License Server.

For detailed information on license activation and/or registration, refer to *EcoStruxure™ Control Expert, Installation Manual*.

Performance Scope

This table shows the main characteristics of the individual software versions:

	Control Expert S	**Control Expert L**	**Control Expert L + Safety add-on**	**Control Expert XL**	**Control Expert XL + Safety add-on**
Programming languages					
Function Block Diagram FBD	+	+	+	+	+
Ladder Diagram (LD) language	+	+	+	+	+
Instruction List IL	+	+	+(2)	+	+(2)
Structured Text ST	+	+	+(2)	+	+(2)
Sequential Language SFC	+	+	+(2)	+	+(2)
Ladder Logic 984 (LL984)	+	+	+	+	+
Libraries(1)					
Standard library	+	+	+(2)	+	+(2)
Control library	+	+	+(2)	+	+(2)
Communication library	+	+	+(2)	+	+(2)
Diagnostics library	+	+	+(2)	+	+(2)
I/O management library	+	+	+(2)	+	+(2)
System library	+	+	+(2)	+	+(2)

	Control Expert S	Control Expert L	Control Expert L + Safety add-on	Control Expert XL	Control Expert XL + Safety add-on
Motion control drive library	-	+	+(2)	+	+(2)
TCP Open library	-	optional	optional(2)	optional	optional(2)
Obsolete library	+	+	+(2)	+	+(2)
MFB library	+	+	+(2)	+	+(2)
Safety library	-	-	+	-	+
Memory card file management library	+	+	+(2)	+	+(2)
General information					
Create and use data structures (DDTs)	+	+	+(2)	+	+(2)
Create and use Derived Function Blocks (DFBs)	+	+	+	+	+
Project browser with structural and/or functional view	+	+	+	+	+
Managing access rights	+	+	+	+	+
Operator screen	+	+	+	+	+
Diagnostic viewer	+	+	+	+	+
System diagnostics	+	+	+	+	+
Project diagnostics	+	+	+(2)	+	+(2)
Trending Tool	+	+	+	+	+
Application converter	PL7 converter Concept Converter Partial conversion	PL7 converter Concept Converter	PL7 converter Concept Converter	PL7 converter Concept Converter	PL7 converter Concept Converter
Managing multi-stations	-	-	-	-	-
Supported platforms					
Modicon M340	All CPUs	All CPUs	All CPUs	All CPUs	All CPUs
Modicon M580	-	BMEP5810••	BMEP5810••	BMEP5810••	BMEP5810••

	Control Expert S	**Control Expert L**	**Control Expert L + Safety add-on**	**Control Expert XL**	**Control Expert XL + Safety add-on**
		BMEP5820•• BMEP5830•• BMEH582040	BMEP5820•• BMEP5830•• BMEH582040 BMEP582040S BMEH582040S	BMEP5820•• BMEP5830•• BMEP5840•• BMEP585040 BMEP586040 BMEH582040 BMEH584040 BMEH586040	BMEP5820•• BMEP5830•• BMEP5840•• BMEP585040 BMEP586040 BMEH582040 BMEH584040 BMEH586040 BMEP582040S BMEP584040S BMEH582040S BMEH584040S BMEH586040S
Momentum	171CBU78090 171CBU98090 171CBU98091	171CBU78090 171CBU98090 171CBU98091	171CBU78090 171CBU98090 171CBU98091	171CBU78090 171CBU98090 171CBU98091	171CBU78090 171CBU98090 171CBU98091
Premium	-	All CPUs **except**: P57 554M P57 5634M P57 6634M	All CPUs **except**: P57 554M P57 5634M P57 6634M	All CPUs	All CPUs
Quantum	-	140CPU31110 140CPU43412 U/A* 140CPU53414 U/A* * Upgrade using OS Loader	140CPU31110 140CPU43412 U/A* 140CPU53414 U/A* * Upgrade using OS Loader	140CPU31110 140CPU43412 U/A 140CPU53414 U/A 140CPU65150 140CPU65160 140CPU65260 140CPU65860 140CPU67060	140CPU31110 140CPU43412 U/A 140CPU53414 U/A 140CPU65150 140CPU65160 140CPU65260 140CPU65860 140CPU67060

	Control Expert S	Control Expert L	Control Expert L + Safety add-on	Control Expert XL	Control Expert XL + Safety add-on
				140CPU67160 140CPU67260 140CPU67261 140CPU67861	140CPU67160 140CPU67260 140CPU67261 140CPU67861
Atrium	-	All CPUs	All CPUs	All CPUs	All CPUs
Simulator	+	+	+	+	+
Openness					
Hyperlinks	+	+	+	+	+
Control Expert Server (for OFS, UAG)	+	+	+	+	+
Software components contained in the software package					
Documentation as context help and PDF	+	+	+	+	+
OS Loader tool + HW firmware	+	+	+	+	+

+ Available

- Not available

(1) Availability of the blocks depends on the hardware platforms.

(2) Available on all PLC except platforms M580 Safety.

Naming Convention

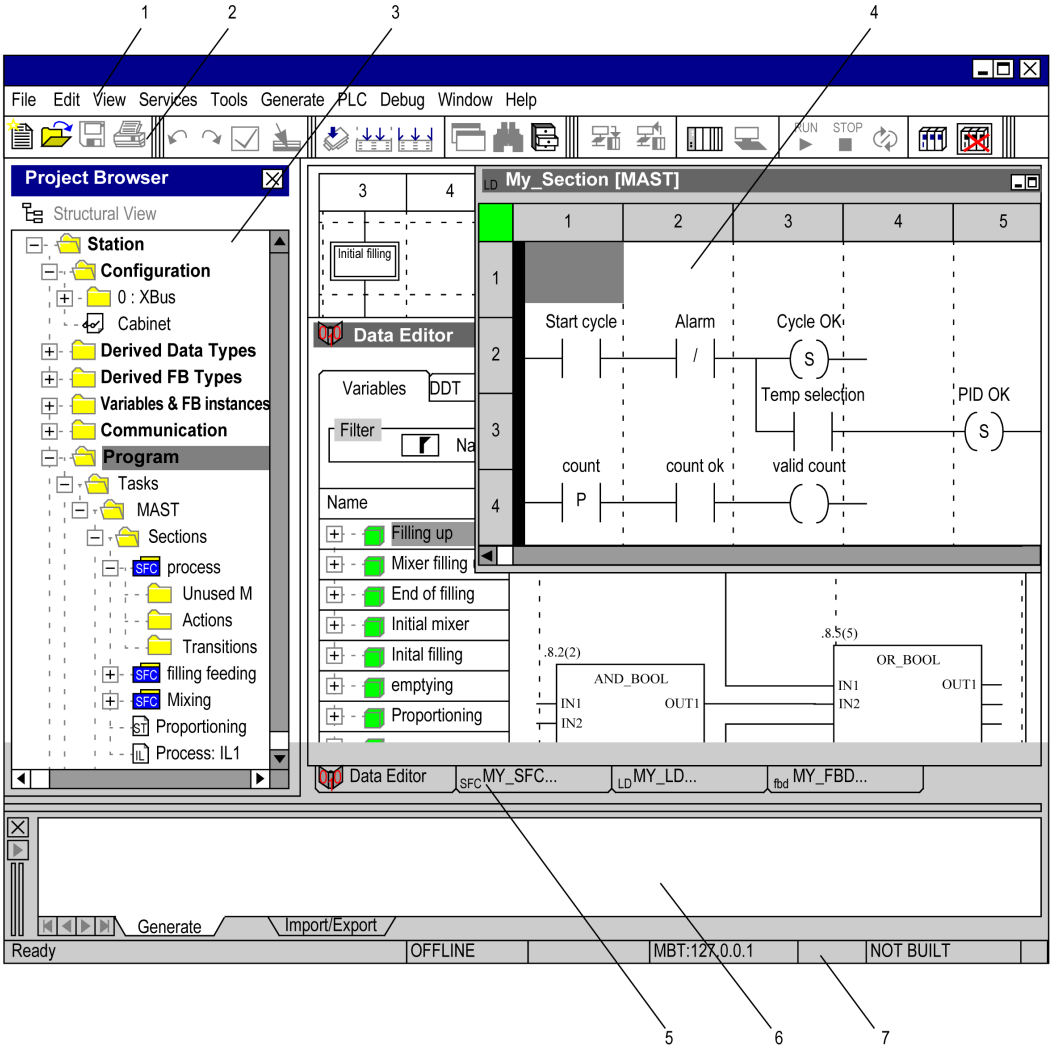
In the rest of this document, "Control Expert" is used as general term for "Control Expert S", "Control Expert L", and "Control Expert XL", with or without Safety add-on.

User Interface

Overview

The user interface consists of several, configurable windows and toolbars.

User interface:



Legend:

Number	Description
1	Menu bar, page 1830
2	Toolbar, page 1835
3	Project Browser, page 161
4	Editor window (programming language editors, data editor, etc.)

Number	Description
5	Register tabs for direct access to the editor window
6	Information window, page 1841 (provides information about errors which have occurred, signal tracking, import functions, etc.)
7	Status bar, page 1843

Programming Language Editors

What's in This Chapter

Function Block Diagram FBD 55
Ladder Diagram (LD) Language 58
General Information about SFC Sequence Language 60
Instruction List IL 64
Structured Text ST 65

Overview

This chapter describes the different programming languages provided by Control Expert.

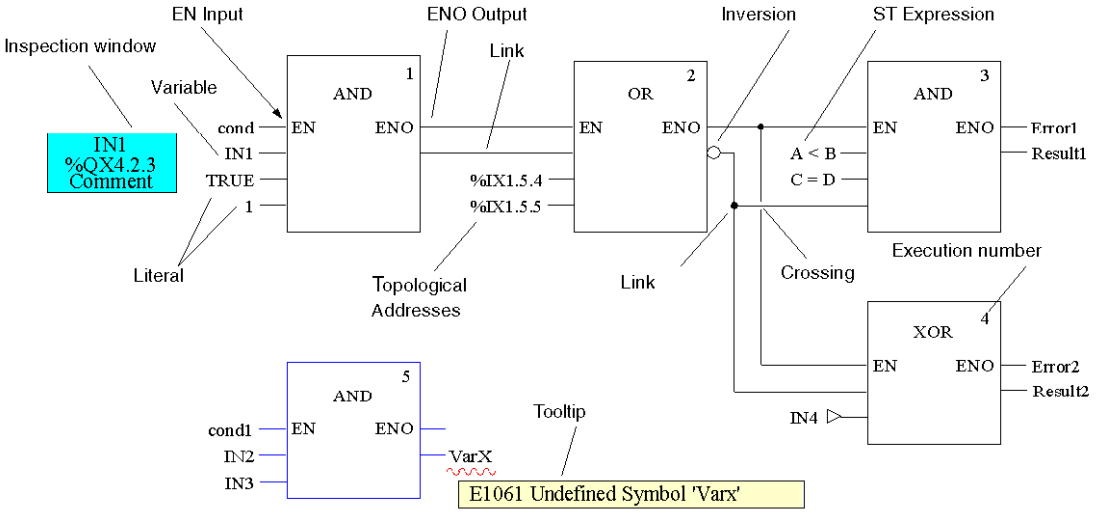
Function Block Diagram FBD

Introduction

The FBD editor is used for graphical function block programming according to IEC 61131-3.

Representation

Representation of an FBD section:



Objects

The objects of the FBD (Function Block Diagram) programming language help to divide a section into a number of:

- Elementary Functions (EFs),
- Elementary Function Blocks (EFBs)
- Derived Function Blocks (DFBs)
- Procedures
- Subroutine calls
- Jumps
- Links
- Actual Parameters
- Text objects to comment on the logic

Properties

FBD sections have a grid behind them. A grid unit consists of 10 coordinates. A grid unit is the smallest possible space between 2 objects in an FBD section.

The FBD programming language is not cell oriented but the objects are still aligned with the grid coordinates.

An FBD section can be configured in number of cells (horizontal grid coordinates and vertical grid coordinates).

The program can be entered using the mouse or the keyboard.

Input Aids

The FBD editor offers the following input aids:

- Toolbars for quick and easy access to the desired objects
- Syntax and semantics are checked as the program is being written.
  - Incorrect functions and function blocks are displayed in blue
  - Unknown words (e.g. undeclared variables) or unsuitable data types are marked with a red wavy line
  - Brief description of errors in the Quickinfo (Tooltip)
- Information for variables and pins can be displayed in a Quickinfo (Tooltip)
  - type, name, address and comment of a variable/expression
  - type, name and comment of an FFB pin
- Tabular display of FFBs
- Actual parameters can be entered and displayed as symbols or topological addresses
- Different zoom factors
- Tracking of links
- Optimization of link routes
- Display of inspection windows

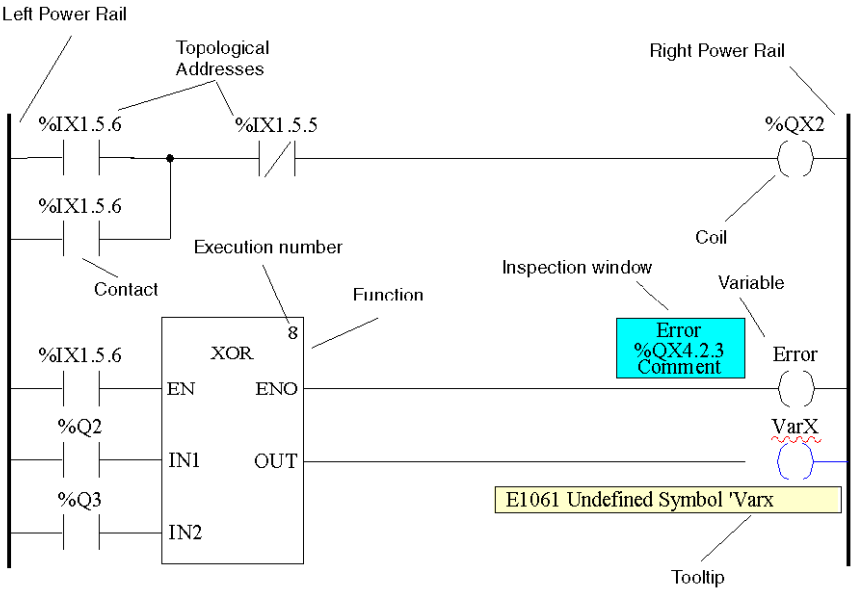
Ladder Diagram (LD) Language

Introduction

The LD editor is used for graphical ladder diagram programming according to IEC 61131-3.

Representation

Representation of an LD section:



Objects

The objects of the LD programming language help to divide a section into a number of:

- Contacts,
- Coils,
- Elementary Functions (EFs)
- Elementary Function Blocks (EFBs),
- Derived Function Blocks (DFBs)

- Procedures
- Control elements
- Operation and compare blocks which represent an extension to IEC 61131-3
- Subroutine calls
- Jumps
- Links
- Actual Parameters
- Text objects to comment on the logic

Properties

LD sections have a background grid that divides the section into lines and columns.

The LD programming language is cell oriented, i.e. only one object can be placed in each cell.

LD sections can be 11-63 columns and 17-3998 lines in size.

The program can be entered using the mouse or the keyboard.

Input Aids

The LD editor offers the following input aids:

- Objects can be selected from the toolbar, the menu or directly using shortcut keys
- Syntax and semantics are checked as the program is being written.
  - Incorrect objects are displayed in blue
  - Unknown words (e.g. undeclared variables) or unsuitable data types are marked with a red wavy line
  - Brief description of errors in the Quickinfo (Tooltip)
- Information for variables and for elements of an LD section, that can be connected to a variable (pins, contacts, coils, operation and compare blocks), can be displayed in a Quickinfo (Tooltip)
  - type, name, address and comment of a variable/expression
  - type, name and comment of FFB pins, contacts etc.
- Tabular display of FFBs
- Actual parameters can be entered and displayed as symbols or topological addresses

- Different zoom factors
- Tracking of FFB links
- Optimizing the link routes of FFB links
- Display of inspection windows

General Information about SFC Sequence Language

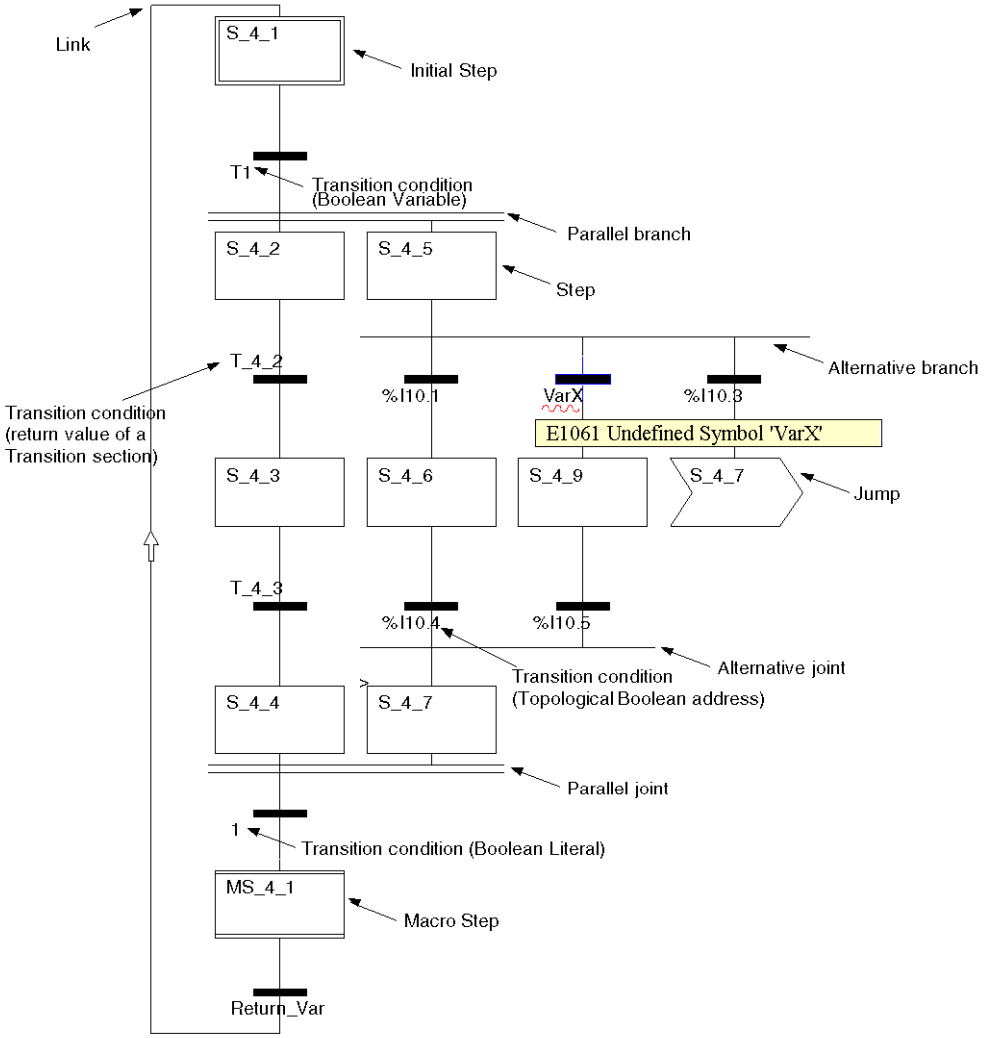
Introduction

The sequence language SFC (Sequential Function Chart), which conforms to IEC 61131-3, is described in this section.

IEC conformity restrictions can be lifted through explicit enable procedures. Features such as multi token, multiple initial steps, jumps to and from parallel strings etc. are then possible.

Representation

Representation of an SFC section:



Objects

An SFC section provides the following objects for creating a program:

- Steps
- Macro steps (embedded sub-step sequences)
- Transitions (transition conditions)
- Transition sections
- Action sections
- Jumps
- Links
- Alternative sequences
- Parallel sequences
- Text objects to comment on the logic

Properties

The SFC editor has a background grid that divides the section into 200 rows and 64 columns.

The program can be entered using the mouse or the keyboard.

Input Aids

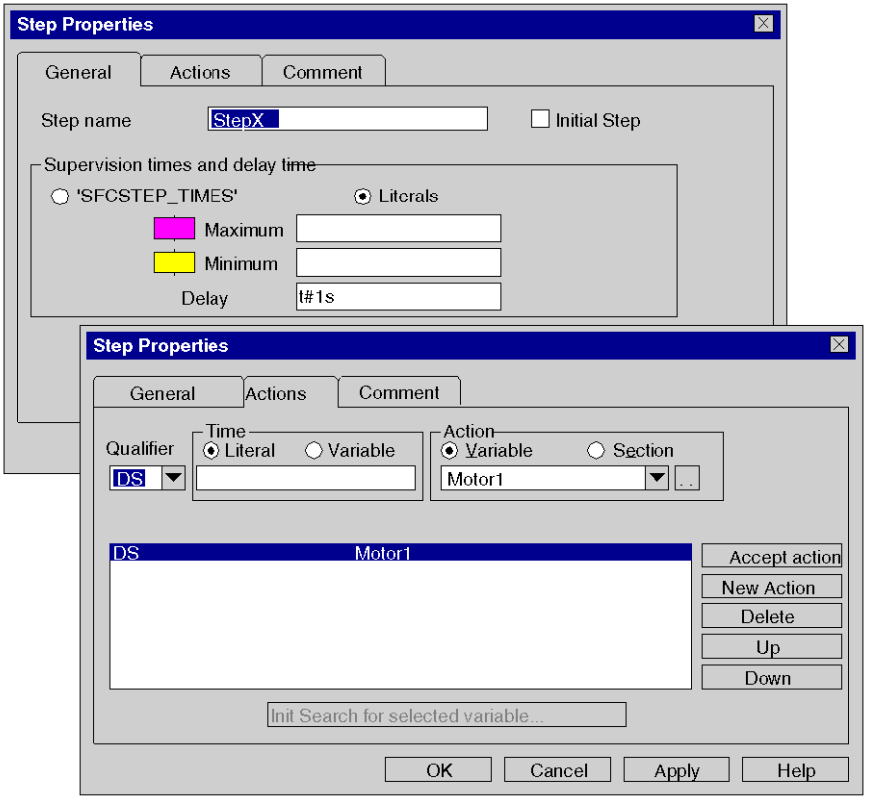
The SFC editor offers the following input aids:

- Toolbars for quick and easy access to the desired objects
- Automatic step numbering
- Direct access to actions and transition conditions
- Syntax and semantics are checked as the program is being written.
  - Incorrect objects are displayed in blue
  - Unknown words (e.g. undeclared variables) or unsuitable data types are marked with a red wavy line
  - Brief description of errors in the Quickinfo (Tooltip)
- Information for variables and for transitions can be displayed in a Quickinfo (Tooltip)
  - type, name, address and comment of a variable/expression
  - type, name and comment of transitions
- Different zoom factors
- Show/hide the allocated actions

- Tracking of links
- Optimization of link routes

Step Properties

Step properties:



The step properties are defined using a dialog box that offers the following features:

- Definition of initial steps
- Definition of diagnostics times
- Step comments
- Allocation of actions and their qualifiers

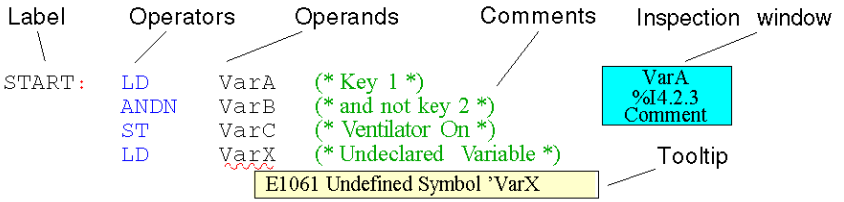
Instruction List IL

Introduction

The IL editor is used for instruction list programming according to IEC 61131-3.

Representation

Representation of an IL section:



Objects

An instruction list is composed of a series of instructions.

Each instruction begins on a new line and consists of:

- An operator
- A modifier if required
- One or more operands if required
- A label as a jump target if required
- A comment about the logic if required.

Input Aids

The IL editor offers the following input aids:

- Syntax and semantics are checked as the program is being written.
  - Keywords and comments are displayed in color
  - Unknown words (e.g. undeclared variables) or unsuitable data types are marked with a red wavy line
  - Brief description of errors in the Quickinfo (Tooltip)
- Tabular display of the functions and function blocks
- Input assistance for functions and function blocks
- Operands can be entered and displayed as symbols or topological addresses
- Display of inspection windows

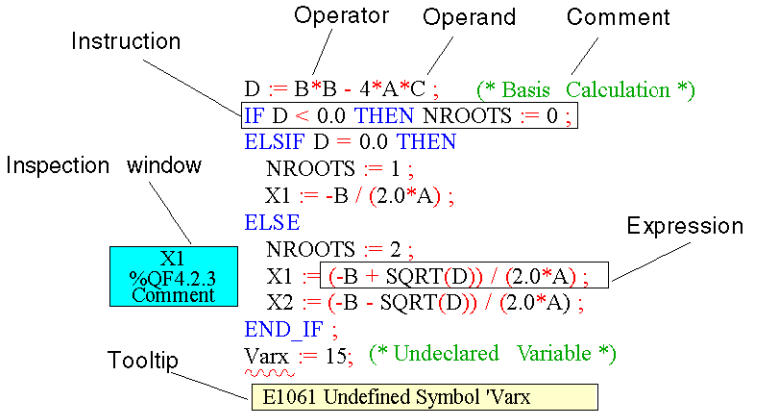
Structured Text ST

Introduction

The ST editor is used for programming in structured text according to IEC 61131-3.

Representation

Representation of an ST section:



Objects

The ST programming language works with "Expressions".

Expressions are constructions consisting of operators and operands that return a value when executed.

Operators are symbols representing the operations to be executed.

Operators are used for operands. Operands are variables, literals, function and function block inputs/outputs etc.

Instructions are used to structure and control the expressions.

Input Aids

The ST editor offers the following input aids:

- Syntax and semantics are checked as the program is being written.
  - Keywords and comments are displayed in color
  - Unknown words (e.g. undeclared variables) or unsuitable data types are marked with a red wavy line
  - Brief description of errors in the Quickinfo (Tooltip)
- Tabular display of the functions and function blocks
- Input assistance for functions and function blocks
- Operands can be entered and displayed as symbols or topological addresses
- Display of inspection windows

Connecting a terminal to a PLC

What's in This Chapter

Methodology for connecting from a PC to a PLC.....68

Subject of this Chapter

This chapter describes how to connect a PC to a PLC.

Methodology for connecting from a PC to a PLC

At a Glance

To program a Modicon M340, Modicon M580, Premium, or Quantum PLC using Control Expert software, observe the rules described below.

Cables References

Select a cable that conforms to your PLC type to physically connect your PLC to your computer:

Platform	Produce	Product Reference
Quantum	Modbus cable	990 NAA 263 20
	Modbus Plus cable	UNY XCA 42 020
	crossed Ethernet cable	490 NTC 000 05(1)
	USB cable	UNY XCA USB 033(2)
Premium	terminal port cable	TSX PCX 1031 (RS 232/RS 485)
	terminal port cable	TSX PCX 3030 (USB/RS 485)
	crossed Ethernet cable	490 NTC 000 05(1)
	USB cable	UNY XCA USB 033(2)
Modicon M340	Modbus cable	TCS MCN 3M4F3C2
	crossed Ethernet cable	490 NTC 000 05(1)

Platform	Produce	Product Reference
	USB cable	BMX XCA USBH •••(2)
Modicon M580	mini-USB to USB-A programming cable (industrial use rated)	BMX XCA USBH •••(2)
	Ethernet straight-through Cat5E cable, CE or UL standard (industrial use rated)	490 NTW 000 ••(U)
	Ethernet straight-through Cat5E cable, CE or UL standard (industrial use/high demand rating)	TCSEC(E/U)3M3M••S4
	Ethernet straight-through Cat5E cable, 1 meter	TCSECN3M3M1S4
	Ethernet crossed Cat5E cable (industrial use rated), CE or UL standard. (3)	490 NTC 000 ••(U)

(1) Use a crossed cable for a PC-PLC link; whereas, use a straight-through cable for PC-Hub or PLC-Hub link.

(2) When connecting a PC to a PLC, the use of a USB 2.0 shielded cable following the USB international standard is strongly recommended. The cables mentioned earlier avoid unexpected behavior of the PLC as they are shielded and tested against electrical noises.

(3) This cable is usable but not recommended as all ports of a M580 system are Auto-MDIX capable and crossed cables could be incompatible with future gigabit systems.

Procedure

The table below describes how to connect from a PC to a PLC.

Step	Action
1	Physically connect your PLC to your computer using the appropriate cables, page 68.
2	Check that for the chosen communication protocol, the Driver corresponding to your operating system is correctly installed. **Note**: The documentation on drivers (see Communication Drivers, Installation Manual) describes the various drivers available according to the operating system. For example, for the PC to be able to use an X-Way address on an Ethernet network, you must install the XIP driver.
3	Verify in the toolbar that the simulator icon is unchecked.
4	Choose the address of the PLC, page 93 to which you want to connect and the type of link, page 99 using the **PLC > Set Address...** command.
5	Connect by selecting the command: **PLC > Connection** **Result**: You can now perform all the operations available on line, set the PLC to RUN or to STOP, download projects, or make program modifications, etc.

Global project management

What's in This Chapter

Access security management 71
Project management 72

Subject of this Chapter

This chapter introduces the various tools used for project management: security and user access rights management, project creation, project backup, transfer of the project to the PLC, management of the user memory, etc.

Access security management

Subject of this Section

This section introduces the tools used to manage security for a project.

Access security management

Security Editor Tool

Refer to *EcoStruxure™ Control Expert, Security Editor, Operation Guide* to manage security for a project.

Project management

Subject of this Section

This section introduces the tools used for project management.

Access to Control Expert Software

At a Glance

It is possible to launch several instances of Control Expert simultaneously on the same workstation. However, the access method is identical for all instances and depends on whether the security option has been enabled:

- If security is active, clicking the Control Expert icon calls up a login dialog box which is used to access Control Expert.
- If security is inactive, you have direct access to Control Expert.

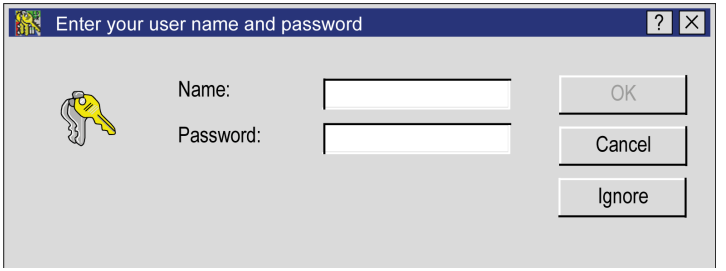
Security Inactive or Active in No Login Mode

If security is inactive, execute the following commands to launch an instance of Control Expert:

Step	Action
1	Click **Start** > **Programs**.
2	Select the group **EcoStruxure Control Expert**. **Result**: A list appears • Language selection • OS Loader • PLC Simulator • Readme • Registration Tool • Types library update • Control Expert
3	From the list, select **Control Expert**. **Result**: Control Expert is launched in the selected language. Special case: If you have not yet activated the software, a message warns you of the number of days of use left, and asks you whether you want to activate. For more information, refer to *EcoStruxure™ Control Expert, Installation Manual*.

Other Cases

When security is active, page 71, execute the following commands to launch an instance of Control Expert:

Step	Action
1	Activate the Control Expert icon: see procedure below. **Result**: The dialog box **Enter your user name and password** is displayed. 
2	Enter your name and password (if necessary).
3	Confirm with **OK**. **NOTE:** If you click **Ignore** Control Expert starts with access setting of user default defined in Security Editor **Policies**.

Services in Offline Mode

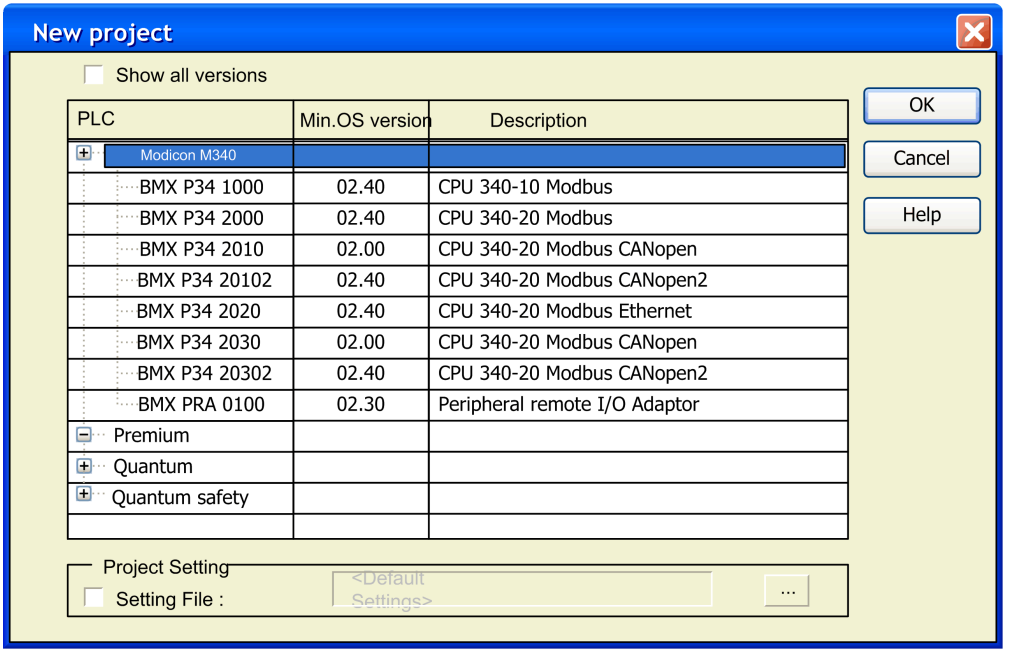
At a Glance

The following operations can be carried out when accessing Control Expert software in off-line mode:

- Creating,
- Opening,
- Converting,
- Analyzing,
- Generating,
- Saving,
- Closing.

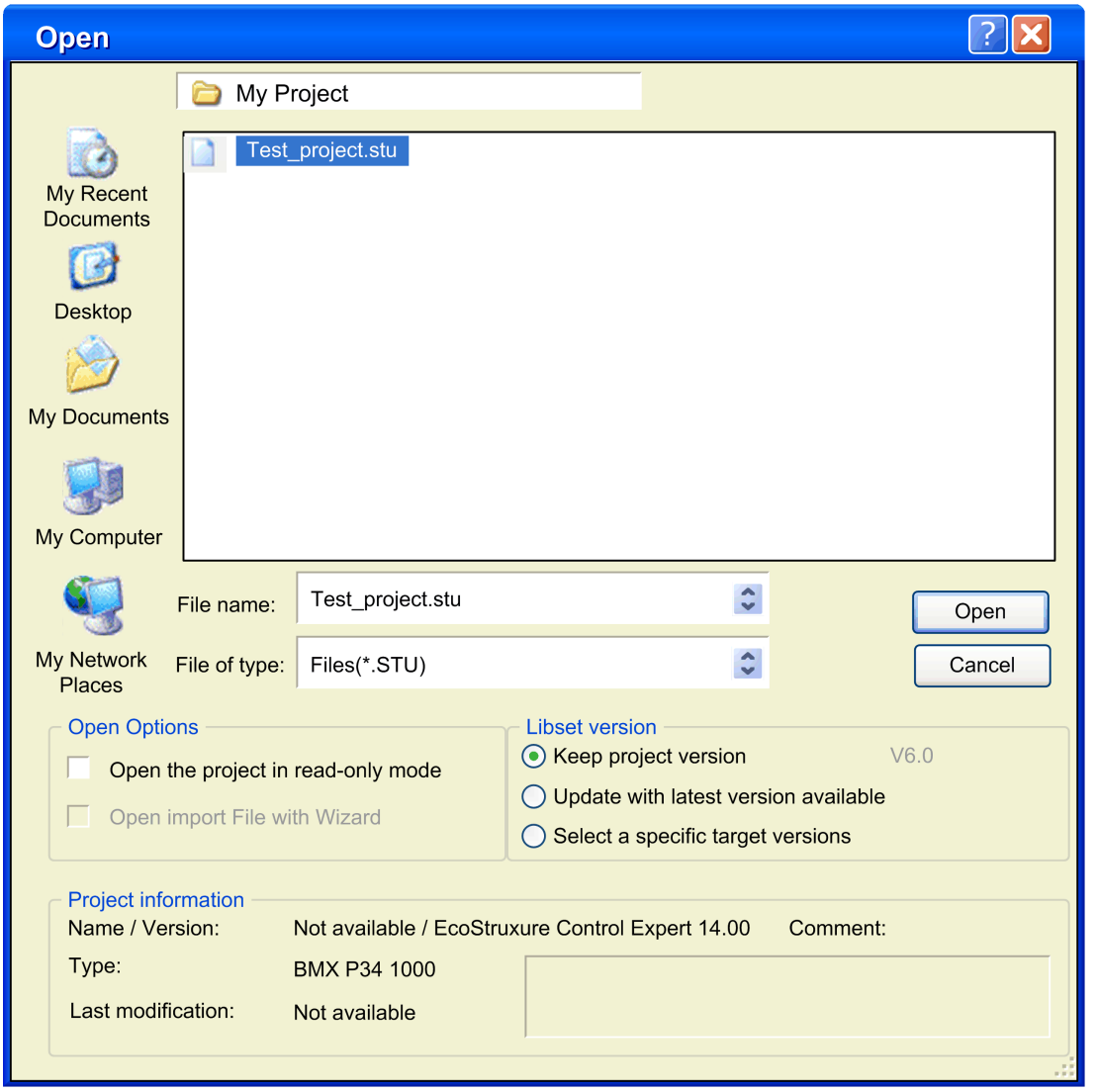
Creating a Project

To create a new project, carry out the following actions:

Step	Action
1	Activate the **New** command in the **File** menu. **Result**: The **New Project** screen appears. 
2	To see all PLC versions, check **Show all versions** box.
3	Select the processor you wish to use from those proposed.
4	To create a project with specific values of project settings, check **Settings File** box and use the browser button to localize the **.XSO** file (Project Settings file). It is also possible to create a new one. If the **Settings File** box is not checked, default values of project settings are used.
5	Confirm with **OK**.

Open Dialog

Open project dialog



Element	Description
Look in **File name** **Files of type**	The upper part of this dialog displays a standard Windows **File Open** dialog.
Opening a legacy LL984 project	Please refer to Importing an LL984 Application, page 107.
Open Options	
Open the project in read-only mode	Opens the project in read-only mode.
Open import File with Wizard	Select this check box to use the import, page 1686 wizard.
Libset version **Libset** (set of libraries) contains all libraries, families, functions and variables (I/O data structures) that can be used to develop an automation project. In general, to open an existing project (**.STU** file) or an archived project (**.STA** file) the Libset version can be selected. To import a **.ZEF** or **.XEF** file, the latest Libset version must be selected. More information on the default **Libset** version used to open a project according to the file type is provided in the libset, page 315 section.	
Keep project version	Select this option to keep the **Libset** version used at file creation time. If the respective **Libset** is not installed, the latest **Libset** version will be used.
Update with latest version available	Select this option to update the project with the latest **Libset** version.
Select a specific target version	Select this option to choose a specific **Libset** version. The **Version** box can be used to select a version.
Project information	
Type	Target PLC.
Name / Version	Project name and software package version.
Last modification	Date and time of last modification of the project.
Comment	Comment associated to the project.

Opening a Project

To open an existing project, carry out the following actions:

Step	Action
1	Choose the **Open** command from the **File** menu.
2	Select the file related to the project (file ***.STU**).
3	Select one of the **Libset version** options.
4	Confirm with **Open**.

Opening a project also restores the Control Expert context, which was saved when the project was closed. This involves:

- The list of open tools,
- For each tool: its window size, position, contents and animation status,
- taskbar and tool configuration.

The context file is a companion file to the <Project_name>.STU file. It is saved in the same directory with the name <Project_name>.ZTX. The **.ZTX** file is not required to open an application. If no **.ZTX** is provided the application is open with default context.

NOTE: When a project file is selected, the software provides a certain amount of information: the project name, associated comment, version and date the project was generated, target PLC, and date of last modification to the source code.

NOTE: The **.STU** file is a work file and it can not be used to switch from one version to another of Control Expert software. In this case, you should use the archive format (**.STA** file) or use the export function in the project and create a **.ZEF** or **.XEF** file.

NOTE: The **.STU** file contains the configuration of the TSX CPP 110 in the **Project Browser**.

A **The file not found** message appears each time the configuration screen of the TSX CPP 110 PCMCIA card is opened. Select the required **.CO** file to update the CANopen configuration.

Opening an Archived Project

To open a project archived in STA format, carry out the following actions:

Step	Action
1	Choose the **Open** command from the **File** menu.
2	Select the file related to the project (file ***.STA**).
3	Select one of the **Libset version** options.
4	Confirm with **Open**.

An STA file is a project archive file that can be accessed using the **Archive** command in the **File** menu.

The properties of the **.STA** file are the following:

- The **.STA** file is very compressed (approximately 50 times more so than the STU). It is used to transfer projects to networks (local or Internet, for example).
- The **.STA** file can be used to transfer projects between versions of Control Expert software.
- The **.STA** file contains the entire project:
  - The PLC binary,
  - The Upload information: comments and animation tables,
  - The operator screens.

  NOTE: When a **.STA** file is selected, the software provides a certain amount of information:
  - The project name,
  - Associated comment,
  - Version and date the project was generated,
  - Target PLC,
  - Date of last modification to the source code,
  - The version of Control Expert used to create this archive.

  NOTE: As the file is compressed, loading takes much longer than for an **.STU** file. We recommend using the **.STA** file to archive the project and the **.STU** file as the actual work file.

  NOTE: To Fully retrieve an application with Quantum Ethernet I/O system and/or 140 NOC 78• 00 topologies, The **Upload information** or **Open** an archive files must be used.

Converting an Existing PL7 or Concept Project

To convert an existing PL7 or Concept project, carry out the following actions:

Step	Action
1	Choose the **Open** command from the **File** menu.
2	Select the file related to the project: • ***.FEF** or ***.DFB**, to launch the conversion procedure for a PL7 application (see EcoStruxure™ Control Expert, PL7 Application Converter, User Manual), • ***.ASC**, to launch the conversion procedure for a Concept application (see EcoStruxure™ Control Expert, Concept Application Converter, User Manual).
3	Confirm with **Open**.

Conversion Wizard

To convert a legacy project partially or as a whole the **Conversion Wizard** can be used.

The **Conversion Wizard** provides options to remap I/O objects (channels, variables etc.) during conversion and to adapt the hardware configuration of the new project concurrently in Control Expert.

Shortcuts to Recent Projects

To simplify access to the current project(s), the software provides a shortcut for accessing the last 8 projects used. In the File menu, activate the shortcut to the project you would like to open.

Archiving a Project

To archive a project in STA format, carry out the following actions:

Step	Action
1	Select the **Save Archive** command from the **File** menu.
2	If necessary, select the directory where the project will be stored (disk and path).
3	Enter the file name.
4	Confirm with **Save**.

NOTE: Archiving is only possible after a **Rebuild all project** with the Upload information, page 551 section of the **Project Settings** window set as follows:

- the **Upload information** and **Comments** check boxes are checked, with the **Animation tables** check box unchecked
- the **Upload information**, **Comments** and **Animation tables** check boxes are checked
- the **Upload information**, **Comments** and **Animation tables** check boxes are unchecked

NOTE: The .STA file partially contains the TSX CPP 110 configuration. Do not forget to backup the **.CO** file (Sycon) for CANopen after archiving of the project.

Analyzing a Project

To analyze the syntax of your project, carry out the following actions:

Step	Action
1	Activate the **Analyze project** command in the **Build** menu. **Result**: the software analyzes the project.
2	Any errors detected are displayed in the information window at the bottom of your screen.

Analyzing and Generating a Project

To analyze and generate your project at the same time (generation of links between input/output modules and declared objects in the project, etc.), carry out the following actions:

Step	Action
1	Activate the **Rebuild all Project** command in the **Build** menu. **Result**: the software analyzes and generates the project.
2	Any errors detected are displayed in the information window at the bottom of your screen.

Analyzing and Generating Modified Project Parts

To analyze and generate **only** the modified parts of your project, carry out the following actions:

Step	Action
1	Activate the **Build Changes** command in the **Build** menu. **Result**: the software analyzes and generates the modified parts of your project.
2	Any errors detected are displayed in the information window at the bottom of your screen.

NOTE: The **Build Changes** analyses and displays only modified sections or sections impacted by modifications. To display all warnings, a **Rebuild all Project** command is needed.

NOTE: The **Build Changes** command can be executed only if the **Rebuild all Project** command has been executed at least once previously.

Backing up a New Project

To back up a new project, carry out the following actions:

Step	Action
1	Activate the **Save** or **Save As** command in the **File** menu.
2	If necessary, select the directory where the project will be stored (disk and path).
3	Enter the file name.
4	Confirm with **Save**.

Backing up an Existing Project

To back up your modifications, select the Save command from the File menu.

Closing a Project

To close your project, carry out the following actions:

Step	Action
1	Activate the **Close** command in the **File** menu.
2	If you have modified your project, a dialog box is displayed with the message that you must save your project.
3	Save your project or close it without saving changes.

NOTE: Closing a project triggers saving the current context in the file <Project_name>.ZTX, page 77.

Exiting Control Expert

To exit Control Expert, carry out the following actions:

Step	Action
1	Activate the **Exit** command in the **File** menu.
2	If you have modified your project, a dialog box is displayed with the message that you must save your project.
3	Save your project or exit it without saving changes.

Services in Online Mode

Programming and monitoring modes

You can access the project in the PLC in Online Mode in the following two ways:

- **Programming**, the PLC is reserved by the terminal; it is possible to access all the functions of the software. It is not possible to connect in programming mode to a PLC that is already reserved.
- **Monitoring**, The PLC is not reserved by the terminal; it is not possible to modify functions of the PLC, (the program), however it is possible to modify values of the variable with the animation table.It is also possible to connect in monitoring mode to a PLC that is already reserved. Links between EFs are not animated.

Note: The command **Tools > Options...Connection** tab lets you choose the default operating mode, page 600.

Equal or Different Modes

On connection, the data in the PLC and the terminal is compared, page 551:

- if the data is identical, this determines an equal online mode, page 92,
- if the data is different, this determines a different online mode, page 92.

Information in the PLC

Two types of information are compared between the PLC and the terminal:

- Executable information is used to identify and execute the project in the PLC:
  - identification of the project,
  - configuration,
  - code sections,
  - DDT types,
  - DFB types,
  - EF and EFB codes,
  - project options.
- Upload information:
  - project upload information: graphic code for languages with LD contacts and chart of FBD function blocks, symbols of static and dynamic variables,
  - comments: variables, code sections, project, DFBs, etc.,
  - animation tables.

User functions

The user functions provided in online mode that can be accessed from the **PLC** menu are the following:

Function	Description
Connection/Disconnection	Connection (change to online mode) or disconnection (change to offline mode).
Set Address	Access to a PLC via the network.
Standard mode	Choice of connection with a PLC.
Simulator mode	Choice of connection with a PLC.
Compare	Comparison of terminal/PLC projects.
Transfer Project to PLC and **Transfer Project from PLC**	Transfer of the program between the terminal and the PLC.
Save Data from PLC to File and **Restore Data from File to PLC**	Transfer of data between a file and the PLC.
Run/Stop and **Init**	Sending a command to the PLC.

Function	Description
Update Upload Information	Update the Upload information in the PLC.
Update Init Values with Current Values	The PLC forces the update of the initial values using current values (this applies to the variables with the save attribute).
Project Backup	Backs up the project in memory.
Memory Consumption	Access to PLC memory usage statistics.
Diagnostics	Access to the debug function.

Cancel/Repeat

The **Cancel/Repeat** commands in the **Edit** menu are not available in the following cases:

- Deletion of a step or macro-step in online mode.
- Switch from **normal view** to **extended view** in SFC debug mode and vice-versa.

Allowed Online Modifications

Overview

The following pages describe modifications that are allowed when Control Expert mode is online.

General Rules Concerning Online Modifications

Online modification is a feature that concerns only the ‘online equal’ state.

If the application present in the PLC is different from the one opened in Control Expert, the state is ‘online different’. One of both applications should be chosen and loaded to be in the ‘online equal’ state.

NOTE: To download the application from Control Expert into the PLC, it is mandatory to stop the PLC. However, the PLC may stay in RUN mode while uploading the application to the PC via Control Expert.

The ‘online equal’ state may only be reached if the connection mode is ‘programming’. To configure the connection mode, use the command **Tool > Options > Connection**.

Two different connection modes are available:

- monitoring, page 83: state is ‘monitoring’ and no online modification is allowed,

- programming, page 83.

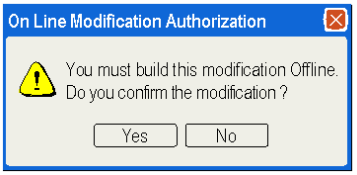
The distinction between executable code and upload information means that there are 2 kinds of modifications:

- those which truly modify the executable information (e.g., which add a new EFB call or a new ST instruction)
- modifications that have an impact only on the Upload information (e.g. concerning comments, graphical information of languages).

Consequently, these kinds of modifications may not need to be built depending on whether upload information is loaded into PLC, comments and animation tables included in upload information.

As a general rule, Control Expert prevents the user from doing actions by mistake that would force them to stop the PLC and/or to perform a complete download:

- either by simply disabling the forbidden functions in online mode
- or by a warning message, as follows:



To know what parts have been modified, the application browser displays a specific mark for the modified parts and in the title bar of the editors by character '*'.

Modify the Program

It is always possible to modify a section of the program in online mode. As soon as a section has been modified, it is no longer animated until the next build and transfer to PLC.

Nevertheless, depending on the kind of modifications, a SFC graph may be reset after the build and PLC update.

Some modifications performed in SFC might need a **rebuild all** command. In such cases, a warning message appears in the editor. For example, deleting an action or a transition section requires a 'rebuild all' command. However, removing an action or a transition section from a step or a transition and keeping the section unused only requires an online build.

NOTE: Adding an SFC section must be done offline, if the two following conditions are fulfilled simultaneously:

- If the project contains at least one SFC section already.
- If the project has been designed with an earlier version than Unity Pro 3.0

  Unity Pro is the former name of Control Expert for version 13.1 or earlier.

This modification requires a **rebuild all** of the project and a PLC stop to download the entire application.

Modify Data

In the data editor, there is no specific operating mode.

Generally, it is allowed to modify the type of a variable or a Function Block (FB) in online mode.

Modifying a data, which is used in the program, may have consequences wherever it is used, on the contrary to an unused variable. Therefore online actions are more restricted on used data.

NOTE: Adding a variable without rebuilding increases the memory usage until next rebuild.

For used EDT variable or used DDT/FB instance, only comments and the initial values can be modified. Unused variable/instance may be deleted. It is allowed to create new variable/instance online.

Concerning DDT type, it is only allowed to change comments if the type is used. It is possible to create new DDT or to delete an unused one.

For DFB type, it is possible to:

- create, modify and delete sections
- change comments
- change initial values
- add a new public or private variable

If there is no instance of the DFB, there is no limitation. The DFB type may even be deleted. Finally creating new DFB type(s) and instantiating it in online mode is straightforward.

Modify the Configuration

For **Modicon M340**, **Premium** and **Atrium** PLCs, it is not possible to add or remove any module in online mode. Parameters of an existing module on the contrary may be configured online. Once modified, the animated mode is no longer available until the next build and PLC update.

For **Modicon Quantum** system PLCs, it is possible to add or delete a selection of modules and change the parameters of an existing module in STOP and RUN mode. For more information refer to *Modicon Quantum, Change Configuration on the Fly, User Guide*.

For **Modicon M580** PLCs when the option CCTOF is selected, it is possible to add or delete a selection of modules and change the parameters of an existing module in STOP and RUN mode. For more information refer to *Modicon M580, Change Configuration on the Fly, User Guide*.

Comment Concerning Offline Modifications

Modifications that are only possible offline and new application downloads require an offline build and a rebuild all. In these cases, the modification is not possible online and a message notifies the user.

The rebuild is needed:

- After application creation or import
- Once the application is already built:
  - Changing some project settings
  - CPU change
  - Memory card change
  - Memory card layout change
  - Switching from real target to simulator and vice versa

Furthermore, if a flash PCMCIA card without additional SRAM (TSX MFPP •••M or TSX MFPP •••K) is configured, no modification is allowed online. User is informed by a specific message ‘This action is forbidden because: Flash Cartridge’.

Synthesis of Authorized and Unauthorized Modifications

The following table presents the main authorized and unauthorized online modifications. The second column indicates which modifications have impact only to the upload information and do not have any impact to the executable code:

Type of Modification	Impacts Only Upload Information	Allowed in Online Mode (RUN/STOP)	Requires to Build the Application in Offline Mode
General			
Name of Station, Programs, Sections	Yes	Yes	No

Type of Modification	Impacts Only Upload Information	Allowed in Online Mode (RUN/STOP)	Requires to Build the Application in Offline Mode
Comment of Station, Configuration, Programs, Sections	Yes	Yes	No
Documentation Summary	Yes	Yes	No
Animation Table	Yes	Yes	No
Integrated Operator Screens	Yes	Yes	No
Functional View	Yes	Yes	No
Security Information: Passwords, Protection Attributes,...	No	Yes	No
Switch Simulator <-> Real Target	No	No	Yes
Import/Export Project	No	Yes	No
Program			
Program elements (Sections, Program Units): Add, Change Execution Order	No	Yes	No
Modify the Code of a Section (Task Section, SR, Transition, Action, DFB Sections)	Yes(1)	Yes	No
Modify the Code of EVT Sections	No	Yes(2)	Yes
Modify the Code of SFC Chart	No	Yes	No(1)
Configuration/ Communication			
Premium Configuration: Add/ Delete a Module	No	No	Yes
Premium Configuration: Change Module Parameters	No	Yes	No
Quantum Configuration: Add/ Delete a Module	No	Yes	No
Quantum Configuration: Change Module Parameters	No	Yes	No
Changing Memory Sizes in Configuration Screen	No	No	Yes

Type of Modification	Impacts Only Upload Information	Allowed in Online Mode (RUN/STOP)	Requires to Build the Application in Offline Mode
Global Variables (Used Even in Animation Table or Operator Screen)			
Remove an Used Variable	No	No	Yes
Symbol an Used Variable	No	Yes	No
Type an Used Variable	No	No	Yes
Topological Address of an Used Variable	No	Yes	No
Initial Value of an Used Variable	No	Yes	No
Comment of an Used Variable	Yes	Yes	No
Create, Remove or Modify Unused Variables (EDT, DDT)	No	Yes	No
Create, Remove or Modify Unused Variables (FB)	No	Yes	No
Program Unit Variables (Used Even in Animation Table)			
Remove an Used Variable	No	No	Yes
Symbol an Used Variable	No	Yes	No
Type an Used Variable	No	No	Yes
Topological Address of an Used Variable	No	Yes	No
Initial Value of an Used Variable	No	Yes	No
Comment of an Used Variable	Yes	Yes	No
Effective Parameter of an Used Variable	Yes	Yes	No
Create, Remove or Modify Unused Variables (EDT, DDT)	No	Yes	No
Create, Remove or Modify Unused Variables (FB)	No	Yes	No
Used DFB			
Type Name of Used DFB	No	No	Yes

Type of Modification	Impacts Only Upload Information	Allowed in Online Mode (RUN/STOP)	Requires to Build the Application in Offline Mode
All Comments	Yes	Yes	No
Add a Parameter	No	No	Yes
Add a Public or Private Variable	No	Yes	No
Delete or Change Unused Private Variable	No	Yes	No
Initial Value of Parameters and Variables	No	Yes	No
Other Attributes of Parameters and Variables (Public and Private)	No	No	Yes
Modify the Code of a section	No	Yes	No
Create a New DFB Type	No	Yes	No
Delete an Unused DFB Type	No	Yes	No
Used DDT			
Create a New DDT Type	No	Yes	No
Delete an Unused DDT Type	No	Yes	No
(1) Modifying only the graphical layout or the free form comments does not impact code generation, but only generation of the upload information. **(2)** This is only available in the RUN mode for the Quantum modules.			

Connection and Disconnection

At a Glance

Connection enables you to change from offline mode to online mode. On connection, data in the controller and the terminal is compared:

- If the information is identical, the online mode is Equal.
- If the information is different, the online mode is Different.

By disconnecting, you can change from online mode to offline mode.

Equal Online Mode

On connection to the controller, Control Expert runs in Equal online mode when executable files and the upload information are identical in the controller and in the terminal.

You can modify the project in the same way as in offline mode. Generating the executable also automatically triggers loading of modifications to the controller.

If the project program has been modified in the terminal without regenerating the executable, only the non-modified parts of the project are animated.

To animate the entire project again, generate the executable and update the controller.

Different Online Mode

On connection to the controller, Control Expert runs in Different online mode when executable files and the upload information are different in the controller and in the terminal.

To run in Equal online mode, transfer the project between the terminal and the controller.

In Different online mode, the animation level is limited to access to located variables through their topological addresses. Modification of the program is not authorized.

Connect the Controller

Step	Action
1	Upon the first connection, open the **Set Address** dialog box, page 93 (**PLC > Set Address**).
2	Configure a connection in either the **PLC** or **Simulator** group box.
3	Make the connection (**PLC > Connect**).

Disconnect the Controller

Step	Action
1	Click **PLC > Disconnection**.

Automatic Disconnection

Increase the Timeout communication parameter, page 102 to help avoid an automatic switch from online mode to offline mode (which is possible if the user application execution time is greater than the timeout communication time).

Access a Controller Over a Network

Controller Access

Access a new controller over the network:

Step	Action
1	Open the **Set Address** dialog box (**PLC > Set Address**).
2	In the **PLC** group box, enter the address of the new controller in the **Address** field.
3	In the **PLC** group box, use the **Communication Protocol** pull-down menu to select a communication type, page 98. **NOTE:** Ethway/IP drivers are not available in the **Set Address** dialog box. To use this type of communication, manually enter *ETHWAY01* or *ETHWAY02* in the **Communication Protocol** pull-down menu.
4	Make the desired modifications: • Click the **Communication Parameters** buttons in the **PLC** and/or **Simulator** group boxes to modify the communication parameters, page 102. **NOTE:** The simulator starts on port number 502 by default. To change the port number, enter an unused port number in the **Simulator Address** box. (For example, address 127.0.0.1:511 corresponds to port number 511.) • Click the **Bandwidth** button to modify the connection bandwidth, page 103.
5	Click the **Test Connection** button to test the connection.
6	Click the **OK** button to confirm the modifications.

Set Address Parameters

These parameters appear in the **Set Address** dialog box:

Group box	Parameter	Description
PLC	**Address**	Controller address (by default, the address of the controller physically wired to the computer). The drop-down menu lists any previous addresses entered. Click the trash can icon to clear the drop-down list.
	Communication Protocol	This drop-down selects the type of communication with the network for the controller.
	Communication Parameters	By clicking this button you access the controller communication parameters, page 102.
Simulator	**Address**	Simulator address The drop-down menu lists any previous addresses entered. Click the trash can icon to clear the drop-down list.
	Communication Protocol	This drop-down selects the type of communication with the network for the simulator.
	Communication Parameters	By clicking this button, you access the simulator communication parameters, page 102.
(buttons)	**Bandwidth**	Click this button to display the **Online Mode Services Bandwidth** dialog box, page 103.
	Test Connection	Click this button to execute a connection test without closing the **Set address** dialog box.
(checkbox)	**Speed rate auto-adaptation at the end of download**	Select (check) this box to use this mode for applications that do not require a specific speed rate (such as modems).

Network Address Syntax

The network address syntax is documented in the *Communication Services and Architectures, Reference Manual*.

This table describes the various authorized addresses according to the platforms and media supports:

Platform	Support		Address syntax
Premium	Uni-Telway	SCP01	X Way
	Uni-Telway	UNTLW01	X Way Example: UNTLW01 0.2.1.4 access to the Uni-Telway slave at address 4 which is connected to rack 0, module 2, channel 1 of the local controller.
	Fipway	FIP01, FIP02, FPP2001 (driver for the TSX FPP20 card)	X Way
	Fipio	FIP01, FIP02, FPP2001	X Way Example: FIP02 \2.55\SYS to access to FIPIO connection point 55 through the privileged terminal Fipio (address 63 for Premium) Example: FIP01 SYS to access to Fipio Master through the privileged terminal Fipio (address 63 for Premium).
	PCIway	PCIWAY01, PCIWAY02	X Way
	Ethway	ETHWAY01, ETHWAY02	X Way
	XIP	XIP01 to XIP05	X Way Example: XIP01 {3.5}\2.55\SYS to access a FIPIO connection point 55, through network 3 and station 5 controller. Example: XIP02 {3.5}0.2.0.4 access to a Uni-Telway slave connection at address 4 which is connected on the rack 0 module 2 channel 0 of network 3 and station 5 controller.
	EtherNet/IP	TCP/IP	IP address
	EtherNet/IP	TCP/IP**	IP address; Index or "DNS machine name; Index"
	USB	USB	implicit; no address is necessary as the connection is point to point. However, it is possible to enter an X Way address when the controller acts as an X Way router
	Modbus Plus	MBPLUS01 to MBPLUS04	Modbus Plus Address Example: MBPLUS01 2.1 access to Modbus Plus node address 1 which is connected after the Modus Bridge address 2.

Platform	Support		Address syntax
Quantum	Serial Modbus	MODBUS01 to MODBUS04	Modbus Slave Number
	Modbus Plus	MBPLUS01 to MBPLUS04	Modbus Plus Address
	Ethernet	TCP/IP	IP Address Example: TCPIP 84.0.127.6 access to an Ethernet TCP/IP station.
	EtherNet/IP	TCP/IP***	IP address; Index or "DNS machine name; Index"
	USB	USB	implicit; no address is necessary as the connection is point to point
Modicon M340	Serial Modbus	MODBUS01 to MODBUS04	Direct Address: Modbus slave number Remote address: Link_address.Modbus Slave number Link_address = r.m.c. of Modbus link Direct address example: 66 Bridging example: • 66\\0.0.3{118.159.35.45} for Modbus slave 66 to the controller on Ethernet. • 66\\0.3.0{118.159.35.45} for Modbus slave 66 to the controller on the Ethernet NOE in slot 3.
	Ethernet	TCP/IP	Direct address: IP Address Remote address: Netlink {IP address} Netlink address = r.m.c. of Ethernet link Direct address example: 84.0.127.6 Bridging example: • 139.160.235.20\\0.3.0{118.159.35.45} for Ethernet controller to the controller on the Ethernet NOE in slot 3. • 139.159.35.20\\0.0.0.66 for Ethernet to Modbus slave 66.
	EtherNet/IP	TCP/IP**	IP address; Index or "DNS machine name; Index"
	USB	USB	SYS or empty Bridging example: SYS\\0.0.0.66 for USB to Modbus slave 66.

Platform	Support		Address syntax
Modicon M580	Ethernet	TCP/IP HTTPS Modbus TCP	Direct address: IP Address Remote address: Netlink {IP address} Netlink address = r.m.c. of Ethernet link Direct address example: 84.0.127.6 Bridging example: • 139.160.235.20\\0.3.0{118.159.35.45} for Ethernet controller to the controller on the Ethernet BMXNOR in slot 3. • 139.159.35.20\\0.4.0.66 for Ethernet to Modbus slave 66 of channel 0 of a BMXNOM0200 in slot 4.
	EtherNet/IP	TCP/IP**	IP address; Index or "DNS machine name; Index"
	USB	USB Modbus TCP via USB HTTPS TCP via USB	SYS or empty Bridging example: SYS\\0.4.0.66 for USB to Modbus slave 66 of channel 0 of a BMXNOM0200 in slot 4.
Simulator	Ethernet	TCP/IP Modbus TCP	IP address; Index ** "localhost" "DNS machine name; Index" **

**: "Index" is the Modbus Plus or Modbus index stipulated in the bridge that you want to cross. This access works only in display mode: do not use it to modify the program or to force values.

***: "Index" is the Modbus Plus or Modbus index stipulated in the bridge that you want to cross.

NOTE: Use quotation marks when the address includes characters other than figures, full stops, or semi-colons. Otherwise, the address may be analyzed syntactically.

Access a Controller Beyond a Router

To communicate to several controllers behind routers, configure a network address translation table in EcoStruxure Control Expert to handle different ports. In this case, supply the router address and different ports to reach controllers behind the routers.

Example: For a router address of 139.160.39.12, these controller addresses are local:

Controller Number	Controller Address
1	192.168.0.1:502
2	192.168.0.2:502
3	192.168.0.3:502

The above example shows controllers with different addresses, but using the same port.

Network Address Translation Table: Example of a network address translation table:

Port	Destination
TCP/IP 3501	192.168.0.1:502
TCP/IP 3502	192.168.0.2:502
TCP/IP 3503	192.168.0.3:502

The above example shows the same router address with different ports, each port corresponding to a controller address.

NOTE:

- EcoStruxure Control Expert can communicate with controller 2 using the router address followed by port 3502.
- In the **Set Address** dialog box, manually enter *139.160.39.12:3502*.
- The router sends the message to the address 192.168.0.2:502 (configured in the router in the network address translation table).
- To communicate with other controllers, use the same router address, but change the port number.

Overview of Types of Connection Possible With Schneider Controllers

At a Glance

EcoStruxure Control Expert connects to Modicon M340, Modicon M580, Premium, Atrium, and Quantum controllers in different ways.

Each type of connection has its own characteristics. The following table provides an overview of the different possibilities.

Connection Solutions

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

Do not connect more than one controller over USB on the same computer.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

The following table shows the different solutions that can be used to connect a controller with EcoStruxure Control Expert.

Type of Connection	M340	M580	Premium and Atrium	Quantum
USB	All processors	All processors	On processors: • TSX P57 4634 • TSX P57 554 • TSX P57 5634 • TSX P57 6634	On processors: • 140 CPU 651 50 • 140 CPU 651 60 • 140 CPU 651 60S • 140 CPU 652 60 • 140 CPU 658 60 • 140 CPU 670 60 • 140 CPU 671 60 • 140 CPU 672 60 • 140 CPU 672 61 • 140 CPU 671 60S • 140 CPU 678 61
Uni-Telway	-	-	All processors, via computer port and PCMCIA cards	-
Modbus (1)	On processors: • BMX P34 1000 • BMX P34 2000 • BMX P34 2010	-	-	All processors have a Modbus connection which can be used by EcoStruxure Control Expert.

Type of Connection	M340	M580	Premium and Atrium	Quantum
	• BMX P34 20102 • BMX P34 2020			
XIP (2)	-	-	Via the following modules: • TSX ETY 110 • TSX ETY 4103 • TSX ETY 5103 • TSX ETY PORT • Ethernet channel of TSX P57 4634 • Ethernet channel of TSX P57 5634 • Ethernet channel of TSX P57 6634	-
Modbus Plus	-	-	Via PCMCIA cards TSX MBP 100 (2)(3)(4)	All processors
Ethernet TCP/IP (2)	Processors Ethernet channels: • BMX P34 2020 • BMX P34 2030 • BMX P34 20302 Via the following modules: • BMX NOE 0100 • BMX NOE 0110	All processors	Processors Ethernet channels: • TSX P57 4634 • TSX P57 5634 • TSX P57 6634 Via the following modules: • TSX ETY 4103 (3) • TSX ETY 5103 (3) • TSX ETY PORT (3)	Processors Ethernet channels: • 140 CPU 651 50 • 140 CPU 651 60 • 140 CPU 651 60S • 140 CPU 652 60 • 140 CPU 658 60 Via the following modules: • 140 NOE 211 x0 • 140 NOE 251 x0 • 140 NOE 771 00 • 140 NOE 771 01 • 140 NOE 771 10

Type of Connection	M340	M580	Premium and Atrium	Quantum
				• 140 NOE 771 11 • 140 NOE 771 20
PCI	-	-	On Atrium processors only: • TSX PCI 57 204 • TSX PCI 57 354	-
Fipio (2)	-	-	All controllers connected to the Fipio bus to which the PC running EcoStruxure Control Expert is connected, as well as all those accessible via X-Way routers	-
Fipway (2)	-	-	All controllers connected to the Fipway network to which the PC running EcoStruxure Control Expert is connected, as well as all those accessible via X-Way routers	-

(1) Bluetooth connection is limited to Diagnostics and Animation.

(2) Requires a valid application to be installed on the controller and requires the communication cards or modules used to be configured correctly.

NOTE: For Modicon M580 and M340 controllers, the TCP/IP connection does not require a valid application inside the controller to work.

- For details on Modicon M580 default address configuration, refer to *Control Expert Configuration Tabs* (see Modicon M580, Hardware, Reference Manual).
- For details on Modicon M340 default address configuration, refer to *Methods for IP Addressing* (see Modicon M340 for Ethernet, Communications Modules and Processors, User Manual).

(3) Cannot be used to view application-specific dialog boxes in online mode.

(4) Programming mode, page 83 cannot be accessed via a Modbus Plus network.

Access Communication Parameters

Access Controller Parameters

Access the communication parameters for the controller:

Step	Action
1	Open the **Set Address** dialog box, page 93 (**PLC > Set Address**).
2	In the **PLC** group box, click the **Communication Parameters** button to open the **PLC Communication Parameters** dialog box.
3	Configure these parameters: • **Number of tries:** Enter the number of recovery attempts to make. • **Timeout (ms):** Enter a timeout delay to use for recovery: ◦ *Uni-Telway:* Use the value 4000 ms. ◦ *Modbus ASCII:* Use the value 6000 ms. • **Speed rate auto-adaptation at the end of download:** Select (check) this box to automatically use the maximum speed permitted by the connected media (to a maximum of 115 KBaud). This option is only available in *Uni-Telway* mode for TSX573•• and TSX574•• controllers. For other controller types, the speed at 115 KBaud, this option has no effect. • **Speed at 115 KBds:** Select (check) this box to automatically set the project loading speed to 115 KBaud. • **Driver Settings:** Click this button to display the driver management dialog box (see *Communication Drivers, Installation Manual*).
4	Click the **OK** button to confirm the configuration.

Access Simulator Parameters

Access the communication parameters for the simulator (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual):

Step	Action
1	Open the **Set Address** dialog box, page 93 (**PLC > Set Address**).
2	In the **Simulator** group box, click the **Communication Parameters** button to open the **Simulator Communication Parameters** dialog box.
3	Configure these parameters: • **Number of tries:** Enter the number of recovery attempts to make. • **Timeout (ms):** Enter a timeout delay to use for recovery.
4	Click the **OK** button to confirm the configuration.

Access Bandwidths

You may divide up the bandwidth quota allocated to EcoStruxure Control Expert for the selected communication medium between the different functions available in online mode.

Connected Mode Services Bandwidth dialog box is used to define the percentage of bandwidth allocated to each of the 4 categories of functionality in online mode:

- **Animation Support**: animation tables, operator screens, animation of language editors, application-specific dialog boxes. The higher the percentage allocated, the higher the frequency with which the data in the controller is scanned.
- **PLC Monitoring**: monitoring of the overall operating mode of the controller (Run or Stop), or of its tasks for the "Program debugging" function. The higher the percentage allocated, the higher the frequency with which the state of the controller and its tasks is scanned.
- **PLC Screen** (controller debug dialog box): the higher the percentage allocated, the higher the frequency with which the controller and application information is scanned.
- **Diagnostic** (Diagnostic Viewer). The higher the percentage allocated, the higher the frequency with which the alarms in the controller are acquired.

With the foregoing parameters, you can optimize performance in online mode by adjusting the parameters to suit:

- the characteristics of the project loaded onto the controller: numerous and frequent diagnostics alarms, many animation screens open at once
- the use made of online mode: the most frequently used functions.

Procedure:

Step	Action
1	Access the **Set Address** dialog box, page 93.
2	Click the **Bandwidth** button to open the **Connected Mode Services Bandwidth** dialog box.
3	Using the sliders, adjust the parameter values to suit the characteristics of your project and the functions you use most frequently in online mode. **NOTE:** Click the **Default** button to revert to the default values. **NOTE:** The **Freeze** option is used for fine adjustment (in increments) of the parameters to suit the characteristics of the project loaded in the controller. It is used to retain the value of a parameter once the response time of the corresponding function is determined, and then adjust only the parameters corresponding to the other functions.
4	Click the **OK** button to confirm the configuration.

Project comparison

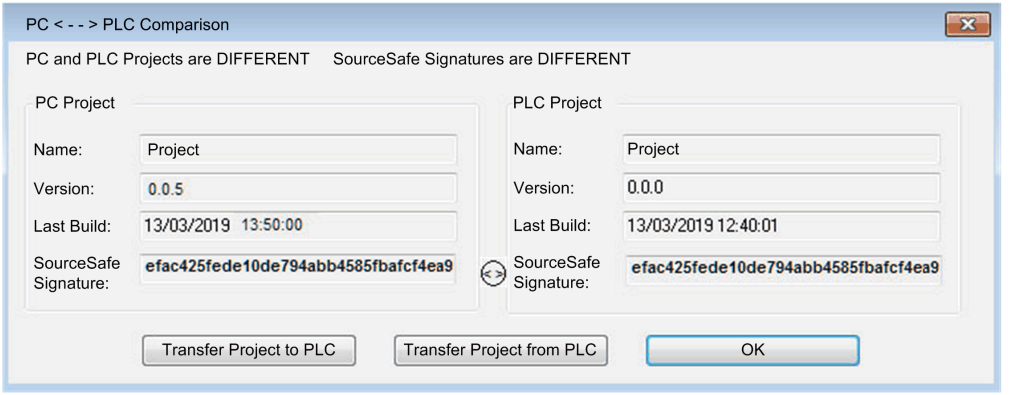
At a Glance

Project comparison provides you with a concise overview of the possible differences between a project embedded in the PLC and a project in the terminal. For each project, the following information is displayed:

- **Name**,
- **Version**,
- **Last build**: date and time the executable was last generated.
- **SourceSafeSignature**: an SHA256 algorithmic fingerprint of the safe application.

Procedure

Perform the following actions to launch project comparison:

Step	Action
1	Activate the **PLC > Compare** command, which displays the screen below.  **NOTE:** Both the project and the safe application SourceSafeSignature values are compared. The combination of the two comparisons is displayed in an icon placed between the two SourceSafeSignature values: • The icon in the graphic, above, indicates both the Projects and the SourceSafeSignatures are DIFFERENT. •  indicates both the Projects and the SourceSafeSignatures are EQUAL. •  indicates the Projects are DIFFERENT, but the SourceSafeSignatures are EQUAL.
2	Confirm with **OK**.

Different projects

If the projects are different, 2 commands let you carry out a transfer between the terminal and the PLC or vice versa, thus changing to Equal online mode:

- **Transfer Project to PLC**, transfer the project from the terminal to the PLC,
- **Transfer Project from PLC**, transfer the project from the PLC to the terminal.

NOTE: When the transfer is not possible, the corresponding button is grayed out (for example, the **Transfer Project from PLC** command is grayed out when the project embedded in the PLC is invalid or when information required for uploading is missing or invalid).

Conversion of Legacy Applications/Projects

Introduction

There are 2 possibilities to convert legacy applications to Control Expert.

- Global conversion provides to convert a legacy application (PL7, Concept) to Control Expert as a whole.
- Partial conversion with the conversion wizard provides to select parts of a legacy application to be converted and to control the conversion result.

To convert a legacy application to Control Expert, you must at first export the application out of the legacy software (PL7, Concept).

NOTE: In case of global conversion with a default hardware configuration (see below) as well as in case of partial conversion, the hardware configuration must be edited and completed by the user.

Global Conversion

To convert a legacy application as a whole you select the file exported out of the legacy software in Control Expert via **File > Open**.

The respective converter (PL7 Converter, Concept Converter) automatically generates a Control Expert application.

For detailed information on the converters, please refer to *EcoStruxure™ Control Expert, PL7 Application Converter, User Manual* or *EcoStruxure™ Control Expert, Concept Application Converter, User Manual*.

- Global conversion for Premium and Quantum contains also a hardware configuration.
- Applications with PLCs which do not have a direct equivalent in Control Expert (Compact, Momentum, TSX Micro) should be converted using the partial conversion with the conversion wizard.

However, if global conversion is chosen, a default hardware configuration is generated.

- For global conversion of Compact applications you can select either Quantum or Modicon M340 as target PLC. Please refer to *Importing an LL984 Application*, page 107.
- Global conversion of Momentum applications contains only a default Quantum hardware configuration with a CPU (140 CPU 534 14A/U) and a power supply (140 CPS 424 00). If needed the default CPU can be replaced.
- Global conversion of TSX Micro applications contains only a default Modicon M340 hardware configuration with a CPU (BMX P34 2020) and a power supply (BMX CPS 2000). If needed the default CPU can be replaced.

Partial Conversion

To convert a legacy application partially to Control Expert, use the conversion wizard.

You can launch the conversion wizard in Control Expert via **Tools > Convert Partially**.

With the conversion wizard you can:

- select the parts of the application to be converted
- remap I/O objects before conversion
- adapt concurrently the hardware configuration of the new application in Control Expert
- modify the amount of used memory in the CPU

The partial conversion is possible for TSX Micro, Premium, Quantum, Momentum and Compact applications.

With partial conversion for Compact and Momentum applications it is recommended to prepare a Modicon M340 hardware configuration.

For detailed information, please refer to chapter *Conversion Wizard*, page 1742.

Importing an LL984 Application

Introduction

When you import an application to EcoStruxure Control Expert from other programming software, some functions may behave differently.

For example:

- Cold start behavior of EcoStruxure Control Expert LL984 applications is different from legacy LL984 applications.
- Imported Concept IEC coils behave differently than imported LL984 coils.
- EcoStruxure Control Expert MATH functions might not behave the same as legacy LL984 (overflow, underflow, etc.).

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

- After importing an application program, edit the result carefully to ensure that all the logic elements have been ported completely and are located correctly in the program.
- Test the application before running it to verify that it operates as expected.
- If necessary, edit the program to ensure that the imported logic operates as intended.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

General Message after Import

After the import check the log file, page 116 for error and warning messages.

You have to **Analyze** and **Build** the project.

Import

To import an LL984 application use the **File > Open** menu command.

Please refer to **Open** dialog box, page 76.

You can import the following legacy LL984 applications:

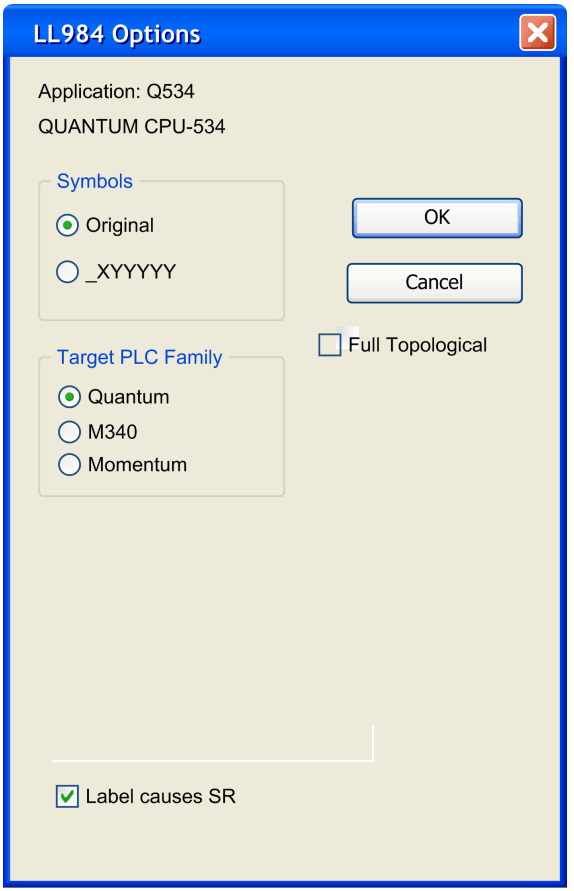
- Concept Projects (**.ASC*)
- Modsoft Projects (**.CFG*)
- ProWORX32 Databases (**.PWX*)
- ProWORX Nxt (**.DCF*)

  NOTE: Imported MSTR functions are not designed to work with the Modicon M340 controllers. After an application is converted to EcoStruxure Control Expert, it must be manually updated. The MSTR function must be replaced by the READ_VAR and WRITE_VAR functions.

For further information please refer to Import / Export of LL984 Applications (see *EcoStruxure™ Control Expert, LL984 Editor, Reference Manual*).

LL984 Options Dialog Box (-> Target Controller Quantum)

After selecting a legacy LL984 application file, the following **LL984 Options** dialog box is displayed. The following graphic shows an example with a legacy Quantum LL984 source application and **Quantum** selected in **Target PLC Family**.



Element	Description
Application	Displays the name of the legacy application to be imported.
Source PLC Family	Displays the controller family of the legacy application to be imported.
OK	Starts the import.
Cancel	Displays a message . . . "Convert" step failed, no application is imported, and the following message is generated in the import log file: "Conversion terminated by User".
Full Topological	This check box is only used to convert applications to Modicon M340 controllers.

Element	Description
Symbols	
Original	Select this option to use the original variable names of the legacy application in your new EcoStruxure Control Expert application.
_XYYYYY	Select this option to automatically convert the legacy variable names. The new variable names will be composed according to the following list: • **_**: an underscore will be the leading sign of the converted variables. • **X**: represents the data type in the memory area. ◦ **0**: %M (memory bit) ◦ **1**: %I (discrete input) ◦ **3**: %IM (input word) ◦ **4**: %MW (memory word) • **YYYYY**: represents the address number. For example **_400001** stands for%MW1.
Target PLC Family	
Quantum	Select this option to use the Quantum as target controller.
M340	Select this option to use the Modicon M340 as target controller. The Modicon M340-specific parameters for full topological addressing are listed in a separate table below.
Momentum	Select this option to use the Modicon Momentum under EcoStruxure Control Expert as target controller.
Label causes SR	Select this option to import a segment containing labels as an SR segment, even if its scheduling does not express this. Labels are allowed only in SR segments. For details please refer to Calling Subroutines, page 1173.

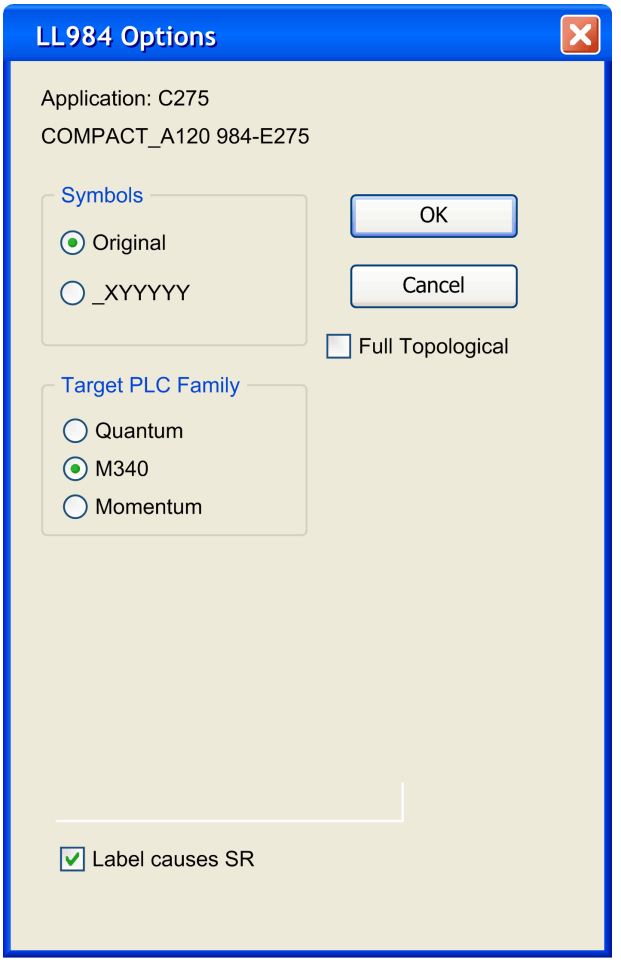
Element	Description
Shift 984 Remote	This additional option appears if the legacy application controller family is a 984 family. If selected, a Quantum local rack is added to the application and all drop numbers are increased by one until an unused drop number is reached, keeping the old 984 rack as the first remote rack (drop number 2) without the old controller. If the option is not selected, conversion attempts to translate the local rack and its modules to Quantum, as far as possible. A common situation encountered is that 984 local drops often contain more racks than allowed with Quantum.
Long Comments are in project and in directory	This additional dialog box displays the path where the long comments are stored if the legacy application contains long comments. This path is identical to the path where the legacy application is stored plus one additional directory for the long comments. Long comments appear twice in the conversion: • Inside the internal project database of the data editor. • As text files in the additional application path directory for long comments. To back up all information of the application, only the files in the additional application path directory are to be saved. To achieve consistency between data editor and text files, comment in both places must be edited together.

LL984 Options Dialog Box (-> Target Controller Modicon M340)

Converting a legacy LL984 application to a Modicon M340 application you can determine the kind of addressing. By default Mixed topological / State RAM addressing is used (option **Full Topological** not selected).

Mixed Topological / State RAM Addressing (-> Modicon M340)

The following graphic shows an example for converting a legacy Compact application to a Modicon M340 application with Mixed topological / State RAM addressing (option **Full Topological** not selected).



NOTE:

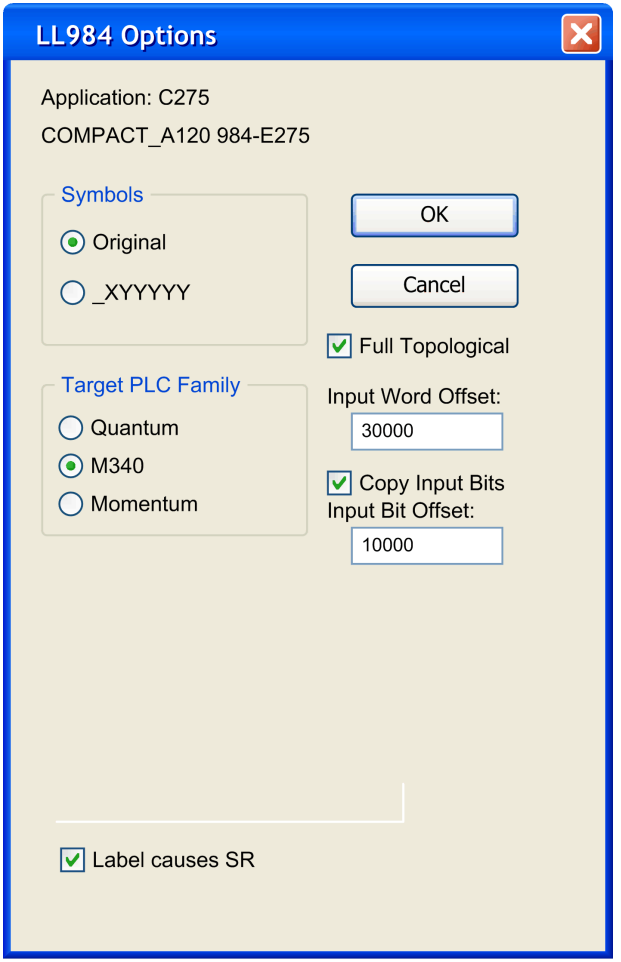
To import a legacy LL984 Compact application that uses Modbus requests to communicate with an HMI device, use State RAM addressing to preserve the Modbus exchange between the controller and the HMI.

Full Topological Addressing (-> Modicon M340)

To convert a legacy LL984 application to a Modicon M340 application with full topological addressing, the option **Full Topological** has to be selected.

- All word variables get a declaration with a symbol and the topological address in the data editor, if not already present.
- **Input Words** are copied to a mirror area of memory words, allowing SCADA access.
- Discrete words are directly addressed with topological addresses in the programs, if no symbols are defined for them, showing in this way directly their hardware connection to the controller.
- If SCADA access is required to discrete inputs, the input bits can be copied as well to a mirror area of memory bits by checking **Copy Input Bits**.
- SCADA access for inputs with **Full Topological** checked needs modified addresses in the SCADA system – the offsets in the dialog box must be added to the original offsets and the memory areas must be used instead of the input areas.

The following graphic shows an example for converting a legacy Compact application to a Modicon M340 application with full topological addressing.



For converting a legacy LL984 application to a Modicon M340 application with full topological addressing, the following additional parameters are used:

Element	Description
Full Topological	Activate this option to convert input word addresses (3x) of the legacy LL984 application to a separate, mirrored area of the %MW areas of the target Modicon M340 application. If this option is not activated, the legacy LL984 application is converted into a State RAM Modicon M340 application.
Input Word Offset	Enter the offset for the input word addresses (3x) in the mirrored area of the %MW area of the target Modicon M340 application.
Copy Input Bits	Activate this option to additionally convert input bit addresses (1x) to the mirrored area of the %M area of the target Modicon M340 application.
Input Bit Offset	Enter the offset for the input bit addresses (1x) in the mirrored area of the %M area of the target Modicon M340 application.

NOTE: With this **Full Topological** option activated you must manually change the access to the %I and %IW addresses in the SCADA components.

Converting a Legacy Compact Application to Modicon M340 with Full Topological Addressing

During conversion of a legacy Compact application to a Modicon M340 application with full topological addressing, the import automatically creates separate ST code sections for input and output operations with copying commands to transfer I/O data to the mirrored memory areas.

Please refer to *Module Emulation* (see EcoStruxure™ Control Expert, LL984 Editor, Reference Manual).

Import Behavior

During import of a legacy LL984 application, problems may be detected:

- controller family not supported
- I/O family not supported
- I/O module not supported
- rack count exceeded
- instruction/function block not supported
- non-coil in column 11

Generally such an issue does not stop the import.

Instead a message is displayed for each of these detected problems in the **Import/Export** tab of the **Output Window** at the bottom of the EcoStruxure Control Expert user interface.

Double-click such a message in the **Output Window** to access the respective dialog box in your application.

If detected problems occur the following is done during import:

- Not supported controllers or I/O families are replaced by a default Quantum/Modicon M340 configuration.
- Not supported I/O modules are ignored in case of Quantum as target controller family.
- Exceeding racks and their modules are ignored.
- Not supported instructions are represented by empty DFBs that look identical to the LL984 instruction. In such a case proceed as described under *Not Supported Legacy LL984 Instructions and User Loadables* (see *EcoStruxure™ Control Expert, LL984 Editor, Reference Manual*).
- Equation networks with questionable conversions (for example implicit type cast) will contain the imported equation statement and an additional message.

  Equation networks which are not compliant with Modicon M340 variable alignment rules, will cause a build error during **Analyze**.

  For details please refer to Equation Networks (see *EcoStruxure™ Control Expert, LL984 Editor, Reference Manual*).
- The import of a legacy application that contains duplicated identifiers is aborted. For details please refer to Duplicated Identifiers (see EcoStruxure™ Control Expert, LL984 Editor, Reference Manual).

If parts of a legacy application are not imported, after **Analyze** and **Build** of the project an appropriate message is displayed. In this case you have to adapt the application manually. (For example in case of Modicon M340 by selecting **Mixed topological and State RAM** in the *Memory Tab*, page 1225 of the modules and subsequently fixing the related variables).

Import Log File

After import a message informs you that the conversion and import is finished and reminds you to have a look at the import log file.

The import log file is stored in the root directory of your application.

You have to **Analyze** and **Build** the project.

Transfer the Project Between EcoStruxure Control Expert and the Controller

Introduction

The transfer of a project facilitates these copying actions:

- Copy the open project from EcoStruxure Control Expert to the memory of the controller that corresponds to the selected address.
- Copy the embedded project from the controller to EcoStruxure Control Expert. It then becomes the active EcoStruxure Control Expert project.

  NOTE: If you opened Control Expert editor from the Topology Manager of EcoStruxure Control Expert, you cannot transfer the project from the controller. Depending on the controller platform, proceed as follows:

  - M580, M340, MC80, and Momentum platforms: Transfer (upload) the project from the controller to the system project by using the Topology Manager.
  - Quantum and Premium platforms: Transfer the project from the controller to EcoStruxure Control Expert Classic.

Transfer from EcoStruxure Control Expert to the Controller

Transfer the active project to a controller:

Step	Action
1	Use the **PLC > Set address** command to define the controller to which you would like to transfer your project, page 93.
2	Change to online mode, page 91.
3	Select the **PLC > Transfer Project to PLC** command, which displays the corresponding dialog box, page 118. **NOTE:** You can select the **PLC Run after Transfer** check box to automatically switch the controller to Run after the transfer.
4	Select the **Transfer** command. **NOTE:** If the project has not been generated in advance, a dialog box appears that allows you to generate it before the transfer (**Rebuild All then Transfer**) or interrupt the transfer (**Cancel Transfer**). **Result**: The transfer progress is displayed in a dialog box. You can interrupt the transfer with the **Esc** key to invalidate the controller project. **NOTE:** In the event that the project is transferred to a Flash memory card, the transfer can take several minutes.

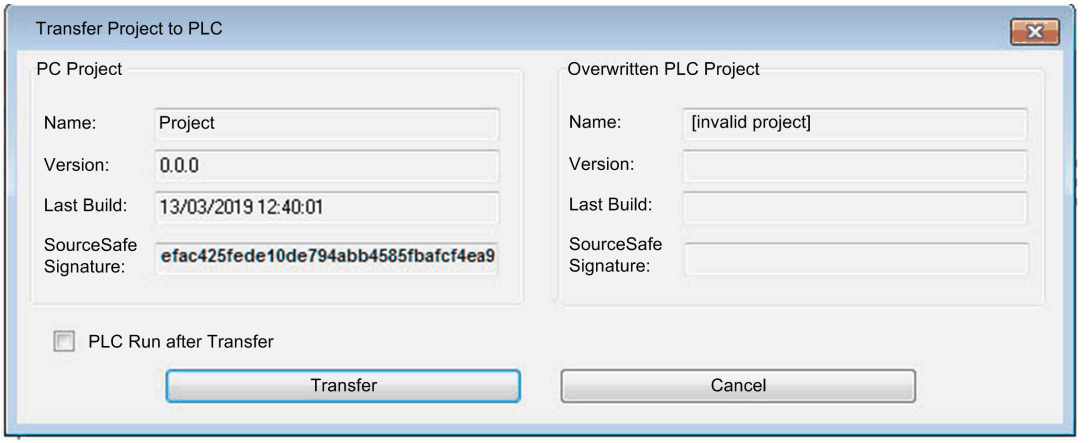
Transfer From the Controller to EcoStruxure Control Expert

Perform these actions to transfer the contents of the controller to EcoStruxure Control Expert:

Step	Action
1	Use the **PLC > Set address** command to define the controller from which you would like to transfer the project, page 93.
2	Change to online mode, page 91.
3	Select the **PLC > Transfer Project from PLC** command to display the corresponding dialog box. **Result**: The transfer progress is displayed on the screen. You can interrupt the transfer with the **Esc** key.

EcoStruxure Control Expert to Controller Transfer Dialog Box

Use this dialog box to transfer the project from EcoStruxure Control Expert to the controller:



Parameter Descriptions

This table describes the parameters of the EcoStruxure Control Expert to controller transfer dialog box.

Parameter	Description
PC Project	Identification field for the project in EcoStruxure Control Expert: • **Name** • **Version** • **Last Build**: date and time the executable was last generated. • **SourceSafeSignature**: An SHA256 algorithmic fingerprint of the safety-related application (for safety-related controllers). **NOTE:** An icon located between the **SourceSafeSignature** entries in the **PC Project** field and the **Overwritten PLC Project** field indicates a comparison between the two values: •  indicates the values are DIFFERENT. •  indicates the values are EQUAL. • No icon indicates there is no application in the controller.
Overwritten PLC Project	Identification field for the project in the controller. This project is overwritten by the transfer: • **Name** • **Version** • **Last Build**: date and time the executable was last generated. • **SourceSafeSignature**: An SHA256 algorithmic fingerprint of the safety-related application (for safety-related controllers).
PLC Run after Transfer	If this box is selected, the controller changes to Run after the transfer.
Transfer	Transfer command.
Cancel	Cancels the transfer.

This table describes the parameters of the controller to EcoStruxure Control Expert transfer dialog box.

Parameter	Description
Overwritten PC Project	Identification field for the project in EcoStruxure Control Expert: Name, version, date and time the executable was last generated. The transfer overwrites the project.
PLC Project	Identification field for the project in the controller: Name, version, date and time the executable was last generated.
Transfer Project from PLC	Transfer command.
Cancel	Cancel the transfer.

Parameter Descriptions

Transfer Projects Deployed From EcoStruxure Control Expert

If you transfer to EcoStruxure Control Expert Classic a project that was deployed by using EcoStruxure Control Expert, after the transfer completes, EcoStruxure Control Expert Classic has the same limited functionality as Control Expert editor that opens in the Topology Manager.

For example, the **Rebuild all Project** and **Transfer Project to PLC** commands are disabled. However, the **Build Changes** command is available for changes made offline or online, if applicable. A message is displayed to inform you about this limitation after the transfer completes.

Incompatibility During a Transfer

These conditions during the transfer can trigger an on-screen message:

- An incompatibility between the selected project and the controller configuration is detected during the transfer from EcoStruxure Control Expert to the controller.
- An incompatibility between the project in the controller and the EcoStruxure Control Expert version is detected during the transfer from the controller to EcoStruxure Control Expert.

Update of Upload Information

Upload Information

Upload information consists of:

- Project upload information: graphic code for languages with LD contacts and chart of FBD function blocks, symbols of located and unlocated variables,
- Comments: variables, code sections, project, DFBs, etc.,
- Animation tables.

These are always saved when a project backup is performed on the terminal (STU or STA file), but may or may not be included with the executable code, page 551 when building the project which is then transferred to the PLC.

Use of Upload Information

Upload information should only be loaded in the PLC where this is strictly necessary, in order to optimize the available memory size and save this for the executable code, and also to improve the performance of modifications in online mode.

In the coding and debug phase, this information is generally present on the workstation and therefore does not need to be transferred to the PLC, thus achieving a real gain in performance.

In the maintenance and operation phase, this information should be included in the PLC where the application needs to be restored on empty terminals (i.e. that do not have the project STU or STA file). However, in addition to the gain in memory space, the absence of the Upload information also forms a protection against reading and writing.

Choice of Upload Information

By default, the upload information necessary to rebuild the project program is not embedded in the PLC.

You can choose the information to have embedded:

- The information necessary for uploading the project,
- Comments (variables and types),
- The animation tables.

To access the Upload information, page 551 select: **Tools > Project Settings > PLC Embedded data**

Update

When making modifications in online mode, it is possible to choose between 2 modes of updating the Upload information:

- Automatic update; in which case the update of Upload information is implicit on each build,
- Update on user request; in which case the update is performed explicitly, by the clicking **PLC > Update Upload Information**.

The way in which upload information is updated is chosen in the Project Settings screen, page 596 and is accessible by: **Tools > Project Settings > PLC Embedded data**

NOTE: The upload information is updated implicitly whenever a terminal project is transferred to the PLC (where this is included with the executable code).

On connection, a message asks you to confirm the update of the Upload information, if this has not already been performed.

Sending a command to the PLC

At a Glance

The **Run/Stop** and **Init** commands enable the project to be controlled from the terminal when it is stored on a target PLC:

- **Run/Stop** launches or stops execution of the project, (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual)
- **Init** initializes the project.

Procedure

You must perform the following actions to control the PLC:

Step	Action
1	Use the **PLC > Set address** command to define the target PLC that you wish to control, page 93.
2	Change to online mode.
3	Activate the **PLC > Run/Stop** command to launch or stop execution of the project or the **PLC > Init** command to initialize the project.
4	Confirm the command.

Project Backup Management for Modicon M340

At a Glance

Automatic and manual accesses are carried out between the Schneider BMX RMS ••••• memory card and EcoStruxure Control Expert or the controller to:

- restore a project in the controller
- save the controller project on the memory card
- compare the controller project and the memory card project

The memory card needs a structure to contain program and data. Formatting without using the SD Card Writer tool destroys this structure.

NOTICE
INOPERABLE MEMORY CARD Format the memory card using EcoStruxure Control Expert SD Card Writer tool. **Failure to follow these instructions can result in equipment damage.**

NOTICE
LOSS OF APPLICATION DATA • Do not store the SD card where there is static electricity or probable electromagnetic fields. • Do not store the SD card in direct sunlight, near a heater, or other locations where high temperatures can occur. • Do not bend the SD card. • Do not drop or strike the SD card against another object. • Keep the SD card dry. • Do not touch the SD card connectors. • Do not disassemble or modify the SD card. **Failure to follow these instructions can result in equipment damage.**

Restore Function

The restore function is carried out as follows:

- Automatically:
  - after a power cycle
  - on a warm start
  - on a cold start
- Manually, using the EcoStruxure Control Expert command **PLC > Project Backup... > Backup Restore**.

  NOTE:

  - When you insert the memory card in run or stop mode, power cycle to restore the project on the controller.
  - If the application contained in the memory card includes the RUN AUTO option, then the controller will automatically restart in RUN mode with this application as soon as the inserted memory card content is restored to the controller.

⚠WARNING
UNINTENDED EQUIPMENT OPERATION Ensure that the intended user application is contained in the memory card before inserting it into the controller. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

Save Function

The save function is carried out as follows:

- Automatically, after:
  - a download, if the memory card is present and not write-protected
  - online modification, if the memory card is present and not write-protected
  - detection of a system bit %S66 rising edge
- Manually, using the command **PLC > Project Backup... > Backup Save**.

  NOTE: If you remove the memory card when the backup is in progress, the program on the memory card is lost. With system bit %S65 you can disable processor access to the memory card.

  If the memory card is not present or write-protected, the download in Processor memory is allowed but the online modification is not allowed.

Compare Function

You can compare your project with the one contained on the memory card. To do so, use the command **PLC > Project Backup... > Compare**.

Memory Card State and Diagnostic

EcoStruxure Control Expert and the controller front panel display the status of the memory card:

Display on	Icon/LED State	Description
Status bar at bottom of EcoStruxure Control Expert		Project on memory card is different from controller project.

Display on	Icon/LED State	Description
		Memory card is not present on the controller or is inoperable.
	no icon	Project in memory card is equivalent to controller project.
LED on bottom of memory card slot	On	Access to the memory card is enabled.
	Flashing	Activity between the memory card and the controller.
	Off	Access to the memory card is disabled. You can remove the memory card.
CARDERR LED on PLC front panel	On	Memory card is missing or memory card is not detected, or project on memory card is different from controller project.
	Off	Memory card is detected, and project in memory card is equivalent to controller project.

Project backup for Premium

At a Glance

Certain PLCs Premium allow you to save the project program contained in the executable zone (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) (RAM) of the PLC in the backup memory zone.

The available backup memory card is the **TSX MFP B 096K**.

Control Expert enables the use of these functions:

- Comparison
- Automatic restore (shaded field) when the PLC powers up, if the project program is different
- Save PLC program on the memory card
- Clearing of card data

NOTE: The backup function is not available when the project is already being executed on a RAM or EPROM memory card.

Comparison

You can compare your project with the one contained in the backup memory. To do this, perform the following actions:

Step	Action
1	Activate the command **PLC > Project backup > Compare...**. **Result**: A dialog box appears informing you of the result of the comparison.

Backup Memory Card → RAM Transfer

You Must Perform the Following Actions to Restore the Contents of the Backup Memory to the Executable Zone of the PLC:

Step	Action
1	Set the **WP** switch on the backup memory card to **ON** (lock).
2	Insert the backup memory card into the PLC.
3	**Result**: The memory card automatically transfers the contents of the backup memory to the executable zone of the PLC.

NOTE: Inserting the memory card in its slot triggers a cold start of the PLC.

RAM → backup memory card transfer

You must perform the following actions to transfer the project program to the backup memory:

Step	Action
1	Set the **WP** switch on the backup memory card to **OFF** (unlock).
2	Insert the backup memory card into the PLC.
3	Activate the command **PLC > Project backup > Save...**. **Result**: A command confirmation screen is displayed.
4	Confirm with **OK**.
5	Remove the backup memory card from the PLC.
6	Set the **WP** switch on the backup memory card to **ON** (lock).

Clearing procedure

You must perform the following actions to clear the backup memory:

Step	Action
1	Set the **WP** switch on the backup memory card to **OFF** (unlock).
2	Insert the backup memory card into the PLC.
3	Activate the command **PLC > Project backup > Clear...**. **Result**: A command confirmation screen is displayed.
4	Confirm with **OK**.

Save/Restore Data Between a File and the Controller

At a Glance

The data transfer function allows you to store the value of the following data (see *EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual*) in a file:

- boolean-type located data: `%Mi`
- word-type located data: `%MWi`
- unlocated variables
- function block instances

There are 2 types of data files:

- DAT:
  - Saving controller data to a *.DAT file is allowed for all previous data types. *.DAT contains only data blocks.
  - Restoring all data blocks from a *.DAT file to the controller is allowed if the project has not been built since the Save was done.If a build was done, only BOOL and WORD located data (%Mi and %MWi) are restored.
- DTX:
  - Saving controller data to a *.DTX file is allowed for all previous data types. In addition, application references (at the time of the Save) are saved.

    NOTE: I/O and device DDT objects cannot be saved in DTX data files.
  - Restoring data from a *.DTX file is allowed even if the application has been built and/or data has been modified since the Save was done. Compatibility Rules for Restore Using a *.DTX File, page 132 explains the restoration process and the types of data that cannot be restored.

    NOTE: It is recommended to convert any *.DAT files to *.DTX file by restoring the *.DAT file and then saving using the *.DTX format.

NOTE: The data transfer function can be used for migrating from different controller range. You can save M340 or Premium data to a *.DTX file and restore to M580 controller the *.DTX file.

⚠CAUTION
LOSS OF DATA Before transferring unlocated variables and function block instances data to the controller, ensure that your application is in a compatible state with the saved data. **Failure to follow these instructions can result in injury or equipment damage.**

Procedure for Saving Controller Data

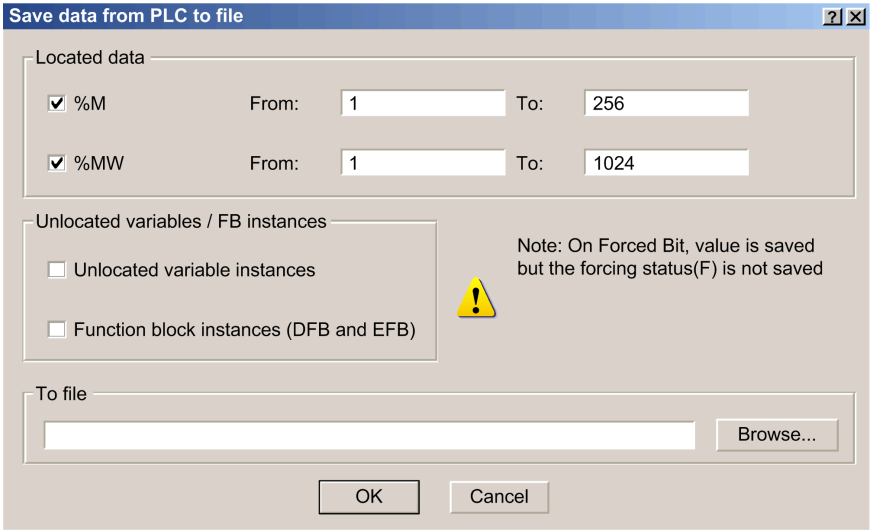
If the controller is in RUN mode, the saving of data from the controller to a file may need several application cycles and the data, in the file, may be de-synchronized, which may affect application execution during restoration.

NOTE: On forced bits, value is saved but the forcing status (F) is not saved.

⚠WARNING
UNINTENDED EQUIPMENT OPERATION Before saving the data, verify the impact on the application execution. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

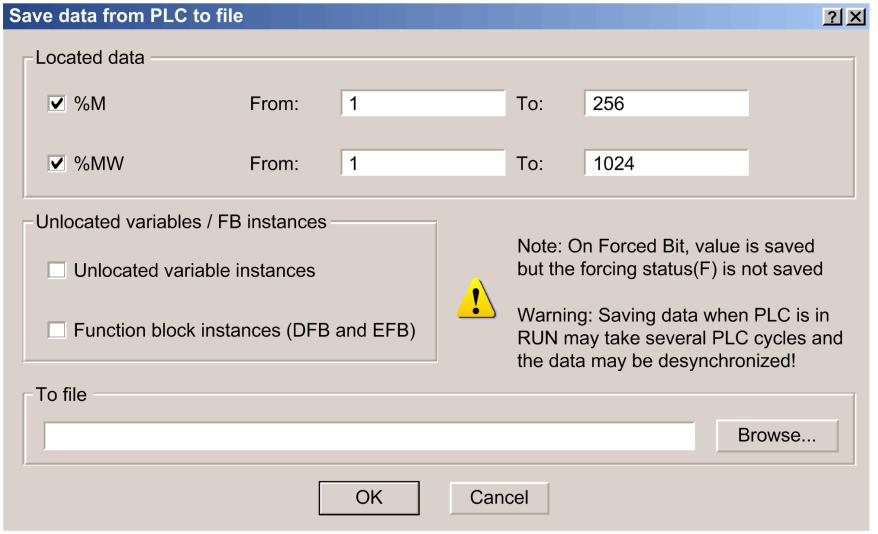
You must perform the following actions to save the data from the controller to a file:

Step	Action
1	To enable the menu item **Save Data from PLC to File**, the following must be true: • The controller is loaded with an application. • The controller is in RUN or STOP mode. • EcoStruxure Control Expert is running and is connected to the controller. • The Application built in EcoStruxure Control Expert is equal to those in the controller. • The Application in EcoStruxure Control Expert is in the built state.
2	Activate the **PLC > Save Data from PLC to File** command. **Result**: The **Save Data from PLC to File** dialog box (controller in STOP mode) opens.

Step	Action
	 For more information, refer to the corresponding dialog box, page 130.
3	Using the check boxes, select the data type(s) to be transferred. For located data you must also define the range of data to be transferred by entering the start and end address of the range.
4	Define the path and the name of the file to which the data is to be saved.
5	Confirm with **OK**.

Data Transfer Dialog Box for Transfers to a File

The **Save Data from PLC to File** dialog box (controller in RUN mode) is shown below:



The following table shows the parameters of the **Save Data from PLC to File** dialog box.

Parameter	Description
Content	This field is used to define the type and range of the data to be transferred. The checkboxes are used to define the type of located data. **From**: and **To**: fields are used to define the range for located data. Default values: • **%M** and **%MW** checkboxes are checked • **%M** and **%MW** range is set to the controller values • **Unlocated variables instances** and **Function block instances (DFB and EFB)** are not checked
To file	This field is used to define the path and name of the file.
Browse	This button can be used to browse the disk to define the file name and path.

Procedure for Restoring Data to the Controller

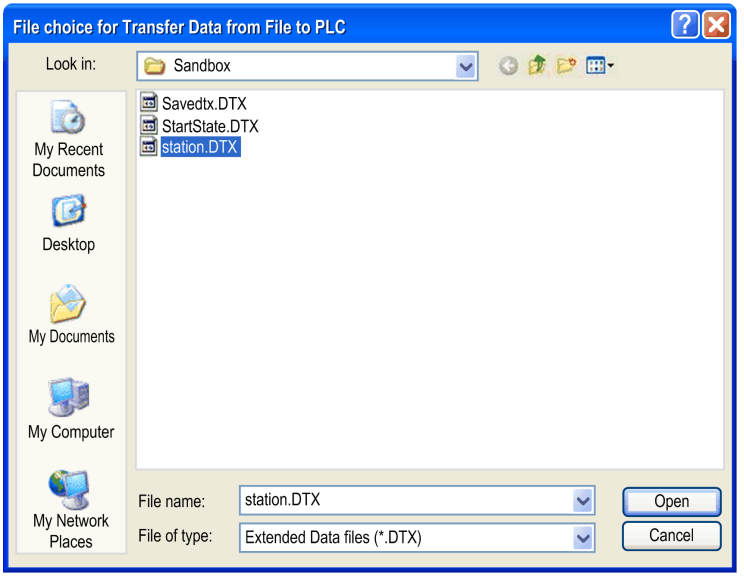
If forced bits are detected in the controller, forcing status (F) and value will not be updated in the controller.

If the controller is in RUN mode, the restoring of data to the controller from a file may need several application cycles and the data may be de-synchronized, which may affect application execution.

⚠WARNING
UNINTENDED EQUIPMENT OPERATION Before restoring the data, verify the impact on the application execution. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

You must perform the following actions to restore the data from a file to the controller:

Step	Action
1	To enable the menu item **Restore Data from File to PLC**, the following must be true: • The controller is loaded with an application. • The controller is in RUN or STOP status. **NOTE:** It is strongly recommended to do the transfer with the controller in STOP mode. • EcoStruxure Control Expert is running and is connected to the controller. • The Application built in EcoStruxure Control Expert may be different than those in the controller. • The Application in EcoStruxure Control Expert is built state.
2	Activate the **PLC > Restore Data from File to PLC** command. If the controller is in RUN status, a message is displayed allowing to perform the transfer while the controller is still operating (**Ok**) or to cancel the transfer. If forced bits are detected in the controller, a message is displayed allowing to perform the transfer (**Ok**) or to cancel the transfer.

Step	Action
	Result: The **Restore Data from File to PLC** dialog box opens. 
3	Select the file to be restored to the controller.
4	Confirm with **Open**.
5	A confirmation message is asking you if you want to proceed with the transfer.

Compatibility Rules for Restore Using a *.DTX File

At a Glance

The DTX file allows a restore even if there have been changes in the application since the save was done.

When a restore is performed, if the build number of the application is greater than the version of the DTX file, EcoStruxure Control Expert analyses the content of the file to verify the compatibility with variables of the controller application.

The following tables present the rules followed for restoring variables when there is a difference between the *.DTX file and the controller application. This condition results from building the application after the data was saved in the *.DTX file.

⚠WARNING
UNINTENDED EQUIPMENT OPERATION Before restoring the data, verify the impact on the application execution. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

Controller Versions Compatibility (Premium, Quantum , M340, M580)

The memory allocation for DTX files is not the same for all controller versions:

The DTX file is created from ...	Result
Controller version 2.0 or earlier	The data cannot be restored on controller version 2.1 or any subsequent comparable version(s).
Controller version 2.1 and any subsequent comparable version(s)	The data cannot be restored on controller version 2.0 or earlier.

Simple Variable Compatibilities

The following table presents the rules followed for restoring simple variables:

Case	Data File Contents	Controller Application Contents	Restore Behavior	EcoStruxure Control Expert Message
1	Var1	Deleted	Var1 is not restored	Var1: this variable no longer exists.
2	-	New Var2	Var2 is kept in memory	-
3	Var3	Var3 was renamed into Var3-3 (name changed)	The restore CANNOT BE PERFORMED due to technical limitations. Var3 is deleted and Var3-3 is a new variable.	Var3: this variable no longer exists.
4	Var4 is a simple compatible type.	Var4 was changed to a different simple type. Source value is compatible with target type.	The Var4 is converted into the new type, which has the same value as source.	Var4: type of the variable is changed (from TYPE1 to TYPE2). Types are compatible.

Case	Data File Contents	Controller Application Contents	Restore Behavior	EcoStruxure Control Expert Message
5	Var5 is a simple compatible type.	Var5 was changed to a different simple type. Source value is not compatible with target type.	The restore CANNOT BE PERFORMED. Var5 is converted into a different type (the target) and its value is either: • longer than target type • not compatible with target type	Var5: type of the variable is changed (from TYPE1 to TYPE2) but value may not be compatible. + Var5: value not compatible for conversion between TYPE1 and TYPE2.
6	Var6 is a simple incompatible type.	Var6 was changed into another type.	Restore of original value CANNOT BE PERFORMED. Var6 retains its application value.	Var6: type of the variable is changed (from TYPE1 to TYPE2). But types are not compatible.

Structured Variable Compatibilities

The following table presents the rules followed for restoring structured variables:

Case	Data File Contents	Controller Application Contents	Restore Behavior	EcoStruxure Control Expert Message
DDT Types				
7	Var7 is a DDT type.	Var7 was changed to a simple type.	The restore CANNOT BE PERFORMED because type is not compatible.	Var7: type of the variable is changed (from TYPE1 to TYPE2). But types are not compatible.
8	Var8 is a simple type.	Var8 was changed to a DDT type.	The restore CANNOT BE PERFORMED because type is not compatible.	Var8: type of the variable is changed (from TYPE1 to TYPE2). But types are not compatible.
9	Var9 is a DDT type.	Var9 is changed to another DDT type.	The restore CANNOT BE PERFORMED because type is not compatible.	Var9: type of the variable is changed (from TYPE1 to TYPE2). But types are not compatible.
10	Var10 is a DDT type.	The DDT type has changed.	Make the mapping of the field (see the cases from 1 to 9 and cases from 11 to 16 for array type sub-fields).	Var10.SubField1: … (see cases 1 to 9 and 11 to 16).
ARRAY Types				
11	Var11 is an array.	Var11 is an array of the same type, same starting index but longer.	Transfer the data file values. The rest of the longer application array is kept.	-

Cas-e	Data File Contents	Controller Application Contents	Restore Behavior	EcoStruxure Control Expert Message
12	Var12 is an array.	Var12 is an array of the same type, same starting index but shorter.	Transfer the data file values until target array is full. The rest of the source array is lost.	Var12: length of ARRAY… array (range # 2) is shorter. Last indexes will be lost.
13	Var13 is an array.	Var13 is an array of the same type, but starting index is lower or a different type. Ending index is the same	The restore CANNOT BE PERFORMED because starting indexes must be identical.	Var13: low bound of ARRAY… array (range #2) has changed. Cannot restore.
14	Var14 is an array.	Var14 is an array of the same type, but starting index is higher or a different type. Ending index is the same	The restore CANNOT BE PERFORMED because starting indexes must be identical.	Var14: high bound of ARRAY… array (range #2) has changed. Cannot restore.
15	Var15 is an array.	Var15 is an array of the same type, but the lower and upper indexes are shifted. The length is the same.	The restore CANNOT BE PERFORMED because starting indexes must be identical.	Var15: bounds of ARRAY… array (range #2) have changed. Cannot restore.
16	Var16 is an array.	The type of the array has changed	See cases 1 to 10 and 7 to 10.	See cases 4 to 10 and 11 to 16.

Elementary Type Compatibilities

BOOL, BYTE, INT, UINT, DINT, UDINT, WORD, and DWORD are compatible types if no lost of value are detected.

Examples are:

- a DINT variable can be changed into an INT variable if only the 2 first bytes of the DINT are used
- an UINT or UDINT variable can be changed into an INT variable if the value is compatible.

REAL, STRING, TOD, TIME, DATE, and DT are not compatible types.

The following table presents the compatibility between the elementary types:

	BOOL	EBOOL	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	REAL	STRING	TOD	TIME	DATE	DT
BOOL		YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
EBOOL	YES		NO	PDL	PDL	NO	PDL	PDL	NO	NO	NO	NO	NO	NO	NO
BYTE	NO	NO		YES	YES	YES	YES	YES	YES	NO	NO	NO	NO	NO	NO
INT	NO	NO	PDL		PDL	YES	YES	YES	YES	NO	NO	NO	NO	NO	NO
UINT	NO	NO	PDL	PDL		YES	YES	YES	YES	NO	NO	NO	NO	NO	NO
WORD	NO	NO	PDL	YES	YES		YES	YES	YES	NO	NO	NO	NO	NO	NO
DINT	NO	NO	PDL	PDL	PDL	PDL		PDL	YES	NO	NO	NO	NO	NO	NO
UDINT	NO	NO	PDL	PDL	PDL	PDL	PDL		YES	NO	NO	NO	NO	NO	NO
DWORD	NO	NO	PDL	PDL	PDL	PDL	YES	YES		NO	NO	NO	NO	NO	NO
REAL	NO	NO	NO	NO	NO	NO	NO	NO	NO		NO	NO	NO	NO	NO
STRING	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO		NO	NO	NO	NO
TOD	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO		NO	NO	NO
TIME	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO		NO	NO
DATE	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO		NO
DT	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	

NOTE: PDL means Possible Data Loss.

Compatibility of Aliases

The following table presents the rules followed for restoring aliases:

Case	Data File Contents	Controller Application Contents	Restore Behavior	EcoStruxure Control Expert Message
1	Map1 is unlocated. %MW100 is used.	Map1 is located on %MW100.	Value of the data file for Map1 is set to %MW100.	Map1: restored at %MW100
2	Map2 is located on %MW100.	Map2 is no longer located on %MW100, %MW100 is used.	Value of data file for Map2 is set at the new address of Map2 (unlocated data area). %MW100 of the application keeps its current value.	Map2: restored into unlocated data area. %MW100: keeps current value.
3	Map3 is located on %MW100.	Map3 is located on %MW200.	Restores the value of %MW100 at address %MW 200.	Map2: restored from %MW100 to %MW200.

Case	Data File Contents	Controller Application Contents	Restore Behavior	EcoStruxure Control Expert Message
	%MW200 is used	%MW100 is used.	%MW100 of the application keeps its value.	%MW100: keeps current value.
4	Map4 is located on %MW100, but %MW area not exported.	Map4 is no longer located on %MW100, %MW100 is used.	Restore CANNOT BE PERFORMED because the source value is not in data file.	Map4: value at address %MW100 is not available.
5	Map5 is unlocated but in an area not exported.	Map5 is located on %MW100.	Restore CANNOT BE PERFORMED because the source value is not in data file.	Map5: value at address %MW100 is not available.
6	Map6 is located at %MW100 and %MW200 is used, but %MW area is not exported.	Map6 is now located on %MW200 and %MW100 is used.	Restore CANNOT BE PERFORMED because the source value is not in data file.	Map6: value at address %MW100 is not available.
7	Map7 does not exist or is an alias.	Map7 is located at %MW100.	Restore CANNOT BE PERFORMED to avoid the lost of the current variable.	Map7: mapped variable that didn't exist at the file generation or was an alias %MW100 value not restored.
8	Topological addresses, %CH, %SW, etc.	-	-	MapVar: variable mapped on an unauthorized address (%CH0.2.0).

Restoration of FB Instances and Other Variables

The following table presents the rules followed for restoring certain special variables:

Type of Entity	EcoStruxure Control Expert Message
FB Instances: for all fields (input, output, I/O, private, public) that is not a pointer, the fields are restored or not by using the rules above. If any field cannot be restored, the FB Instance is not restored	-
SFC variables cannot be restored.	Step1: SFC types (SFCSTEP_STATE) are not allowed for restore Step1: SFC types (SFCCHART_STATE) are not allowed for restore
Hidden variables cannot be restored.	-
IODDT variables cannot be restored.	IOVar: IODDT types (T_ANA...) are unknown so not allowed for restore.
Device DDTs cannot be restored.	DevDDT: Device DDT types (T_...) are not allowed for restore.

Restoration of Forced Variables

NOTE: If forced bits are detected in the controller before performing the restore data, forcing status (F) and value are not updated in the controller after the restore data is performed.

The following table presents the rules followed for restoring forced variables:

Variable	Value when the save data from the controller to file is performed	Value before performing the restore data from file to the controller	Value in the application after the restore data is performed
Located	0	1	0
Located	0	forced to 1	forced to 1
Unlocated	0	1	0
Unlocated	0	forced to 1	forced to 1
Located	1	0	1
Located	forced to 1	0	1(1)
Unlocated	1	0	1
Unlocated	forced to 1	0	1(1)
Located	1	forced to 0	forced to 0
Located	forced to 1	0	1(1)
Unlocated	1	forced to 0	forced to 0
Unlocated	forced to 1	0	1(1)
Located	forced to 0	1	0(1)
Located	forced to 1	forced to 0	forced to 0
Unlocated	forced to 0	1	0(1)
Unlocated	forced to 1	forced to 0	forced to 0

(1) The restored value is not forced.

Transferring Current Values

At a Glance

The functions below apply to the following variables:

- Located and unlocated variables
- Public and Privates variables of DFBs

Update Init Value with Current Value

The **Update Init value with Current Value** function is used to replace the initial values in the PLC of these variables with their current values if their **Save** attribute is checked.

When the initial values are updated in the PLC, a "*" is displayed in the status bar to indicate that the initial values in the PLC are different from the initial values displayed in the variable editor.

Use **PLC > Update Init Values with Current Values** to activate this function.

Update Local Init Values with PLC Init Values

When the Initial values have been modified by the function **Update Init value with Current Value** or by using %S94 in the application, the **Update Local Init Values with PLC Init Values** function updates the initial values on the PC in the variable editor.

After using this function "*" is no longer displayed in the status bar.

Use **PLC > Update Local Init Values with PLC Init Values** to activate this function.

NOTE: to be taken in account in the code a build change is needed.

NOTE: when disconnecting the PLC from Control Expert, to update the STU file with the same values as the PLC, you have to answer Yes, confirm modification, backup the PLC project, connect the PLC again, make a build changes and save the STU file.

Memory usage

At a Glance

The memory usage function is used to view:

- the physical distribution of the PLC memory (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) (internal memory and memory card)
- the space taken up in the memory by a project (data, program, configuration, system)

It can also be used to reorganize the memory where possible.

NOTE: The memory usage screen is not available in simulation mode. This screen is only available in standard mode when you have built the application.

If the application has been built and if it is in NOT BUILT state due to a program modification, the screen is accessible, but it corresponds to the application built previously. Modifications will be taken into account at the next build.

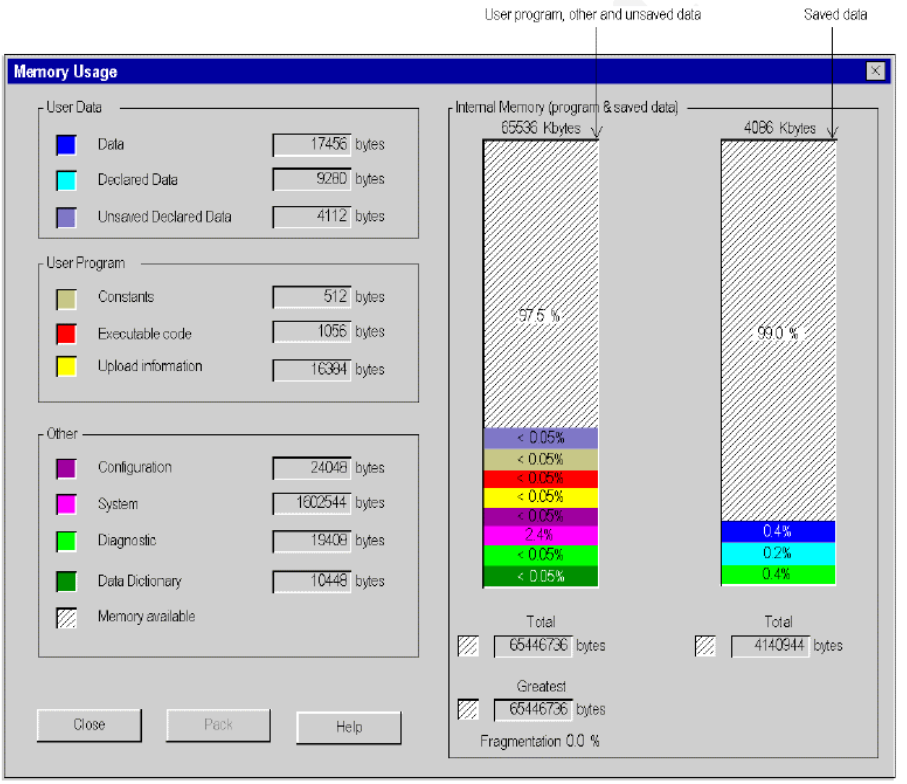
Procedure

Perform these actions to access the memory usage details of the PLC:

Step	Action
1	Activate the **PLC > Memory Consumption...** command, which displays the corresponding screen. The memory usage statistics of a project can only be accessed if you have generated its executable in advance.
2	If you want to reorganize the memory to optimize it, activate the **Pack** command.

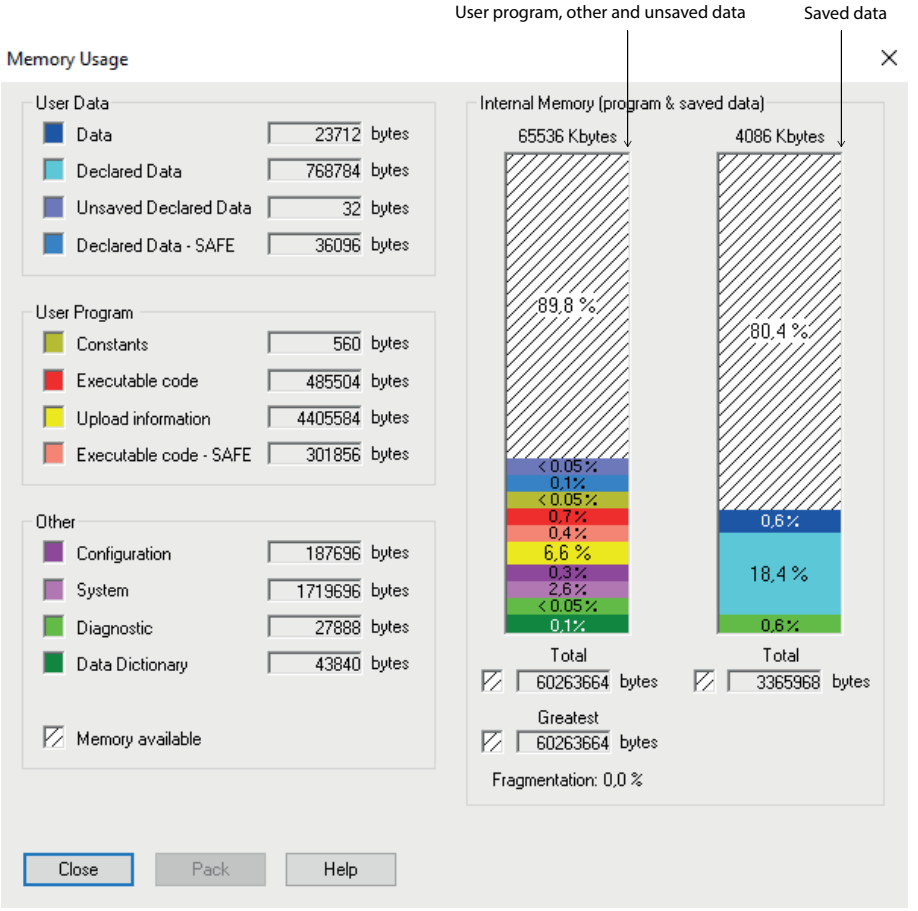
Memory usage screen Modicon M580

The memory usage screen is shown below:



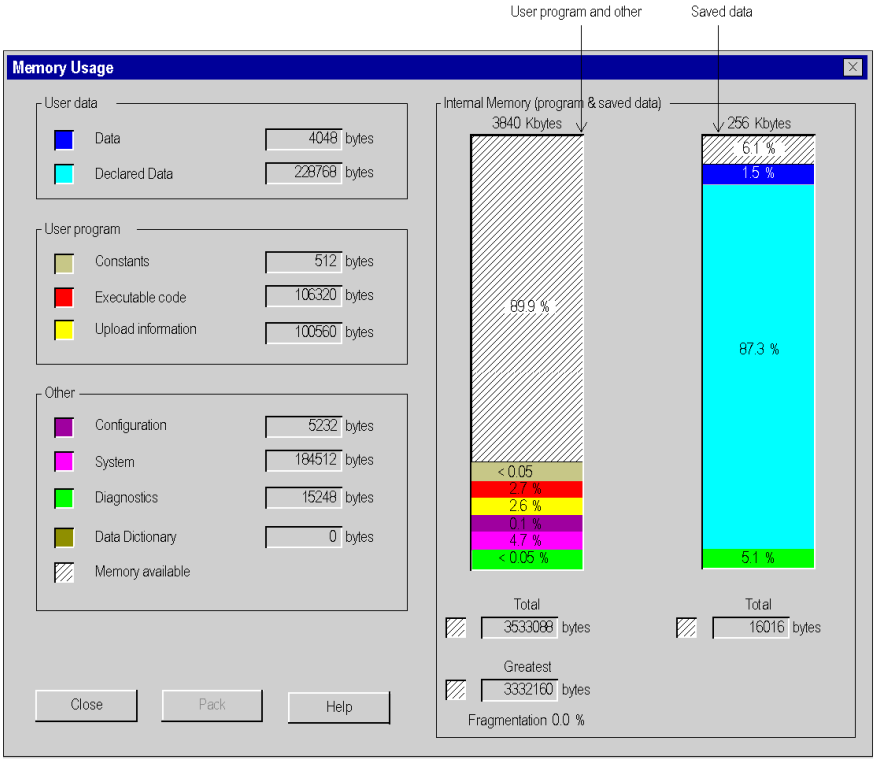
Memory usage screen Modicon M580 Safety

The memory usage screen is shown below:



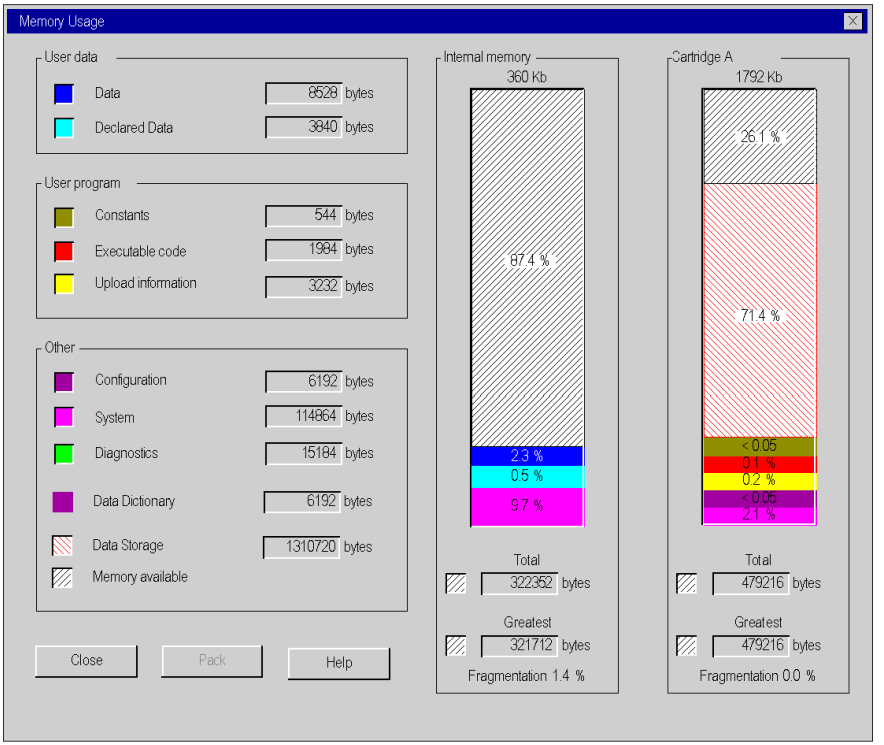
Memory usage screen Modicon M340

The memory usage screen is shown below:



Memory usage screen Premium/Quantum

The memory usage screen of a PLC with a memory card and data storage is shown below. If the PLC does not have a memory card, only the information relating to the internal memory is displayed.



Description of the parameters

The following information fields are available:

Parameter	Description
User Data	This field indicates the memory space (in words) taken up by user data (objects relating to configuration): • **Data**: located data associated with the processor (%M, %MW, %S, %SW, etc.) or the input/output modules, • **Declared Data**: unlocated data (declared in the data editor) saved after power cut. • **Unsaved Declared Data**: unlocated data (declared in the data editor) not saved after power cut (available for the Hot Standby CPU BMEP586040). • **Declared Data – SAFE**: located data associated with the Safety program (only available for Safety controllers).
User Program	This field indicates the memory space (in words) taken up by the project program: • **Constants**: static constants associated with the processor (%KW) and the input/output modules; initial data values, • **Executable code**: executable code of the project program, EFs, EFBs and DFB types, • **Upload information**: information for uploading a project (graphic code of languages, symbols, etc.). • **Executable code – SAFE**: executable code associated with the Safety program (only available for Safety controllers).
Other	This field indicates the memory space (in words) taken up by other data relating to the configuration and the project structure: • **Configuration**: other data relating to configuration (Page0 for a Quantum PLC, hardware configuration, software configuration), • **System**: data used by the operating system (task stack, catalogs, etc.), • **Diagnostic**: information relating to process or system diagnostics, diagnostics buffer, • **Data Dictionary**: Dictionary of symbolized variables with their characteristic (address, type....) • **Memory available**: xxx • **Data Storage**: file and data storage (only for Premium and Quantum).
Internal Memory	This field shows the organization of the PLC's internal memory. It also indicates the memory space available (**Total**), the largest possible contiguous memory space (**Greatest**) and the level of fragmentation (due to online modifications).
Cartridge A / Cartridge B	Only for Premium and Quantum, this field shows the organization of the memory cards. It also indicates the memory space available (**Total**), the largest possible contiguous memory space (**Greatest**) and the level of fragmentation (due to online modifications).
Pack	This command is used to reorganize the memory structure.

Memory re-organization

Memory re-organization is activated using the **Pack** command, page 151.

Memory re-organization can be performed in online or offline mode (Even if the PLC is in Run or in Stop).

NOTE: Certain blocks cannot be moved in online mode. You will attain a lower level of fragmentation by re-organizing the memory in offline mode.

Data Memory Protect

Introduction

Select the **Data memory protection** checkbox to help protect data from a modification request in:

- offline mode
- monitoring mode
- programming mode (when the project status is **DIFFERENT**).

NOTE: You can modify a variable with **Data memory protection** only when EcoStruxure Control Expert is connected in programming mode and the project status is **EQUAL**.

A variable with this protection can be modified only by the program. You cannot modify a variable with this additional protection in monitoring mode through the EcoStruxure Control Expert language editor, animation table, and so on. This is the same behavior for external tools (HMI, SCADA, …).

NOTE: The **Data memory protection** settings can be modified only off-line.

The data protection applies to these data:

- Located data
  - A range of %M and %MW
  - System data %S and %SW
  - Input / output data
- Unlocated data
  - Elementary unlocated data
  - Structured unlocated data
  - Safety-related unlocated data

NOTICE
COMMUNICATION LIMITATION Do not activate the **Data memory protection** function on a variable or range if it will be written by some means other than with the program. **Failure to follow these instructions can result in equipment damage.**

Controller Firmware Limitation

These controllers support the **Data memory protect** functionality:

- M580 controller v3.20 and any subsequent supporting version
- M340 controller v3.30 and any subsequent supporting version

NOTE: Replacing a controller with version earlier than v3.20 for M580 or v3.30 for M340 is possible only if the **Data memory protect** setting for the project is not activated.

Activate the Feature

To activate the **Data memory protect** functionality:

Step	Action
1	Access the **PLC embedded data** group box (**Tools > Project Setting > PLC embedded data**).
2	Select the **Data memory protect** box.
3	Click **Apply** to apply the protections.

Increased Protection of Located Data

The **Data memory protect** function applies to these located data:

- A range of %M and %MW(1)
- System data %S and %SW
- Input / output data:
  - Local I/O objects (LIO)
  - Remote I/O objects (EIO)
  - Distributed I/O objects (DIO): with a topological vision or with a DDT/DDDT vision

(1) For a valid configuration, array variables that are mapped on a %MW range are entirely *inside* or entirely *outside* the %MW range with this additional protection.

The **Data memory protect** function is enabled in the controller configuration window (**Project > Configuration > PLC Bus > CPU**).

For the procedure of the located data protection enforcement, refer to *Protecting Located Data*, page 1359.

Increased Protection of Mapping Addresses

The properties of a located variable have priority on the properties of an unlocated variable. Thus, the unlocated variable that is mapped to a located address automatically takes the protection of this address. The protection of this unlocated variable is reinforced the same way as that of the located variable to which it is mapped.

Depending on the type of the located variable to which the unlocated variable is mapped, the following protection enforcement of the mapping address is applied:

- On %MW/%M flat: the mapping address is not modifiable.

  The attribute with the reinforced protection depends on the %MWi/%M protect configuration
- On %SW/%S: the mapping address is not modifiable.

  The attribute with the reinforced protection depends on the %SWi/%Si protect configuration
- On a topological object: the mapping address is not modifiable.

  The attribute with the reinforced protection depends on the I/O protect configuration

You cannot modify this automatic reinforced protection in the **Data Editor** window.

Data Protect Attribute

Use the **Data Protect** attribute to help protect of unlocated data in the **Data Editor** window.

This attribute is enabled when these conditions are met:

- The variable is not mapped to a located address.
- The variable is not managed.
- The variable is not a sub-field of a structure (DDT, array EFB instance, DFB instance).
- The variable is not a Device DDT.
- The variable is not a constant.

The **Data Protect** attribute is disabled in all other cases.

Increased Protection of Unlocated Data

General Description

The data protection function applies to these unlocated data:

- Elementary
- Structured
- Safety-related

The unlocated data memory protection function is enabled in the **Data Editor** window. By default, unlocated data are not protected.

Consider these conditions when you reinforce the protection of unlocated data:

- An alias variable takes the same protection as the aliased variable.
- A reference variable has always the reinforced protection; it is not affected by the protection of the referenced variable.
- The data protection can also be applied to DDT, Device DDT and IODDT data; and it is enabled in the I/O protection checkbox in the controller configuration window.

The data memory protection function is also available for the Program Unit data:

- Input data:
  - The data memory protection is applied to the input data if there is no effective parameter on it.
  - Input data becomes “Read” when an effective parameter is set.
- Output data:
  - The data memory protection is not applied to the output data.
  - Output data becomes “Read” when an effective parameter is set.
- Input/output data takes the protection of its effective parameter.

Protecting Elementary Unlocated Data: The protection of unlocated elementary data is reinforced individually on an element-by-element basis.

Protecting Structured Unlocated Data: The protection of structures of unlocated data can be reinforced globally. The modification is done once for the entire structure and applies to all sub-fields except sub-fields that are read-only by intrinsic access right.

The data protection function applies to these unlocated data structures:

- DDT
- Device DDT
- IODDT
- Array
- EFB instance

- DFB instance

  NOTE: The protection enforcement of DDT, Device DDT and IODDT unlocated data can be enabled in the I/O protection check box of the controller configuration window.

Protecting Safety-related Unlocated Data

In safety-related applications, the data protection function applies to the following unlocated data:

- Global variables
- Process resource data:
  - Input variables (with or without an effective parameter)
  - Output variables (with or without an effective parameter)
  - Private variables
  - FB Instances
  - DDT and device DDT
- Safety-related resource data:
  - Input variables (with or without an effective parameter)
  - Output variables (with or without an effective parameter)
  - Private variables
  - Safety-related FB Instances
  - Safety-related DDT and device DDT

WRITE PROTECT Attribute

You cannot set the **WRITE PROTECT** attribute in the Program Unit because write protection does not impact variables that you cannot modify (such as read-only variables in the Program Unit).

Consider these other characteristics of write protection for variables:

- You *can* enable write protection for an input variable when there is no effective parameter on it.
- You *can* enable write protection for a private or public variable.
- You *cannot* enable write protection for an output or input/output variable because these are read-only variables.
- You *cannot* enable write protection for an external variable because it inherits the attribute of the reference.

Reinforced Protection of Unlocated Data

Define unlocated data for reinforced protection:

Step	Action
1	In the EcoStruxure Control Expert main window, activate the data memory protection feature, as described above.
2	In the EcoStruxure Control Expert **Project Browser**, double-click **Variables & FB Instances** to open the **Data Editor** window, or click **Tools > Data Editor** in the EcoStruxure Control Expert toolbar.
3	In the **Data Protect** column, select (check) the boxes that correspond to data for which protection is reinforced. **NOTE:** When you select the **Data Editor** attribute of a structured unlocated variable, this action applies to the subfields of the selected structure.

NOTE: The unlocated variable that is mapped on a located address automatically takes the protection of this address. Thus, the protection of this unlocated variable becomes reinforced the same way as that of the located variable to which it is mapped.

Function Pack for Memory

At a Glance

With the **Pack** command it is possible to reorganize the memory so as to optimize it.

To avoid certain detected errors (Internal Memory full or Cartridge Memory full) when you do a build after online modifications, the **Pack** command must be carried out before the online modifications.

NOTE: When the **MEM** status of the Status bar becomes **Red**, it is recommended to make a **Pack** command by clicking on the **Pack** button in the Memory usage screen, page 144 of the PLC.

The necessity of **Packing** the internal (or cartridge) memory is evaluated by the PLC operating system and recommended when:

- the degree of fragmentation (due to many online modifications for instance) of the memory is becoming high,
- and/or the size of the largest contiguous block of available memory is becoming low compared to the total size of available memory.

NOTE: For Quantum PLCs, the **Pack** command is applicable as of OS version 2.3.

Elements that Require Large Amounts of Memory

Certain project elements and actions consume a large amount of memory:

- configuration with a Cartridge memory:
  - with a large section,
  - with several actions in the same SFC section.
- configuration with an internal memory:
  - with a large DFB instances,
- configuration with an internal memory in a Hot StandBy project:
  - several DFB instances,
  - adding or deleting DFB instances,
  - modifying an SFC section.

Procedure

Perform the following actions to use the **Pack** command:

Step	Action
1	Activate the **PLC > Memory Consumption:** command, which displays the corresponding screen.
2	In the Internal Memory or Cartridge area check the percentage values of: • **Memory available**, • **Fragmentation**, • **Greatest** / **total of Memory available**.
3	Check if these values are in the ranges listed in the tables below.
4	If Yes, click on the **Pack** button.
5	Above the **Pack** button, if the warning **Memory optimization is not terminated, please continue to pack** appears, the **Pack** button must be clicked again.
6	The **Pack** function is finished when: • the **Pack** button is grayed, • the **Fragmentation** field is equal to 0, • the **Greatest** value is near the **Total Memory available** value.

How the PLC detects it's recommended to make a Pack command for Internal Memory

The following table shows how the PLC detects it's recommended to make a **Pack** command, regarding the field values in the Internal memory area:

If the Percentage of Memory available is,	**then if the Percentage of Fragmentation is,**	**OR if Greatest (1) / Total Available Memory is:**
>25%	>15%	<50%
[15% ... 25%]	>10%	<80%
<15%	>5%	<90%
(1) Greatest is the Largest free contiguous block of memory available, this information is given in the Memory usage screen.		

How the PLC detects it's recommended to make a Pack command for Cartridge Memory

The following table shows how the PLC detects it's recommended to make a **Pack** command, regarding the field values in the Cartridge memory area:

If the Percentage of Memory available is,	**then if the Percentage of Fragmentation is,**	**OR if Greatest (1) / Total Available Memory is:**
>25%	>10%	<70%
[15% ... 25%]	>5%	<90%
<15%	>3%	<95%
(1) Greatest is the Largest free contiguous block of memory available, this information is given in the Memory usage screen.		

Memory Card Access for Modicon M340

Overview

BMX RMS ••••• memory cards are used to save projects, Web pages, and data in general.

They are mainly used when inserted into a PLC. They may also be read with an SD-card drive or reader.

Memory Card in the PLC

When the memory card is in the Modicon M340, it may be accessed as follows:

- with the PLC, automatically
- with the Control Expert commands, page 122 **PLC > Project Backup... > ....**
- by programming with the Memory Card File Management EFBs (See Memory Card File Management in the System library)
- with an FTP, page 155 client (for processors with an Ethernet connection) to manage files on the memory card

Memory Card in an SD-card Drive or Reader

When the memory card is inserted into an SD-card drive or reader, it may be accessed as a data storage medium (for example, a USB key or hard drive). The files on the memory card may thus be viewed.

NOTE: In order to read the memory card in an SD-card drive or reader, install the Reliance driver first. This driver is available on the Automation Device Maintenance CD-ROM.

Tree Structure of Files on the Memory Card

When the memory card is used in an SD-card drive or reader, or via FTP, its contents are accessible with a file explorer. The tree structure shown includes three directories:

- *DataStorage*, which includes all data files from Memory Card File Management EFBs (see EcoStruxure™ Control Expert, System, Block Library)
- *Firmware*, which includes all files used by the Automation Device Maintenance software
- *Web*, which includes all Web pages (see Modicon M340 for Ethernet, Communications Modules and Processors, User Manual)

It is also possible to create directories for personal files.

NOTE: The memory zone for project files is not accessible by users.

FTP and Memory Card for Modicon M340

Overview

With an FTP client, it is possible to transfer files between the memory card in the Modicon M340 PLC (with an embedded Ethernet port connection), a destination, and/or a source such as a hard drive or server.

Any FTP client installed on your computer may be used. We recommend using Filezilla or Microsoft Explorer.

You can transfer:

- data files used by the memory card file management EFBs
- web pages used by the HTTP server
- personal files that are useful for your project

NOTE: Downloading write-protected files to the memory card can prevent the Automation Device Maintenance software from properly upgrading the module. Some FTP clients (for example, the Windows Explorer client) cannot remove write-protected files from the card. You can delete write-protected files from the card with some FTP clients (for instance, FileZilla) that are available for free download over the Internet.

Using FTP

The following table describes how to use FTP with Internet Explorer.

Step	Action
1	Open Internet Explorer.
2	Input the FTP address in the Address field. Example: ftp://login:password@FTP_Server_IP_Address • Login: datastorage • Password: datadownload **Result**: The contents of the memory card are displayed.
3	Copy and/or paste the files between the memory card and your desired location.

The following modules have a **time out value of 15 minutes** and a **maximum number of eight sessions**:

- BMX P34 20•0
- BMX NOE 01•0
- BMX NOC 0401

- BMX NOR 0200
- BME P58 ••••
- BME NOC 03•1
- TSX P57 •634
- TSX ETY •10••
- TSX ETY PORT
- 140 CPU 651 •0
- 140 NOE 771 ••
- 140 NOC 78• 00

Project Management with DTMs

At a Glance

The Control Expert Project and Archive files can be saved and opened even if DTMs are not installed in the DTM Hardware Catalog, page 270.

Building a complete Control Expert project requires that all DTMs in the project are installed.

A DTM audit tool provides a comparison between the version of the project DTMs and the PC DTMs.

Opening STU and STA Files

Opening the *.stu Project file and *.sta Archive files restores the DTM topology tree and device configurations.

If there are DTMs in the restored project that are not installed on the host PC, there is no warning message. Services such as Rebuild all or Check devices, page 294 must be used to check for uninstalled DTMs.

Saving STU and STA Files

The DTM topology tree and device configurations are saved in these files.

ZEF Files

The DTM topology tree and device configurations is saved and restored in the *.zef import/export files.

XEF Files

The DTM topology tree and device configurations cannot be saved nor restored in the *.xef import/export files.

DTM Audit Tool

A tool is provided to have a clear vision of the DTM versions embedded in a project and the versions installed on the PC. Depending on the DTM versions compatibility, the tool informs if the project can be opened and built.

File format compatible with the DTM audit tool:

- *.stu project file
- *.sta archive file
- *.zef import/export file

Launching the DTM audit tool:

- **Start > Programs > EcoStruxure Control Expert > DTMAuditTool**. The tool can be launch even if Control Expert is not running.
- In Control Expert, select **Tools > DTMAuditTool**.

Opening a project (*.stu or *.sta) or an inport/export file (*.zef) in Control Expert: the DTM audit tool analyzes the DTM versions compatibility.

- If they are compatible, the tool does not display any message.
- If one or more DTM versions are not compatible, the tool displays a table with the DTM versions and compatibilities.

  Then you can choose to cancel the project opening or continue, being informed that DTM versions are not compatible and that the project cannot be built.

When DTMs are available on Schneider Electric DTM Catalog Server, a link is available to download and install the required DTM versions. It is then possible to download and install the DTM from this link.

After download and installation of the DTM, the Control Expert DTM catalog must be updated following one of these procedures:

- close and open again Control Expert (the DTM catalog is then automatically updated).

- select **Tools > Hardware Catalog**, select the **DTM catalog** tab and click on the **Update** button.

  NOTE: Although DTM from EDS file is already installed on the PC, the DTM Audit tool informs you to add the missing EDS file to the library and the project cannot be built. This behavior results from a difference between the Device Name created during the EDS file import to the DTM Hardware Catalog (with or without the setting **New Naming Convention**) and the Device Name used in the DTM browser of your project.

To fix this mismatch:

- Remove the inconsistent EDS file from the DTM Hardware Catalog.
- close and open again Control Expert
- Re-import the EDS file to match the Device Name in the application (with or without the setting **New Naming Convention**.
- Update the DTM catalog

For more detailed information about removing or importing EDS file, refer to chapter DTM Hardware Catalog, page 270.

Complete Build Management

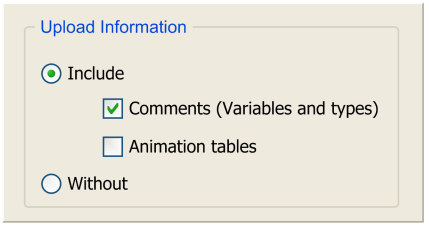
Rebuild all project, page 81 (DTM modifications or not) requires that all DTMs in the project are installed on the PC because the **Check devices** service is part a complete build.

If any DTMs are not found, a detected error is displayed in 2 places:

- The Build tab in the Output Window
- DTM topology tree

Partial Build Management

The user can choose not to include the DTM configuration in a PCL binary:



A partial project build that does not take into account the DTM configurations does not call the **Check devices** service, therefore, this build does not need to have all the DTMs installed on the PC.

Upload Project

The Control Expert Transfer Project from PLC, page 84 (Upload project) function uploads the current DTM topology tree and the device configurations stored in the PLC (but not the information in the Profibus Remote Bus Master and the devices). This function is not the same as the Load data from device DTM, page 297 service.

Some DTMs have a Compare function that allows comparing the configuration in the device to the configuration in the PLC.

Download Project

To use the Control Expert **Transfer Project to PLC** (Download project) function with DTMs:

Step	Action
1	Construct a DTM topology tree.
2	Use the Store to device, page 297 service on each DTM that has a physical device on the network.
3	Call the **Transfer Project to PLC** function to download the Control Expert project that includes a DTM topology tree and device configurations.

Project browser

What's in This Chapter

Introduction to the project browser 161
Introduction to the various directories of the project browser 203
Summaries of the utilities associated with the structure view and function view 225

Subject of this chapter

This chapter introduces the Control Expert browser that allows you to move around your project, via the structure view or function view of its directory tree.

Introduction to the project browser

About this sub-chapter

This sub-chapter provides general information on the project browser.

Introduction to the project browser

General points

The project browser allows you to display the contents of a Control Expert project and to move around its various elements: configuration, data, program, etc. To do this you can display your project in 2 different ways:

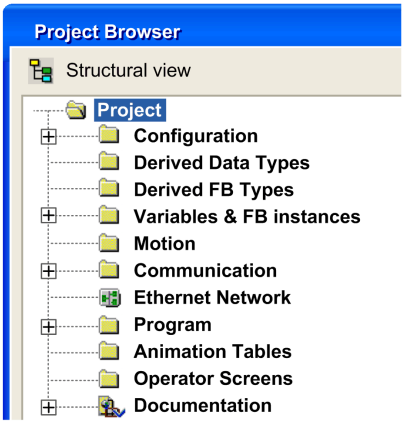
- **Structural View**,
- **Functional View**.

Structural view

The structural view displays the directory tree of the Control Expert project and allows you to navigate within this with direct access:

- to the configuration,
- to the DDT and DFB types,
- to the variables (EDT, DDT, DFB and EFB instances),
- to motion functions,
- to communication functions,
- to the program,
- to the animation tables,
- to the operator screens,
- to the project documentation (title page, general information).

The following illustration shows the structural view of a project:



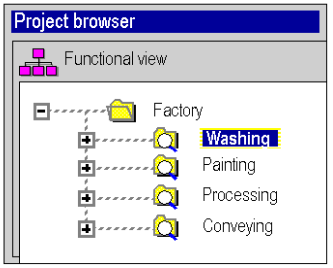
By default, the browser displays the second level of the directory tree. To access the other levels, you need to expand the directories.

NOTE: The default project directory name is **Project**. You can change this name by accessing the project properties dialog box using the **Properties** command from the contextual menu.

Functional view

The functional view displays the directory tree of the project, broken down into functional modules , page 237. This breakdown does not take into account the order of execution of the program by the PLC.

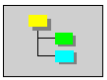
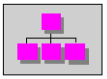
The following illustration shows the functional view of a project:



By default, the browser displays the first level of the directory tree. To access the other levels, you need to expand the directories.

Navigating between the functional view and structural view.

In the toolbar of the project browser you will find the following icons that allow you to display the various project views:

	Allows you to display the structural view of the project.
	Allows you to display the functional view of the project.
	Allows you to display the structural view and the functional view in series, page 164.
	Allows you to display the structural view and the functional view in parallel, page 165.

Structural view and functional view of the project

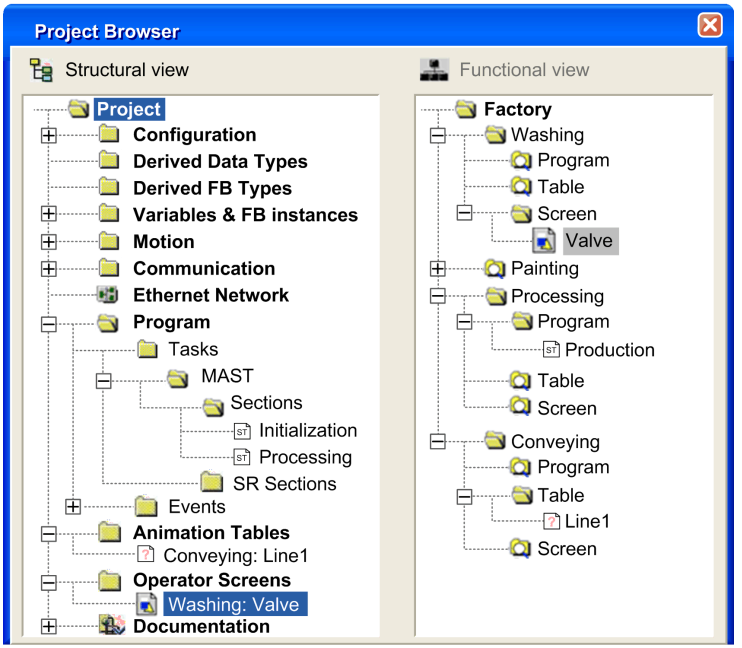
At a Glance

The project browser allows you to simultaneously display the structural view and functional view of the project. You can choose to display these views:

- in series. Both views are displayed, one next to the other,
- in parallel. Both views are displayed, one above the other.

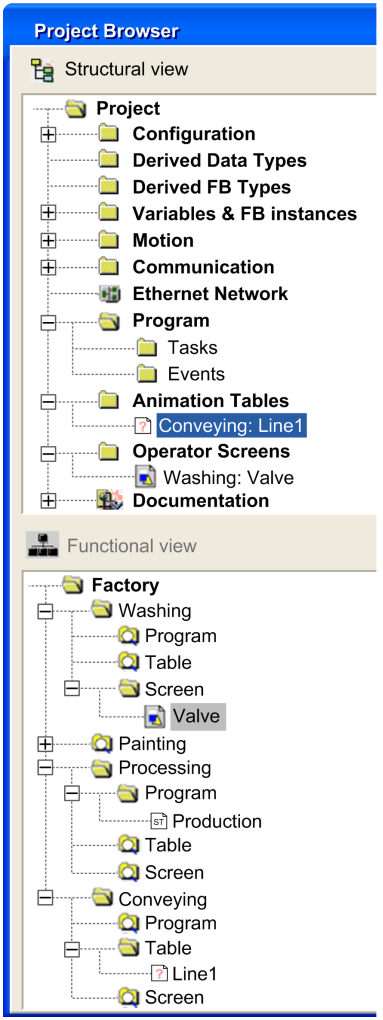
Display of both views in series

The following illustration shows a series display of the structural view and functional view:



Display of both views in parallel

The following illustration shows a parallel display of the structure and functional views:



The Zoom In and Zoom Out commands

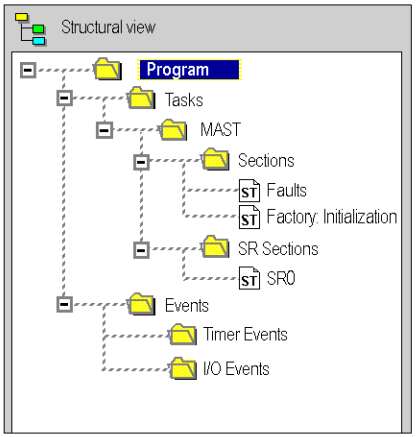
At a Glance

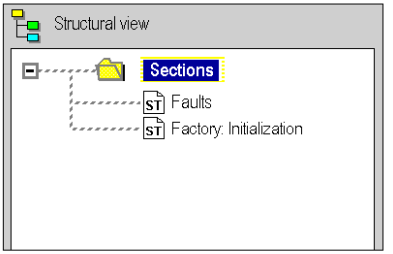
In the functional view and structural view of the project browser, you can display a specific directory tree level by using the **Zoom In** command. You can apply this command in a level below the one you are in. This command is no longer available once you cannot expand the tree structure any further.

The **Zoom Out** command allows you to go back to the previous display of the project browser.

Displaying a specific level of the directory tree

The following table shows you how to use the **Zoom in** command in the structural view:

Step	Action
1	In the project browser select the directory you wish to display (for example **Program**). **Result**: The selected element appears highlighted.
2	Activate the **Zoom In** command from the **Display** menu or from the contextual menu. **Result**: Only the selected directory appears in the structural view (for example **Program**): 

Step	Action
3	From the directory (for example **Program**) select the directory (for example **Sections**) that you wish to display. **Result**: The selected element appears highlighted.
4	Activate the **Zoom In** command from the **Display** menu or from the contextual menu. **Result**: Only the selected directory appears in the structural view (for example **Sections**): 

NOTE: These steps can also be used with a functional view.

Returning to the structural view of the project

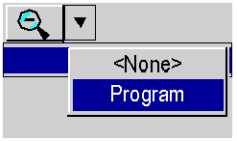
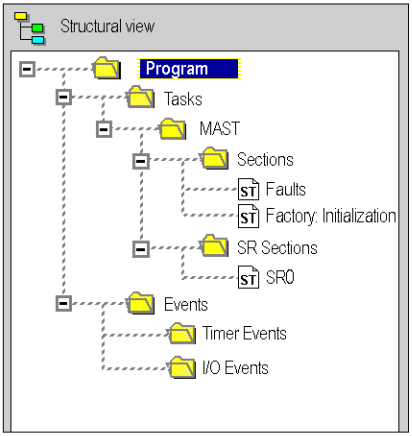
The following table shows you how to use the **Zoom Out** command in the structural view:

Step	Action
1	In the project browser select the directory from the structural view. **Result**: The selected element appears highlighted.
2	Activate the **Zoom Out** command from the **Display** menu or from the contextual menu or by clicking on the  icon. **Result**: The structural view appears along with the directory tree of the project.

NOTE: These steps can also be used with a functional view.

Returning to directory tree level

The following table shows you how to use **Zoom Out** command in a lower level of the directory tree of the structural view:

Step	Action
1	In the project browser select the directory from the structural view (for example **Sections**). **Result**: The selected element appears highlighted.
2	Expand the menu using the arrow to the right of the  icon. **Result**: The menu with the choice of different views appears.
3	In the menu (for example **Program**) select the structural view to be displayed. **Result**: The directory (for example **Program**) appears in the structural view:  Selecting **None** displays the entire structural view of the project.

NOTE: These steps can also be used with a functional view.

The commands Contract all and Expand all

At a Glance

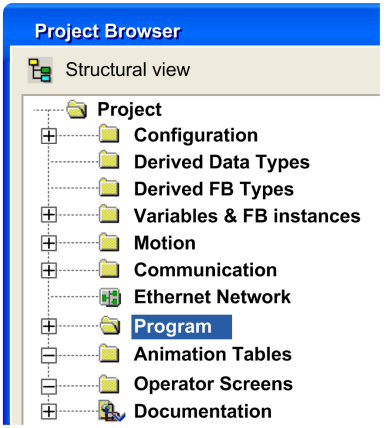
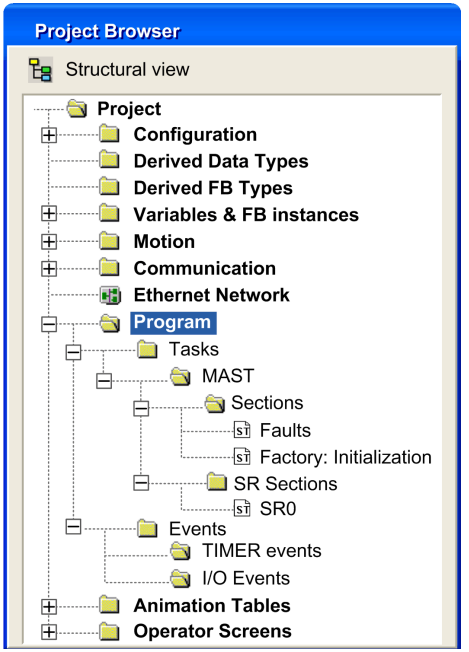
In the functional view and structural view of the project browser, you can display a specific directory tree level and its sub-directories by using the **Expand all** command.

The command **Contract all** allows you to contract an entire directory.

NOTE: The following steps show how to use the **Contract all** and **Expand all** commands with a structural view. These steps can also be used with a functional view.

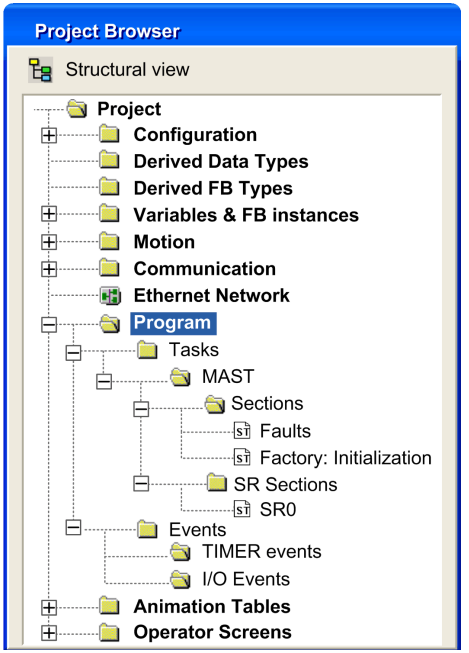
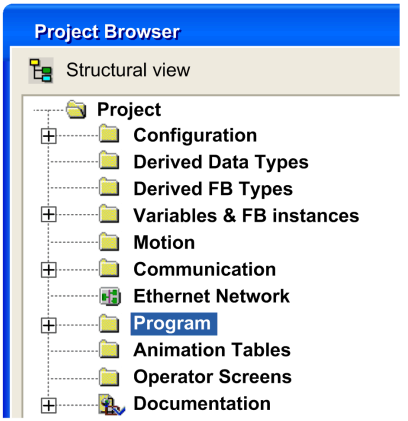
How to expand a directory

The following table shows you how to use the **Expand all** command in the structural view:

Step	Action
1	In the project browser select the directory you wish to expand (for example **Program**). **Result**: The selected element appears highlighted: 
2	Activate the **Expand all** command from the **Display** menu or from the contextual menu. **Result**: The selected directory is expanded completely: 

How to contract a directory

The following table shows you how to use the **Contract all** command in the structural view:

Step	Action
1	In the project browser select the directory you wish to contract (for example **Program**). **Result**: The selected element appears highlighted: 
2	Activate the **Contract all** command from the **Display** menu or from the contextual menu. **Result**: The selected directory is contracted: 

The Goto command

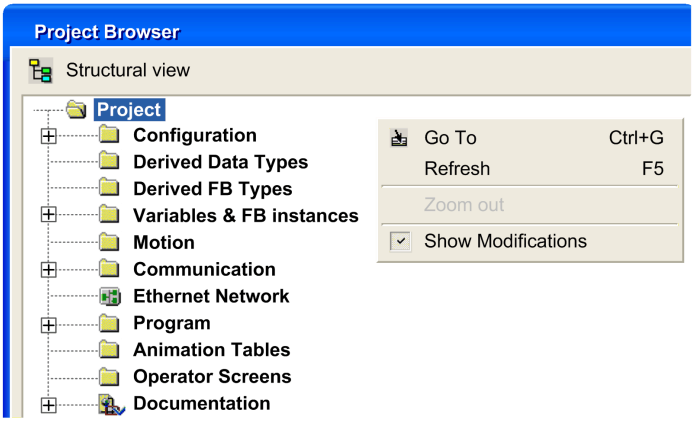
At a Glance

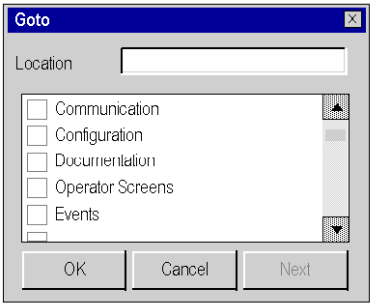
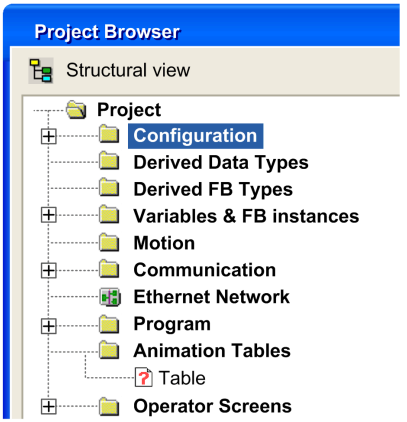
The **Goto** command allows you to access a desired location in the project browser.

In the functional view or structural view, the levels are not always expanded. The **Goto** command allows you to go directly to what you are looking for.

Finding an element

The following table shows you how to use the **Goto** command in the structural view of the browser:

Step	Action
1	Click the right mouse button on the background of the structural view. **Result**: The contextual menu appears. 
2	Select the **Goto** command from the menu.

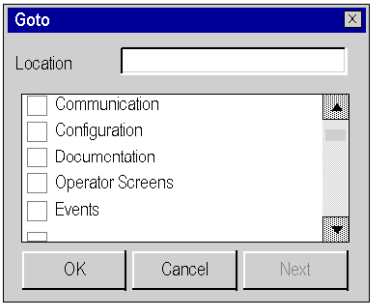
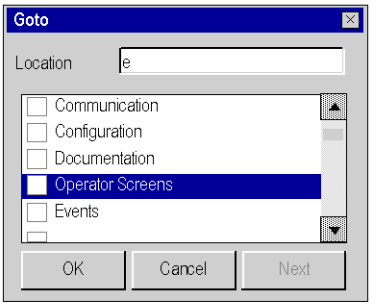
Step	Action
	Result: The **Go to** window appears. 
3	In the window, double-click the element you want to go to. **Result**: The element appears shaded in the structural view (Configuration in the example). 
4	In the **Goto** window confirm your choice with **OK**.

Elements with the same name

Where one or more elements have the same name (examples: sections). In the **Go to** window the element appears with a marker. ☑. The **Next** command in the **Go to** window is unshaded. This allows you to pinpoint one by one the different elements with the same name when the **Go to** window is active.

Fast access to an element

In order to access an element in the window more quickly in the **Go to** window, you can use the **Location** function. Typing in the first letter of the element gives you access to the elements beginning with these letters. The following table shows you how to use the **Location** function in the structural view of the browser:

Step	Action
1	Click the right mouse button on the background of the structural view.
2	Select the **Goto** command from the menu. **Result**: The **Go to** window appears. 
3	In the **Location** field type the first letter of the element you want to go to. **Result**: The element appears highlighted in the **Go to** window. 
4	In the **Go to** window confirm your choice with **OK**.

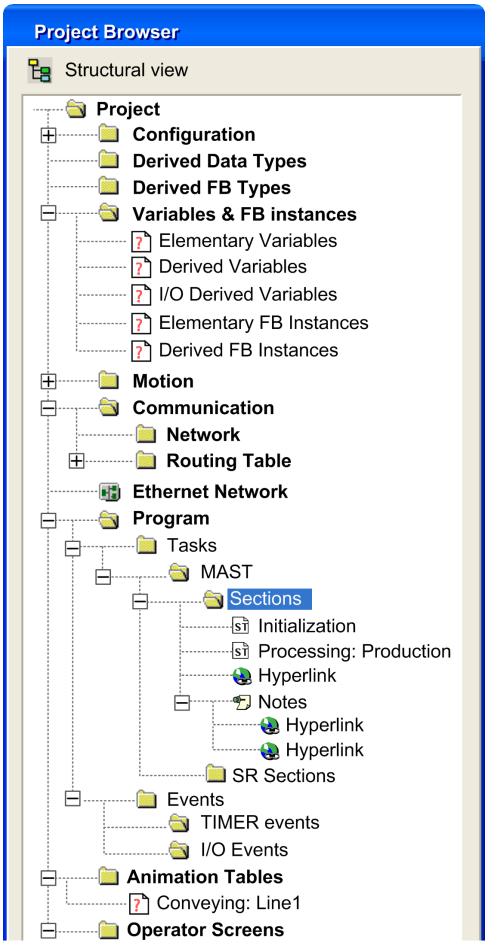
User Directory

At a Glance

In the **Project** directory and in all the directories of the project browser, you can create hyperlinks, page 1774 and user directories for grouping of hyperlinks.

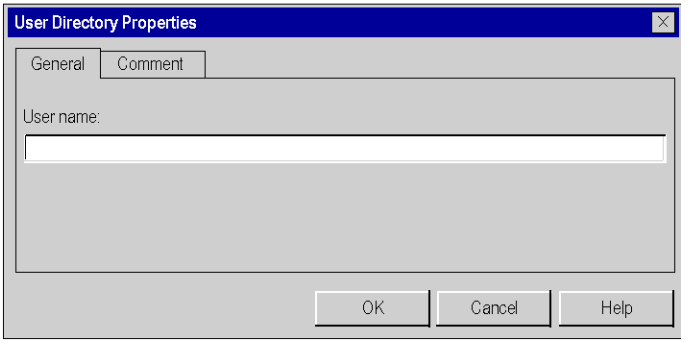
NOTE: You cannot nest a user directory in a user directory and a hyperlink in a hyperlink.

The following illustration shows a hyperlink and a user directory in the directory **Program** :



Creating a User Directory

The following table shows you how to create a user directory in the structural view of the browser:

Step	Action
1	In the project browser select the directory in which you wish to create a user directory. **Result**: The selected element appears highlighted.
2	Select the **Add User Directory.. command from the popup menu**.. **Result**: The **User Directory Properties** window appears: 
3	Give the user directory a name in the **General** tab.
4	In the **Comment** tab enter the comment associated with the user directory.
5	Click **OK** to confirm your choice. **Result**: The user directory appears in the project browser.
6	You can now create hyperlinks, page 1774 in the directory provided for this purpose.

Project Properties

At a Glance

Whichever view is displayed, you can access the project properties by using the **Properties** command from the contextual menu (which can be reached by right-clicking the mouse on the **Project** folder). This gives you access to a dialog box which allows you to:

- access information on the current project,
- define certain parameters of the project.

Description of the Tabs

The dialog box includes the following tabs and actions:

Tab	Description
General	**Name**: allows you to define the name of the project. **Type**: indicates the type of Control Expert software used. **Libset version**: indicates the FB library version used for your current project.
Program & Safety Protection	Password setting applies to the following: • **Sections & Program Units**: sections password settings, page 192 • **Safety**(1): safety password settings, page 186
Project & Controller Protection	Password setting applies to the following: • **Application**(2): application password settings, page 179 • **Firmware**(3): firmware password settings, page 190 • **Data Storage**(4) or **Web Diagnostics / Data Storage**(5): data storage password settings, page 195
Identification	Allows you to identify the project: • current version with the option of automatic increment • creation date • generation date • Last rebuild all • Last partial build • SourceSafeSignature (safety PACs) When **Automatically incremented** is set: • Major and Minor counters are not active • Build counter is incremented at build • Last rebuild All date is updated at Rebuild All • Last partial build date is updated at Rebuild All and Build Change When **Automatically incremented** is not set: • Major, Minor and Build counters can be incremented manually by the user • Last rebuild All date is updated at Rebuild All • Last partial build date is updated at Rebuild All and Build Change **NOTE: Automatically incremented** is set by default at project creation. It can be deactivated by default when the option **Auto incrementation of project version** is deselected (menu: **Tools > Options > General)**.

Tab	Description
Comment	Allows a comment to be associated with the project.
1. Only for Modicon M580 Safety CPUs. **2.** Except for Modicon Momentum. **3.** Only for Modicon M580 CPUs as of OS version 2.00, BME NOC 03••, and BME CXM 0100 modules. **4.** Only for Modicon M580 CPUs. **5.** Only for Modicon M580 CPUs as of firmware version.	

NOTE: By default, the dialog box is called **Properties of Project**. If you give the project another name (**myproject** for example), this dialog box is called **Properties of myproject**.

Application Protection

Overview

EcoStruxure Control Expert has a password mechanism to help guard against unauthorized access to the application.

EcoStruxure Control Expert password protects these actions:

- Open the application in EcoStruxure Control Expert.
- Connect to the controller in EcoStruxure Control Expert.

Setting an application password helps prevent unwanted application modification, download, or opening of application files. The password is encrypted within the application.

In addition to setting the password, you can encrypt `.STU`, `.STA`, and `.ZEF` files. The file encryption feature in EcoStruxure Control Expert helps prevent modifications and reinforces the protection of intellectual property. The file encryption option is protected by a password mechanism.

NOTE: When a controller is managed as part of a system project, the application password and file encryption are disabled in Control Expert editor and managed with Topology Manager.

Password Construction

The construction of the password conforms to the IEEE 1686–2013 standard.

A valid password contains at least 8 characters and includes at least one uppercase letter, one lowercase letter, one number, and one non-alphanumeric character ($, %, &, etc.).

NOTE: The application password is cleared when you export a non-encrypted project to an `.XEF` or `.ZEF` file.

Create a New Project

By default, a new EcoStruxure Control Expert Classic application (project) has these characteristics:

- The project is not protected by a password.
- The project application files are not encrypted.

When you create a project, you can exercise these options in the **Security enforcement** dialog box:

- Set a password for the application.
- Apply encryption to your application files through a file-encryption password.

Access the **Security enforcement** dialog box in EcoStruxure Control Expert Classic:

Step	Action
1	Open the **New Project** window in EcoStruxure Control Expert (**File > New**).
2	Select a controller for the project.
3	Click the **OK** button to open the **Security enforcement** dialog box.
4	Choose to create a project with or without passwords and follow the directions in the appropriate table below.

No Passwords: Create a project without an application password:

Step	Action
1	Access the **Security enforcement** dialog box.
2	In the **Security enforcement** dialog box, select **I do not want to set an application password for this project**.
3	Click the **OK** button to continue.

With Passwords: Use the directions below to create a project with an application password and an (optional) file encryption password.

NOTE: You can configure an expiration date for these passwords on the **Policies** tab in the Security Editor.

Step	Action
1	Access the **Security enforcement** dialog box.
2	In the **Application password** group box, create a password to help protect the application and increase the security for access to the controller: • Enter a password in the **Entry** field. • Reenter the password in the **Confirmation** field.
3	In the **File encryption password** group box, create a password to help protect the intellectual property: • Enter a password in the **Entry** field. • Reenter the password in the **Confirmation** field. **NOTE:** • You can configure a file-encryption password only after you configure an application password. • Use different passwords for the application password and the file-encryption password.
4	Press the **OK** button to apply your password settings and close the **Security enforcement** dialog box.

NOTE:

- If no password is entered, the application files are not encrypted. In this case, the next time you open your EcoStruxure Control Expert project, the **Password** dialog box opens. To access your project, do not type a password and click **OK**. Then use the instructions below to set an application password and enable file encryption.
- You can create or change an application password at any time, but you cannot clear the application password when a file-encryption password is configured for the project.

Set an Application Password

Set the application password:

Step	Action
1	Right-click **Project** in the **Project Browser**.
2	Select **Properties** from the pop-up menu to open the **Properties of Project** window.
3	Select the **Project & Controller Protection** tab.

Step	Action
4	In the **Application** field, click **Change password ...** to open the **Modify Password** window.
5	Enter the new password in the **Entry** field.
6	Enter the confirmation of the new password in the **Confirmation** field.
7	Click **OK** to confirm.
8	Click **OK** or **Apply** in the **Properties of Project** window to confirm the changes.

Change the Application Password

Change the application protection password:

Step	Action
1	Right-click **Project** in the **Project Browser**.
2	Select **Properties** command from the pop-up menu to open the **Properties of Project** window.
3	Select **Project & Controller Protection** tab.
4	In the **Application** field, click **Change password ...** to open the **Modify Password** window.
5	Enter previous password in the **Old password** field.
6	Enter the new password in the **Entry** field.
7	Enter the confirmation of the new password in the **Confirmation** field.
8	Click **OK** to confirm.
9	Click **OK** or **Apply** in the **Properties of Project** window to confirm the changes.

Delete the Application Password

Clearing the application password is not allowed while file encryption is enabled.

Clear the application protection password:

Step	Action
1	Right-click **Project** in the **Project Browser**.
2	Select the **Properties** command from the pop-up menu to open the **Properties of Project** window.
3	Select the **Project & Controller Protection** tab.
4	In the **Application** field, click **Clear password...** to open the **Password** window.

Step	Action
5	Enter the password in the **Password** field.
6	Click **OK** to confirm.
7	Click **OK** or **Apply** in the **Properties of Project** window to confirm the changes.

Auto-Lock Feature

To exercise the option to limit access to the EcoStruxure Control Expert tool after a configured time of inactivity, select (check) the **Auto-lock** checkbox and enter a value in the **Minutes before lock** field to set the timeout for the time of inactivity.

An optional auto-lock feature limits access to the EcoStruxure Control Expert software programming tool after a configured time of inactivity. You can activate the auto-lock feature with the **Auto-lock** checkbox and select the time-out for the time of inactivity with **Minutes before lock**.

If the auto-lock feature is enabled and the configured inactivity time elapses, a dialog box for the application password appears. Behind the dialog box, editors remain open in the same position. As a result, anyone can read the content of the EcoStruxure Control Expert windows but cannot continue to work with EcoStruxure Control Expert.

NOTE: If you have not assigned a password to the project, the dialog box is not displayed.

Password Request Condition

Opening an existing application (project):

When you open an application file, an **Application Password** dialog box opens.

Enter the password and click **OK**.

Result: If the password is correct, the application opens. If the password is incorrect, an on-screen message indicates that the password is not valid, and a new **Application Password** dialog box opens.

If you click **Cancel**, the application is not opened.

Accessing the application in EcoStruxure Control Expert after an auto-lock, when EcoStruxure Control Expert is not connected to the controller or when the project in EcoStruxure Control Expert is *equal* to the project in the controller:

When the auto-lock time is elapsed, an **Application Password** dialog box opens.

Enter the password and click **OK**.

Result: If the password is correct, EcoStruxure Control Expert becomes active again. If the password is incorrect, a message box indicates that the password is not valid, and a new **Application Password** dialog box opens.

Click **Close** to close the unsaved application.

Accessing the application in the controller after an auto-lock, when EcoStruxure Control Expert is connected to the controller and the application in EcoStruxure Control Expert is *different* from the application in the controller:

On connection, if the EcoStruxure Control Expert software application and the controller application are not equal, an **Application Password** dialog box opens.

Enter the password and click **OK**.

Result: If the password is correct, the connection is established. If the password is incorrect, a message box indicates that an incorrect password was entered, and a new **Application Password** dialog box opens.

If you click **Cancel**, the connection is not established.

NOTE: On connection, if the EcoStruxure Control Expert software application and the controller applications are equal, there is no password request. If no password has been initially entered (left empty on project creation), click **OK** to establish the connection on password prompt.

NOTE:

- After three attempts with an invalid password, wait an increasing amount of time between subsequent password attempts. The delay increases from 15 seconds to 1 hour, with the wait increment increasing by a factor of 2 after each attempt with an invalid password.
- For forgotten passwords, refer to the instructions for addressing lost passwords, page 197.

Enable the File Encryption Option

NOTE: Set an application password *before* you enable file encryption.

Enable the file encryption option:

Step	Action
1	Right-click **Project** in the **Project Browser**.
2	Select the **Properties** command from the pop-up menu to open the **Properties of Project** window.
3	Select the **Project & Controller Protection** tab.
4	Select the **File encryption active** checkbox to open the **Create Password** window.
5	Enter the password in the **Entry** field.

Step	Action
6	Enter the confirmation of the password in the **Confirmation** field.
7	Click **OK** to confirm.
8	Click **OK** or **Apply** in the **Properties of Project** window to confirm the changes.

Disable the File Encryption Option

Disable the file encryption option:

Step	Action
1	Right-click **Project** in the **Project Browser**.
2	Select the **Properties** command from the pop-up menu to open the **Properties of Project** window.
3	Select the **Project & Controller Protection** tab.
4	Clear *(uncheck)* the **File encryption active** checkbox to open the **File Encryption Password** window.
5	Enter the password and click **OK** to confirm that the application is not encrypted.
6	Click **OK** or **Apply** in the **Properties of Project** window to confirm all changes.

Change the File Encryption Password

Change the file encryption password:

Step	Action
1	Right-click **Project** in the **Project Browser**.
2	Select the **Properties** command from the pop-up menu to open the **Properties of Project** window.
3	Select the **Project & Controller Protection** tab.
4	In the **File encryption** field, click **Change password ...** to open the **Modify Password** window.
5	Enter the previous password in the **Old password** field.
6	Enter the new password in the **Entry** field.
7	Enter the confirmation of the new password in the **Confirmation** field.
8	Click **OK** to confirm.
9	Click **OK** or **Apply** in the **Properties of Project** window to confirm the changes.

Clear the File Encryption Password

Clear the file encryption password:

Step	Action
1	Right-click **Project** in the **Project Browser**.
2	Select the **Properties** command from the pop-up menu to open the **Properties of Project** window.
3	Select the **Project & Controller Protection** tab.
4	In the **File encryption** field, click **Clear password...** to open the **Password** window.
5	Enter the password in the **Password** field.
6	Click **OK** to confirm.
7	Click **OK** or **Apply** in the **Properties of Project** window to confirm the changes.

NOTE: For forgotten file-encryption passwords, refer to the instructions for addressing lost passwords, page 197.

Compatibility Rules

You cannot open encrypted `.STA` and `.ZEF` application files in EcoStruxure Control Expert 15.0 Classic or earlier versions.

You cannot import encrypted `.ZEF` application files to EcoStruxure Control Expert with Topology Manager.

The compatibility rules between the application version and the EcoStruxure Control Expert/ Unity Pro version apply to `.ZEF` files that are exported without encryption.

NOTE: When file encryption is enabled for your project, you cannot save archived application files (`.STA`) without encryption.

Safety-Related Area Password Protection

At a Glance

Safety-related controllers include a safety-related area password protection function, which is accessible from the **Properties** dialog box of the project. This function is used to help protect project elements located within the safety-related area of the functional safety-related project.

NOTE: When the safety-related area password protection function is active, the safety-related parts of the application cannot be modified

Modifications to the following safety-related parts are not permitted when safety-related area password protection is enabled:

Safety-related Part	**Forbidden action (offline AND online)**
Configuration	Modify controller characteristics
	Add, Delete, Modify a safety-related module in the rack
	Modify safety-related Power supply
Types	Create, Delete, Modify a safety-related DDT
	Change a DDT attribute: from NOT SAFE->defined SAFE-STATE
	Change a DDT attribute: from defined SAFE-STATE->NOT SAFE
	Create, Delete, Modify a safety-related DFB
	Change a DFB attribute: from NOT SAFE->defined SAFE-STATE
	Change a DFB attribute: from defined SAFE-STATE->NOT SAFE
Program-SAFE	Any Change under the **Variables an FB instances** node
	Create Task
	Import Task
	Modify Task
	Create Section
	Delete Section
	Import Section
	Modify Section
Project Settings	Modify SAFE project settings
	Modify COMMON project settings

Encryption

The safety-related area password uses the standard encryption SHA-256.

Safety-Related Area Password Function versus Functional Safety-Related Project User Rights

The activation of the safety-related area password and the implementation of user rights created in the **Security Editor** are mutually exclusive security functions, as follows:

- If the user launching EcoStruxure Control Expert has been assigned a user profile, that user can access the safety-related areas of the safety-related application if the user enters the safety-related area password and has been granted access rights in the **Security Editor**.
- If user profiles have not been assigned, a user can access the safety-related areas of the safety-related application by entering the safety-related area password.

Visual Indicators in EcoStruxure Control Expert

The state of the safety-related area protection function can be visibly detected by viewing the **Program-SAFE** node in the **Project Browser**:

- A locked padlock indicates a safety-related area password has been created and activated.
- An unlocked padlock indicates a safety-related area password has been created but not activated.

  NOTE: If the safety-related application is closed then re-opened, the safety-related area password is automatically activated on re-opening.
- No padlock indicates a safety-related area password has not been created.

Compatibility

As of EcoStruxure Control Expert version 14.0, the safety-related area password function exists for M580 safety-related controllers as of firmware version 2.80.

NOTE:

- Application program `.STU`, `.STA`, and `.ZEF` files (which are created as of EcoStruxure Control Expert version 14.0) cannot be opened in Unity Pro version 13.1 or earlier.
- Replacing an M580 safety-related controller in a EcoStruxure Control Expert version 14.0 application has the following effect:
  - Upgrading from firmware 2.70 to 2.80 (or any subsequent supporting version(s)) adds the safety-related area password functionality to the **Program & Safety Protection** tab of the **Project > Properties** window.
  - Downgrading from firmware 2.80 (and any subsequent comparable version(s)) to 2.70 removes the safety-related area password functionality.

Activating Protection and Creating Password

Procedure for activating the protection of sections and creating the password:

Step	Action
1	In the project browser right-click **Project**.
2	Select **Properties** command from the popup menu. **Result**: The **Properties of Project** window appears.
3	Select the **Program & Safety Protection** tab.
4	In the **Safety** area, activate the protection by checking the **Protection active** box. **Result**: The **Modify Password** dialog box appears.
5	Enter a password in the **Entry** field.
6	Enter the confirmation of the password in the **Confirmation** field.
7	Click **OK** to confirm.
8	Click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Changing the Password

Procedure for changing the project sections protection password:

Step	Action
1	In the project browser right-click **Project**.
2	Select **Properties** command from the popup menu. **Result**: The **Properties of Project** window appears.
3	Select the **Program & Safety Protection** tab.
4	In the **Safety** area, click **Change password ...**. **Result**: The **Modify Password** dialog box appears:
5	Enter previous password in the **Old password** field.
6	Enter the new password in the **Entry** field.
7	Enter the confirmation of the new password in the **Confirmation** field.

Step	Action
8	Click **OK** to confirm.
9	Click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Deleting the Password

Procedure for deleting the project sections protection password:

Step	Action
1	In the project browser right-click **Project**.
2	Select **Properties** command from the popup menu. **Result**: The **Properties of Project** window appears.
3	Select the **Program & Safety Protection** tab.
4	In the **Safety** area, click **Clear password...**. **Result**: The **Access control** dialog box appears:
5	Enter the previous password in the **Password** field.
6	Click **OK** to confirm.
7	Click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Firmware Protection

Overview

Firmware protection by a password helps prevent unwanted access to the module firmware.

Password

The password is case-sensitive and contains 8 to 16 alphanumeric characters. The password robustness is increased when it contains a mix of upper and lower case, alphabetical, numerical, and special characters.

NOTE: When importing a ZEF file, the firmware password is stored inside the module only if the **File encryption** option is selected.

Changing the Password

It is possible to change a password at any time.

NOTE: Firmware password default value in the Control Expert application is: **`fwdownload`**.

- For firmware V4.01 and later, you need to change the firmware password default value, otherwise it will not be possible to build the Control Expert application.
- For firmware versions earlier than V4.01 it is not mandatory but strongly advised to change firmware password default value.

Procedure for changing the firmware protection password:

Step	Action
1	In the project browser right-click **Project**.
2	Select **Properties** command from the popup menu. **Result**: The **Properties of Project** window appears.
3	Select **Project & Controller Protection** tab.
4	In the **Firmware** field, click **Change password ...**. **Result**: The **Modify Password** window appears.
5	Enter previous password in the **Old password** field.
6	Enter the new password in the **Entry** field.
7	Enter the confirmation of the new password in the **Confirmation** field.
8	Click **OK** to confirm.
9	Click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Resetting the Password

Resetting the password assigns its default value to the firmware password in the Control Expert application if the current password is confirmed.

To reset the password:

Step	Action
1	In the project browser right-click **Project**.
2	Select **Properties** command from the popup menu. **Result**: The **Properties of Project** window appears.
3	Select **Project & Controller Protection** tab.
4	In the **Firmware** field, click **Reset password...**. **Result**: The **Password** window appears.
5	Enter current password in the **Password** field.
6	Click **OK** to confirm.
7	Click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. The new password is the default password: **`fwdownload`**. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Program Unit, Section and Subroutine Protection

At a Glance

The protection function is accessible from the **Properties** screen of the project in offline mode.

This function is used to protect the program elements (sections, Program Units).

NOTE: The protection is not active as long as the protection has not been activated in the project.

NOTE: The project protection is effective to the marked program elements only. This does not prevent from:

- Connecting to the CPU
- Uploading application from the CPU
- Changing the configuration
- Adding new Program Units and/or sections
- Changing the logic in a new (not protected) section

Activating Protection and Creating Password

Procedure for activating the protection and creating the password for sections and Program Units:

Step	Action
1	In the project browser right-click **Project**.
2	Select **Properties** command from the popup menu. **Result**: The **Properties of Project** window appears.
3	Select **Program & Safety Protection** tab.
4	In the **Sections & Program Units** field, activate the protection by checking the **Protection active** box. **Result**: The **Modify Password** dialog box appears:
5	Enter a password in the **Entry** field.
6	Enter the confirmation of the password in the **Confirmation** field.
7	Select **Crypted** check box if an enhanced password protection is required. **NOTE:** A project with a crypted password cannot be edited with Unity Pro V4.0 and earlier.
8	Click **OK** to confirm.
9	Click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Notes

If a program element is configured with a protection (read or read/write), when protection has been activated this will be indicated by a locked padlock at the program element level.

If the program element is configured with a protection but the protection is disabled, an open padlock is displayed at the program element level.

Changing the Password

Procedure for changing the project protection password for sections and Program Units:

Step	Action
1	In the project browser right-click **Project**.
2	Select **Properties** command from the popup menu. **Result**: The **Properties of Project** window appears.
3	Select **Program & Safety Protection** tab.
4	In the **Sections & Program Units** field, click **Change password ...**.

Step	Action
	Result: The **Modify Password** dialog box appears:
5	Enter previous password in the **Old password** field.
6	Enter the new password in the **Entry** field.
7	Enter the confirmation of the new password in the **Confirmation** field.
8	Select **Crypted** check box if an enhanced password protection is required. **NOTE:** A project with a crypted password cannot be edited with Unity Pro V4.0 and earlier. Unity Pro is the former name of Control Expert for version 13.1 or earlier.
9	Click **OK** to confirm.
10	Click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Deleting the Password

Procedure for deleting the project protection password for sections and Program Units:

Step	Action
1	In the project browser right-click **Project**.
2	Select **Properties** command from the popup menu. **Result**: The **Properties of Project** window appears.
3	Select **Program & Safety Protection** tab.
4	In the **Sections & Program Units** field, click **Clear password...**. **Result**: The **Access control** dialog box appears:
5	Enter the previous password in the **Password** field.
6	Click **OK** to confirm.
7	Click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Data Storage/Web Protection

Overview

Protection by a password helps prevent unwanted access to the data storage zone of the SD memory card (if a valid card is inserted in the CPU).

For Modicon M580 CPUs in a project created by Control Expert with version:

- Earlier than version 15.1, you can provide password protection for data storage access.
- As of version 15.1, you can provide password protection for both web diagnostics and data storage access.

  NOTE: When a controller is managed as part of a system project, the **Web Diagnostic/ Data Storage** password is disabled in Control Expert editor and needs to be managed by using the Topology Manager.

Password

The password is case-sensitive and contains 8 to 16 alphanumeric characters. The password robustness is increased when it contains a mix of upper and lower case, alphabetical, numerical, and special characters.

NOTE: When importing a ZEF file, the Data Storage/Web password is stored inside the module only if the **File encryption** option is selected.

Changing the Password

It is possible to change a password at any time.

NOTE: Data Storage/Web password has a default value in the Control Expert application. This default value depends on the version of Control Expert:

- **`datadownload:`** Control Expert versions earlier than 15.1
- **`webuser:`** Control Expert versions as of 15.1

Changing the default password is mandatory, or not, depending on the module firmware version:

- As of firmware version 4.01, you need to change the Data Storage/Web password default value, otherwise it will not be possible to build the Control Expert application.
- For firmware versions earlier than 4.01 it is not mandatory but strongly advised to change Data Storage/Web password default value.

Procedure for changing the Data Storage/Web password:

Step	Action
1	In the project browser right-click **Project**.
2	Select **Properties** command from the popup menu. **Result**: The **Properties of Project** window appears.
3	Select **Project & Controller Protection** tab.
4	In the **Data Storage** (or **Web Diagnostics/Data Storage**) field, click **Change password ...**. **Result**: The **Modify Password** window appears.
5	Enter previous password in the **Old password** field.
6	Enter the new password in the **Entry** field.
7	Enter the confirmation of the new password in the **Confirmation** field.
8	Click **OK** to confirm.
9	Click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Resetting the Password

Resetting the password assigns its default value to the Data Storage/Web password in the Control Expert application if the current password is confirmed.

To reset the password:

Step	Action
1	In the project browser right-click **Project**.
2	Select **Properties** command from the popup menu. **Result**: The **Properties of Project** window appears.
3	Select **Project & Controller Protection** tab.
4	In the **Data Storage** (or **Web Diagnostic/Data Storage**) field, click **Reset password...**. **Result**: The **Password** window appears.
5	Enter current password in the **Password** field.
6	Click **OK** to confirm.
7	Click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. The new password is the default password: **`datadownload`**. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Loss of Password

Overview

If you forget your password, proceed as indicated in the following procedures and contact Schneider Electric support.

NOTE: The application password recovery procedure differ depending on whether the file encryption option is enabled or disabled.

Control Expert Application Password without File Encryption Option

The following procedure for resetting the application password is valid when file encryption option is disabled or for application file managed with Control Expert 15.0 Classic or earlier versions.

Schneider Electric support needs a string of alphanumeric characters displayed in the **Password forgotten** pop-up window as soon as you press `SHIFT+F2` in the **Password** dialog box.

The following conditions must be met in order to reach the **Password** dialog box:

- At open time, select the application and the **Password** dialog box is displayed.
- At auto-lock time, the **Password** dialog box is displayed. If you do not remember the password, select **Close**. Open the application again and the **Password** dialog box is displayed.

  NOTE: When the application is closed without entering a password after an auto-lock, all modifications are lost.

Procedure for resetting the application password:

Step	Action
1	**Condition:** The **Password** dialog box is displayed.
2	Press `SHIFT+F2`. **Result:** The **Password forgotten** pop-up window is open and a string of alphanumeric characters is displayed.
3	Copy this string and give it to Schneider Electric support.
4	Receive the generated password from Schneider Electric support. **NOTE:** The password is a temporary password, available as long as you do not modify the application.
5	Enter this password.

Step	Action
6	Modify the password (old password = password provided by Schneider Electric support).
7	Click **Build** > **Build Changes**.
8	**Save** the application.

Control Expert Application Password with File Encryption Option

If you forget your application password when file encryption is enabled, you need to send the application file to Schneider Electric support. Then you receive back the encrypted application file with a new file application password from Schneider Electric support.

NOTE: We highly recommend to change the application password.

CPU Application Password

Procedure for resetting the CPU application password if the respective **.STU* file is available:

Step	Action
1	Open the respective **.STU* file.
2	When the **Password** dialog box is displayed press `SHIFT+F2`. **Result:** The **Password forgotten** pop-up window is open and a string of alphanumeric characters is displayed.
3	Copy this string and give it to Schneider Electric support.
4	Receive the generated password from Schneider Electric support. **Note:** The password is a temporary password, available as long as you do not modify the application.
5	Enter this password.
6	Modify the password (old password = password provided by Schneider Electric support).
7	**Connect** to the PLC.
8	Click **Build** > **Build Changes**.
9	**Save** the application.

Procedure for resetting the CPU application password if the respective **.STU* file is not available:

Step	Action
1	**Condition:** At connection time, the **Password** dialog box is displayed.
2	Press SHIFT+F2. **Result:** The **Password forgotten** pop-up window is open and a string of alphanumeric characters is displayed.
3	Copy this string and give it to Schneider Electric support.
4	Receive the generated password from Schneider Electric support. **Note:** The password provided by Schneider Electric support is a temporary password, available as long as you do not modify the application.
5	Enter this password.
6	Upload the application from CPU.
7	**Save** the application.
8	Modify the password (old password = the one provided by Schneider Electric support).
9	Click **Build > Build Changes**.
10	**Save** the application.

File Encryption Password

Schneider Electric support needs a string of alphanumeric characters displayed in the **Password forgotten** pop-up window as soon as you press SHIFT+F2 in the **Password** dialog box.

To reach the **Password** dialog box:

- Go to **Project > Properties of Project > Project & Controller Protection**
- In the **File encryption** field, click **Clear password...**. The **Password** dialog box is displayed.

Procedure for resetting the file encryption password:

Step	Action
1	**Condition:** The **Password** dialog box is displayed.
2	Press SHIFT+F2. **Result:** The **Password forgotten** pop-up window is open and a string of alphanumeric characters is displayed.
3	Copy this string and give it to Schneider Electric support.
4	Receive the generated password from Schneider Electric support.

Step	Action
	Note: The password is a temporary password, available as long as you do not modify the application.
5	Enter this password and click **OK** to close the **Password** dialog.
6	Click **Change Password** and change the password (the old password = password provided by Schneider Electric support).
7	Click **OK** to close the **Modify Password** dialog, then click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Safe Area Password

Schneider Electric support needs a string of alphanumeric characters displayed in the **Password forgotten** pop-up window as soon as you press `SHIFT+F2` in the **Password** dialog box.

To reach the **Password** dialog box:

- Go to **Project > Properties of Project > Program & Safety Protection**
- In the **Safety** field, click **Change password...**. The **Password** dialog box is displayed.

Procedure for resetting the safe area password:

Step	Action
1	**Condition:** The **Password** dialog box is displayed.
2	Press `SHIFT+F2`. **Result:** The **Password forgotten** pop-up window is open and a string of alphanumeric characters is displayed.
3	Copy this string and give it to Schneider Electric support.
4	Receive the generated password from Schneider Electric support. **Note:** The password is a temporary password, available as long as you do not modify the application.
5	Enter this password and click **OK** to close the **Password** dialog.
6	Click **Change Password** and change the password (the old password = password provided by Schneider Electric support).
7	Click **OK** to close the **Modify Password** dialog, then click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Firmware Password

Schneider Electric support needs a string of alphanumeric characters displayed in the **Password forgotten** pop-up window as soon as you press `SHIFT+F2` in the **Password** dialog box.

To reach the **Password** dialog box:

- Go to **Project > Properties of Project > Project & Controller Protection**
- In the **Firmware** field, click **Reset password...**. The **Password** dialog box is displayed.

Procedure for resetting the firmware password:

Step	Action
1	**Condition:** The **Password** dialog box is displayed.
2	Press `SHIFT+F2`. **Result:** The **Password forgotten** pop-up window is open and a string of alphanumeric characters is displayed.
3	Copy this string and give it to Schneider Electric support.
4	Receive the generated password from Schneider Electric support. **Note:** The password is a temporary password, available as long as you do not modify the application.
5	Enter this password and click **OK** to close the **Password** dialog.
6	Click **Change Password** and change the password (the old password = password provided by Schneider Electric support).
7	Click **OK** to close the **Modify Password** dialog, then click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Data Storage/Web Password

Schneider Electric support needs a string of alphanumeric characters displayed in the **Password forgotten** pop-up window as soon as you press `SHIFT+F2` in the **Password** dialog box.

To reach the **Password** dialog box:

- Go to **Project > Properties of Project > Project & Controller Protection**
- In the **Data Storage** field, click **Reset password...**. The **Password** dialog box is displayed.

Procedure for resetting the data storage password:

Step	Action
1	**Condition:** The **Password** dialog box is displayed.
2	Press `SHIFT+F2`. **Result:** The **Password forgotten** pop-up window is open and a string of alphanumeric characters is displayed.
3	Copy this string and give it to Schneider Electric support.
4	Receive the generated password from Schneider Electric support. **Note:** The password is a temporary password, available as long as you do not modify the application.
5	Enter this password and click **OK** to close the **Password** dialog.
6	Click **Change Password** and change the password (the old password = password provided by Schneider Electric support).
7	Click **OK** to close the **Modify Password** dialog, then click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Introduction to the various directories of the project browser

About this sub-chapter

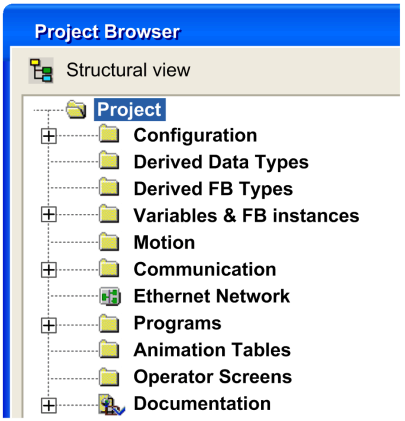
This sub-chapter describes the various directories of the project browser.

Project Directory

At a Glance

The **Project** directory of the structural view allows you to access the structure of the project and the associated services.

The following illustration shows the **Project** directory:



Associated services

The **Project** directory allows you to access the following services, which can be reached via the contextual menu:

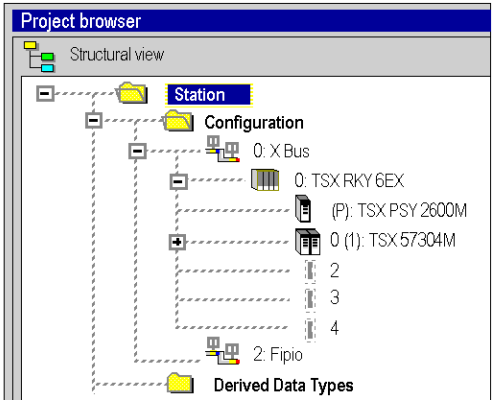
Directory	Services
Project	**Export Project**: allows you to access export of the global project, page 1720. **Project Settings**: allows you to access the project specific settings, page 551. **Properties**: allows you to access the properties of the global project.
Configuration	allows you to access the hardware configuration, page 204 and the parameter settings of the modules.
Derived Data Types	allows you to access the DDTs, page 206.
Derived FB Types	allows you to access the DFB, page 207 types.
Variables & FB instances	allows you to access the variables, page 208 and function block instances.
Motion	allows you to access the declaration, page 210 and configuration of the servodrives.
Communication	allows you to access the configuration of the networks, page 212.
Ethernet Network	allows you to access the **Ethernet Network** configuration, page 213.
Programs	allows you to access the project programs, page 213.
Animation Tables	allows you to access the animation tables, page 217.
Operator Screens	allows you to access the operator screens, page 220.
Documentation	allows you to access the documentation, page 222.

Configuration Directory

At a Glance

The **Configuration** directory of the structural view of the project allows you to access the hardware configuration and the parameter settings of the following modules: bus, rack, module.

The following illustration shows an example of a directory tree of the **Configuration** directory:



Accessible Services

The **Configuration** directory allows you to access the following services, which can be reached via the contextual menu:

Directory	Services
Configuration	**Open**: used to access the bus editor, X Bus in the above example. **Import**: used to import the configuration, page 1698 of the project inputs/outputs (offline mode only). **Import SIS**: used to import the configuration, page 1386 of a project created using the SIS Automation tool. **Export**: used to export the configuration, page 1696 of the project inputs/outputs (offline mode only).
Bus	**Open**: used to access the bus editor, X Bus in the above example. **Go to Bus Master**: displays the processor highlighted in the project navigator, TSX 57304M in the above example.
Rack	**Open**: used to access the bus editor, TSX RKY 6EX in the above example.
Module	**Open**: used to access the input/output editor (module settings). **Export**: used to export the configuration, page 1696 of the module(offline mode only and depending on the module).
Empty slot	**Import**: used to import the configuration of a BMENOC03•1 module with all devices configured behind the NOC master DTM. import the configuration, page 1698 of a module (offline mode only).

Access

From the **Configuration** directory, you can:

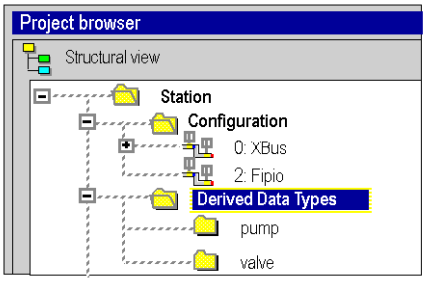
- Configure the PLC rack, page 1301 with:
  - a power supply, page 1308,
  - a processor, page 1310,
  - one or more modules, page 1314.
- Configure field bus devices, page 1319,
- Access the configuration of the rack elements:
  - Premium, page 1361 and Quantum, page 1371 processors,
  - modules, page 1380.

Derived data types (DDT) directory

At a Glance

The **Derived data types** (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) directory of the structural view of the project allows you to access the DDT types.

The following illustration shows an example of a directory tree of the **Derived data types** directory:



Associated services

The **Derived Data Types** directory allows you to access the following services, which can be reached via the contextual menu:

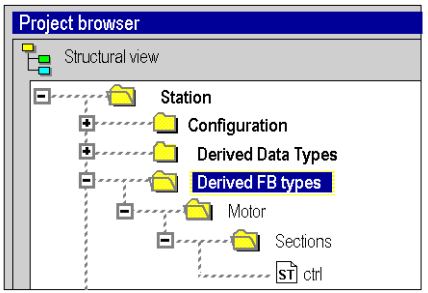
Directory	Services
Derived Data Types	**Open**: allows you to access the **DDT types** tab of the data editor, page 349, from which you can: • create, page 357 a DDT, • manage, page 368 a DDT, **Get from Library**: allows you read access to one or more DDT types from a library, page 314. **Put in Library**: used to archive, page 375 all the DDTs in a library, page 314. **Export**: allows you to access export of all DDT types, page 1705. **Import**: allows you to access import of one or more DDT types, page 1706.
DDT (Input, Output, .etc.)	**Open**: allows you to access the DDT in the **DDT types** tab of the data editor. **Delete**: allows you to delete the DDT. **Put in Library**: allows you write access to the DDT in a library. **Analyze**: allows you to analyze, page 438 the DDT. **Properties**: allows you to access the DDT properties. **Export**: allows you to access export of all DDT types, page 1705.

Derived FB (DFB) types directory

At a Glance

The **Derived FB types** (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) directory of the structural view of the project allows you to access the DFB types.

The following illustration shows an example of a directory tree of the **Derived FB types** directory:



Associated services

The **Derived FB types** directory allows you to access the following services, which can be reached via the contextual menu:

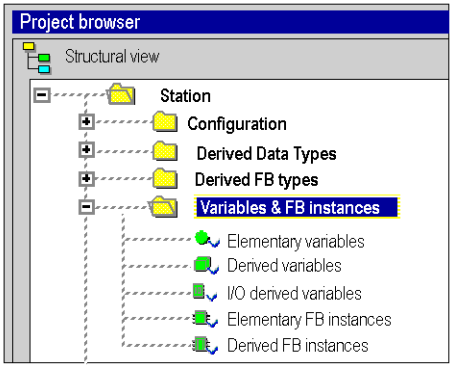
Directory	Services
FB Type	**Open**: allows you to access the **DFB** tab of the data editor, page 1256. **Get from Library**: allows you read access to one or more DFB types from a library, page 314. **Put in Library**: allows you write access to all DFB types in a library. **Export**: allows you to access export of all DFB types, page 1703 of the project. **Import**: allows you to access import of one or more DFB types, page 1704.
DFB Type (Counter, etc.)	**Open**: allows you to access the DFB type in the **DFB** tab of the Data Editor, page 1256. **Delete**: allows you to delete the DFB type. **Put in Library**: allows you write access to the DFB type in a Library, page 314. **Analyze**: allows you to analyze, page 438 the DFB type. **Properties**: allows you to access the properties, page 1268 of the DFB type. **Export**: allows you to access export of the DFB type, page 1703.
Sections	**New Section**: allows you to create a new section, page 502 in the DFB type.
Section XX	**Open**: allows you to access the section, page 502 program language editor. **Delete**: allows you to delete the section. **Properties**: allows you to access the properties, page 505 of the section.

Variables Directory

At a Glance

The **Variables & FB instances** directory of the structural view of the project is used to access the variables (EDT, (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual)DDT, IODDT) (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual)and the function block instances(EFB, DFB). (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual)

The following illustration shows an example of the **Variables & FB instances** directory:



Accessible Services

The **Variables & FB instances** directory allows you access to the following services, which can be reached via the contextual menu:

Directory	Services
Variables & FB instances	**Open**: used to access the variables editor, **Export**: used to access the export of all variables, page 1707 of the project, **Import**: used to access the import of all variables, page 1710 of the project.
Elementary variables **Derived variables** **Decice DDT variables** **I/O derived variables** **Elementary FB instances** **Derived FB instances**	**Open**: used to access the tab corresponding to the variables editor, **Export**: used to access the export of all variables, page 1707 of the family selected (EDT, DFB, etc.) .

Access

From the **Variables & FB instances** directory, you can access the different tabs of the data editor:

- **Variables** tab,
  - Create, page 390 a data instance,
  - Create, page 400 an IODDT type data instance,
  - Create, page 403 a Device DDT type data instance,
  - Modify, page 418 attributes of data instances.
- **DDT Types** tab,
  - Create, page 357 a data instance,
  - Archive, page 375 DDTs in a library, page 314,
- **Function blocks** tab,
  - Create, page 380 a data instance,
  - Modify, page 386 attributes of data instances.
- **DFB Types** tab,
  - Create, page 1256 a DFB type,
  - Configure settings, page 1257 of a DFB type.

Motion Directory

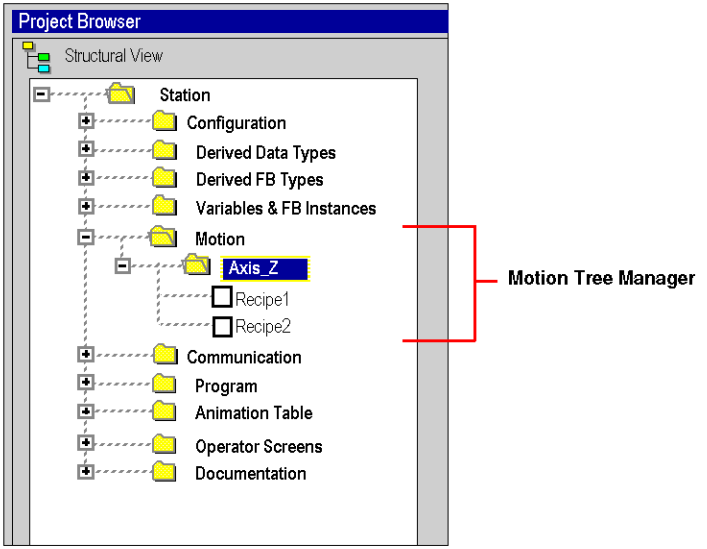
At a Glance

The **Motion** directory of the structural view of the project allows you to access the declaration and configuration of the servodrives.

When declaring a servodrive, various information is required, such as:

- the name given to the servodrive
- the type of servodrive
- the CANopen address of the servodrive
- the reference of the servodrive
- the version of the servodrive
- the input of variable names associated to the axis.

The following diagram shows an example of a tree structure for the **Motion** directory:



In this diagram, the name given to the servodrive is ‘Axis_Z’.

A recipe is linked, by default, each time an axis is created. It is possible to create several recipes (see Premium using EcoStruxure™ Control Expert, Motion Function Blocks, Start-up Guide).

Accessible Services

The **Motion** directory gives you access to the following services, which can be reached via the contextual menu:

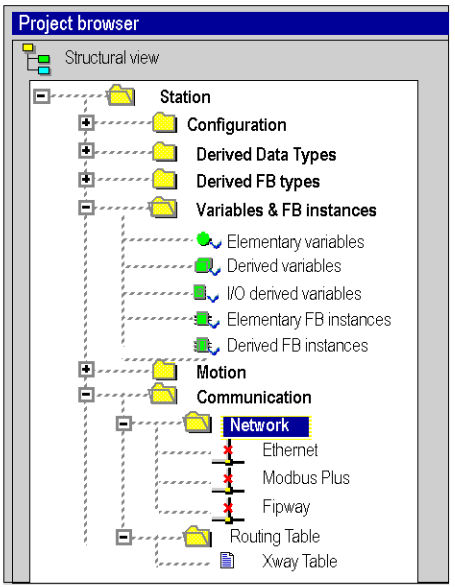
Directory	Service
Motion	**New axis**: allows you to create a new axis.
Axis	**New recipe**: allows you to create a new recipe. **Delete**: allows you to delete an axis. **Properties**: allows you to access the axis properties.
Recipe	**Delete**: allows you to delete a recipe. **Properties**: allows you to access the recipe properties.

Communication Directory

At a Glance

The **Communication** directory of the structural view of the project allows you to access the configuration of the networks.

The following illustration shows an example of a directory tree of the **Communication** directory:



Associated services

The **Communication** directory allows you to access the following services, which can be reached via the contextual menu:

Directory	Services
Communication	**Export**: allows you to access export of all networks, page 1711. **Import**: allows you to access import of one or more networks, page 1712.
Network	**New Network**: allows you to add a network, page 448 to your project.

Directory	Services
Network (Network_1, Modbus Plus_A, etc.)	**Open**: allows you to access the corresponding network editor: Ethernet, Modbus Plus, Fipway. **Export**: allows you to access export of one network, page 1711. **Delete**: allows you to delete a network. **Properties**: allows you to access the network properties.
Routing Table	**Open**: allows you to access the configuration, page 456 of a bridge between 2 networks (routing tables).

Ethernet Network Directory

At a Glance

Double-click the **Ethernet Network** directory of the project structural view provides access to the **Ethernet Network Manager** tool.

This directory is available in a Modicon M580 configuration.

Ethernet Network Window

The **Ethernet Network** window displays configuration information on the PAC networks and allows you to edit some parameters for specific communication adapter modules.

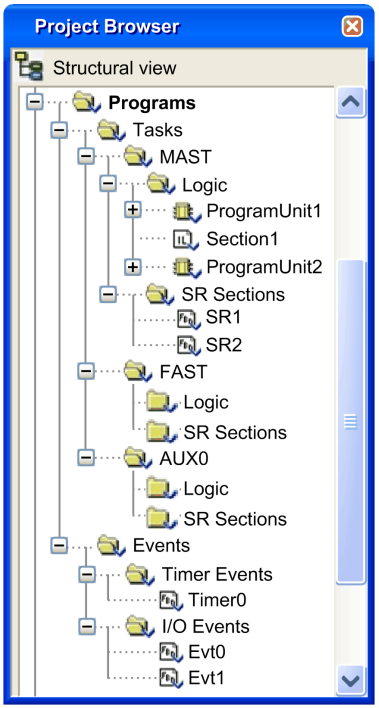
More information on the **Ethernet Network Manager** tool is provided in Modicon M580 topology guides (see Modicon M580, System Planning Guide for, Complex Topologies).

Programs Directory

At a Glance

The **Programs** directory of the structural view of the project allows you to define the structure of the program and to access the language editors of the program elements: Program Units, sections, program modules and event processing.

The following illustration shows an example of a directory tree of the **Programs** directory:



NOTE: When Program Units in the application are not allowed, the **Logic** folder is replaced by the **Sections** folder in the directory tree. For more detailed information refer to the chapter *Description of the Available Functions for Each Type of PLC* (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).

Services associated with the sequential Tasks directories.

The sequential **Tasks** directories (MAST, FAST, etc.) allow you to access the following services, which can be reached via the contextual menu:

Directory	Services
Tasks	**New Task ...**: allows you to create a new sequential task, page 472 (FAST, AUX, AUX0, AUX1). The MAST task is created by default. **Import ...**: allows you to import a task, page 1701.
MAST, **FAST**, etc.	**Delete**: allows you to delete the task. The MAST task cannot be deleted. **Clear**: allows you to clear the contents of the task. This clears all the sections of the task.

Directory	Services
	Export: allows you to export the sequential task, page 1699. **Properties**: allows you to access the properties of the sequential task.
Logic	**New Program Unit...**: allows you to create a new “empty” Program Unit. **New Section ...**: allows you to create a new “empty” section, page 502. **Import ...**: allows you to import a program unit or a section, page 1701. **Create builded Activation Conditions Table**: allows you to initialize an animation table, with the activation condition variables associated with the sections.
Or	
Sections	**New Section ...**: allows you to create a new "empty" section, page 502. **Import ...**: allows you to import a section, page 1701. **Create section activation conditions table**: allows you to initialize an animation table, with the activation condition variables associated with the sections.

Services associated with the Program Unit

The structure of a Program Unit allow you to access the following services, which can be reached via the contextual menu:

Program Element	Services
For each Program Unit	**Analyze**: allows you to analyze the Program Unit. **Delete**: allows you to delete the Program Unit. **Copy**: allows you to copy the Program Unit. **Paste After**: allows you to paste after a duplication of the copied Program Unit, page 500. **Paste Special after...**: allows you to paste after a duplication of the copied Program Unit with effective parameters, page 501. **Import after**: allows you to import a Program Unit or a section, page 1701. **Export ...**: allows you to export the Program Unit, page 1699. **Properties**: allows you to access the properties of the Program Unit.
For the **Interface & Variables**	**Open**: allows you to access the Program Unit data editor.
Sections directory (under Program Unit)	**New Section ...**: allows you to create a new “empty” section, page 502. **Import ...**: allows you to import a section, page 1701

Program Element	Services
For each section, section **Macro-step**, action, or transition	Refer to services associated, page 216 with the section.
Animation Tables directory (under Program Unit)	Refer to services associated, page 218 with the Animation Tables Directory.
For each animation table	

Services associated with the section

The **section** allow you to access the following services, which can be reached via the contextual menu:

Program Element	Services
For each section	**Open**: allows you to access the section, page 502 language editor. **Delete**: allows you to delete the section. **Detach**: allows you to detach the section of the functional module. **Export**: allows you to export the section, page 1699. **Import after ...**: allows you to import a section, page 1701 **Forcing to 0**, allows you to choose Force to 0 as the activation condition of the section (in online mode). **Forcing to 1**, allows you to choose Force to 1 as the activation condition of the section (in online mode). **Unforce**: allows you to choose cancel forcing as the activation condition of the section (in online mode). **Properties**: allows you to access the properties of the section.
For each Section **Macro-step**, action, or transition	**Open**: allows you to access the section language editor. **Delete**: allows you to delete the section. **Properties**: allows you to access the properties of the section.

NOTE: Only the MAST task can contain one or more sections in SFC language.

Services associated with the program modules directories

The program modules directories **(SR Sections)** allow you to access the following services, which can be reached via the contextual menu:

Directory	Services
SR Sections	**New SR Section ...**: allows you to create a new “empty” program module, page 522. **Import ...**: allows you to access import of program module, page 1701.
For each program module	**Open**: allows you to access the program module language editor. **Delete**: allows you to delete the program module. **Export**: allows you to access export program module, page 1699. **Properties**: allows you to access the properties of the program module.

Services associated with the event processing directories

The event processing (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) directories allow you to access the following services, which can be reached via the contextual menu:

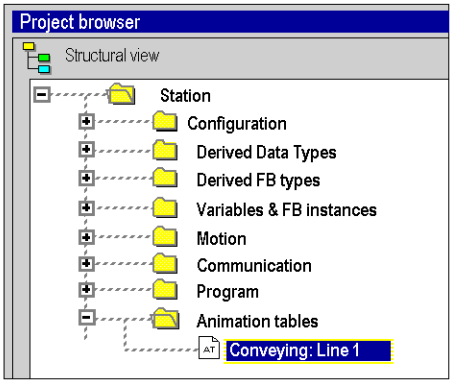
Directory	Services
Timer Events, **I/O Events**	**New Event Section**: allows you to create a new “empty” event processing, page 526. **Import**: allows you to access import of event processing, page 1701.
For each event processing	**Open**: allows you to access the event processing language editor. **Delete**: allows you to delete the event processing. **Detach**: allows you to detach the event processing of the functional module. **Export**: allows you to access export of event processing, page 1699. **Properties**: allows you to access the event processing properties.

Animation Tables Directory

At a Glance

The **Animation Tables Directory** of the structural view of the project allows you to access the animation tables.

The following illustration shows an example of a directory tree of the **Animation Tables** directory:



Associated services

The **Animation Tables** directory allows you to access the following services, which can be reached via the contextual menu:

Directory	Services
Animation tables	**New Animation Table**: allows you to create a new animation table, page 1475. **Paste**: used to paste an animation table into the clipboard or the animation tables directory. **Delete all**: allows you to delete all the animation tables. **Detach all**: allows you to detach all the animation tables from the functional modules. **Export**: allows you to access export of all animation tables, page 1716. **Import**: allows you to access import of one or more animation tables, page 1717. **Make All Tables Permanent**: allows you to change all temporary animation tables to permanent animation tables (see also Permanent and Temporary Animation Tables, page 1473). **Purge Temporary Tables**: allows you to delete all temporary animation tables (see also Permanent and Temporary Animation Tables, page 1473). **Open Forced Bits Table**: allows you to create the animation table that includes all forced bits of a project (see also Animation Table of Forced Bits, page 1500).
For each animation table	**Open**: allows you to access the animation table. **Copy**: allows you to duplicate an animation table. **Paste**: allows you to add a copy of an animation table to the clipboard or the animation tables directory. **Delete**: allows you to delete the animation table. **Detach**: allows you to Detach the animation table from the functional module. **Export**: allows you to access export of the animation table, page 1716. **Properties**: allows you to access the animation table properties.

Access

The **Animation Tables** Directory is used to create an animation table. From this table, you can:

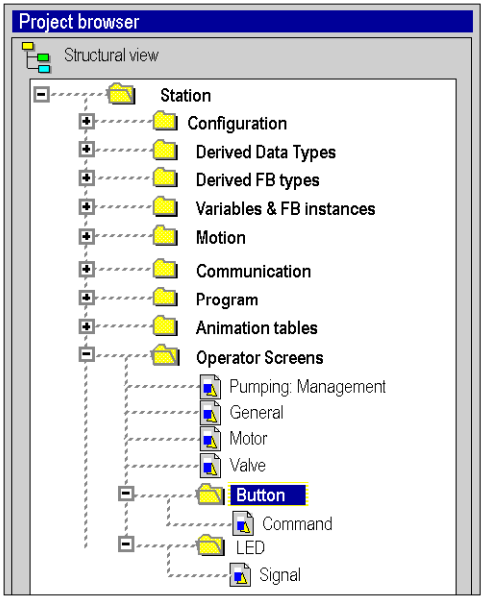
- add, page 1481 data,
- switch to modification, page 1487 mode,
- switch to forcing, page 1492 mode,
- modify or force, page 1497 several variables.

Operator Screens Directory

At a Glance

The **Operator Screens Directory** of the structural view of the project are used to access the operator screens.

The following diagram shows an example of a directory tree of the **Operator Screens** directory:



Associated services

The **Operator Screens** directory allows you to access the following services, which can be reached via the contextual menu:

Directory	Services
Operator Screens	**New screen**: allows you to create a new operator screen. **New family**: allows you to create a new family of screens. **Paste**: allows you to paste an operator screen into the clipboard or the operator screen directory. **Delete all**: allows you to delete all the operator screens. **Detach all**: allows you to detach all the operator screens from the functional modules. **Clean unuse image**: allows you delete all the created images that are not used. **Import**: allows you to access import of one or several screens, page 1715 and/or family of screens. **Export**: allows you to access export of all screens, page 1713. **Messages list**: allows you to access the message list for the operator screens.
For each family of screens	**New screen**: allows you to create a new operator screen. **Copy**: allows you to duplicate the family of screens and its associated operator screens. **Paste**: allows you to add a copy of a screen family and its operator screens to the clipboard or the operator screens directory. **Delete**: allows you to delete the family of screens. **Detach**: allows you to detach the functional module family. **Export**: allows you to access export of the family of screens, page 1713. **Family Properties**: allows you to access the properties of the family of screens.
For each operator screen	**Open**: allows you to access the operator screens editor. **Copy**: allows you to duplicate the operator screen. **Paste**: allows you to add a copy of an operator screen to the clipboard or the operator screen directory. **Delete**: allows you to delete the operator screen. **Detach**: allows you to detach the operator screen from the functional module. **Export**: allows you to access export of the operator screen, page 1713. **Screen properties**: allows you to access the properties of the operator screen.

Access

The **Operator Screens** directory is used to create screens. From these screens, you can:

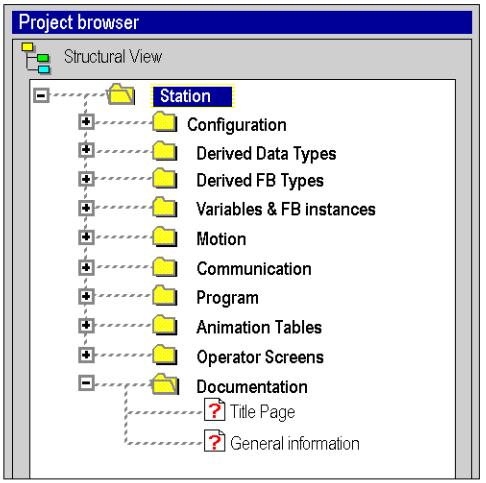
- create, page 1580 objects,
- insert objects from a library, page 1646,
- modify the attributes, page 1591 of the objects,
- manipulate, page 1616 the objects that make up the screen,
- use the screens in online mode, page 1635.

Documentation Directory

At a Glance

The **Documentation** directory of the structural view of the project allows you to access the documentation.

The following illustration shows an example of a **Documentation** directory:



Associated services

The **Documentation** directory allows you to access the following services, which can be reached via the contextual menu:

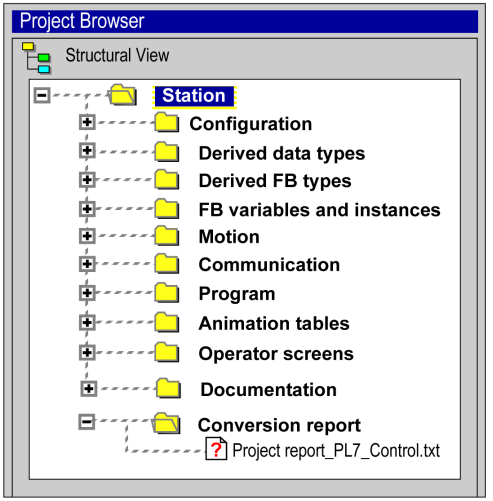
Directory	Services
Documentation	**Open**: allows you to access the documentation, page 1660 tool.
Title Page	**Print Configuration**: also gives access to the print configuration settings.
General information	**Open**: this heading allows you to specify various information about the project. A dialog box allows you to enter the information and to create hyperlinks, page 1814.

Conversion Report Directory

At a Glance

The **Conversion Report** directory is displayed in the structural view of the project when opening a PL7 (see EcoStruxure™ Control Expert, PL7 Application Converter, User Manual) project with a **.FEF** extension or a Concept (see EcoStruxure™ Control Expert, Concept Application Converter, User Manual) project with an **.ASC** extension. This directory enables you to access a conversion report for a project.

The following illustration shows an example of a directory tree of the **Conversion Report** directory:



Services

The **Conversion Report** directory provides you with:

- general information (application name, source file, PLC type, etc.),

- warnings,
- errors.

Summaries of the utilities associated with the structure view and function view

About this sub-chapter

This sub-chapter provides summaries of the utilities associated with the function view and structure view.

Summary of the services associated with the structural view

At a Glance

The following tables summarize the services associated with the structural view of the project and indicate the operating modes in which you can use them.

Project Directory

The **Project** directory, page 203 allows you to access the following services.

Directory	Services	Mode		
		Off-line	On-line Stop	On-line Run
Project	**Export Project**	Yes	No	No
	Properties	Yes	Yes	Yes

Configuration Directory

The **Configuration** directory, page 204 allows you to access the following services.

Directory	Services	Mode		
		Off-line	On-line Stop	On-line Run
Configuration	**Export**	Yes	No	No
	Import	Yes	No	No
	Import SIS	Yes	No	No
Bus	**Open**	Yes	Yes	Yes
	Go to Bus Master	Yes	Yes	Yes
Rack	**Open**	Yes	Yes	Yes
Module	**Open**	Yes	Yes	Yes

Derived FB Types Directory

The **Derived FB Types** directory, page 207 allows you to access the following services.

Directory	Services	Mode		
		Off-line	On-line Stop	On-line Run
FB Type	**Open**	Yes	Yes	Yes
	Export	Yes	No	No
	Import	Yes	No	No
	Put in Library	Yes	No	No
	Get from Library	Yes	No	No
For each type of DFB	**Open**	Yes	Yes	Yes
	Export	Yes	No	No
	Put in Library	Yes	No	No
	Delete	Yes(1)	Yes(1)	Yes(1)
	Analyze	Yes	Yes	Yes
	Properties	Yes	Yes	Yes

Directory	Services	Mode		
		Off-line	On-line Stop	On-line Run
Section	**Open**	Yes	Yes	Yes
	Delete	Yes	No	No
	Properties	Yes	Yes	Yes
(1) Only if the DFB type is not instantiated in the project.				

Derived Data Types Directory

The **Derived Data Types** directory, page 206 allows you to access the following services.

Directory	Services	Mode		
		Off-line	On-line Stop	On-line Run
Derived Data Types	**Open**	Yes	Yes	Yes
	Export	Yes	No	No
	Import	Yes	No	No
	Put in Library	Yes	No	No
	Get from Library	Yes	No	No
For each DDT	**Open**	Yes	Yes	Yes
	Export	Yes	Yes	No
	Put in Library	Yes	No	No
	Delete	Yes(1)	Yes(1)	Yes(1)
	Analyze	Yes	Yes	Yes
	Properties	Yes	Yes	Yes
(1) Only if the DDT type is not instantiated in the project.				

Variables & FB instances Directory

The **Variables & FB instances** directory, page 208 allows you to access the following services.

Directory	Services	Mode		
		Off-line	On-line Stop	On-line Run
Variables	**Open**	Yes	Yes	Yes
	Export	Yes	No	No
	Import	Yes	No	No
EDT, DDT, EFB, DFB	**Open**	Yes	Yes	Yes
	Export	Yes	No	No

Motion Directory

The **Motion** directory, page 208 allows you to access the following services.

Directory	Services	Mode		
		Off-line	On-line Stop	On-line Run
Motion	**New axis**	Yes	No	No
Axis	**New recipe**	Yes	No	No
	Delete	Yes	No	No
	Properties	Yes	Yes	Yes
Recipe	**Delete**	Yes	No	No
	Properties	Yes	Yes	Yes

Communication Directory

The **Communication** directory, page 212 allows you to access the following services.

Directory	Services	Mode		
		Off-line	On-line Stop	On-line Run
Communication	**Export**	Yes	No	No
	Import	Yes	No	No
Network	**New Network**	Yes	No	No

Directory	Services	Mode		
		Off-line	On-line Stop	On-line Run
For each network	**Open**	Yes	Yes	Yes
	Export	Yes	Yes	Yes
	Delete	Yes	Yes	Yes
	Properties	Yes	Yes	Yes
Routing Table	**Open**	Yes	Yes	Yes

Programs Directory

The **Programs** directory, page 213 allows you to access the following services.

Directory	Services	Mode		
		Off-line	On-line Stop	On-line Run
Tasks	**New Task**	Yes	No	No
	Import	Yes	No	No
For each task, (MAST, FAST, AUX•)	**Clear**	Yes	Yes	No
	Delete	Yes(1)	No	No
	Export	Yes	No	No
	Properties	Yes	Yes	Yes
Sections(3) or Logic(3)	Refer to the tables below for services availability according to the operating mode.			
SR Sections	**New SR Section**	Yes	Yes	Yes
	Import	Yes	No	No
For each program module	**Open**	Yes	Yes	Yes
	Delete	Yes(2)	Yes(2)	Yes(2)
	Export	Yes	No	No
	Import after	Yes	No	No
	Properties	Yes	Yes	Yes

Directory	Services	Mode		
		Off-line	On-line Stop	On-line Run
Timer Events, I/O Events	**New Event Section**	Yes	Yes	Yes
	Import	Yes	No	No
For each event processing	**Open**	Yes	Yes	Yes
	Delete	Yes(2)	No	No
	Detach	Yes	Yes	Yes
	Export	Yes	No	No
	Properties	Yes	Yes	Yes

(1) Except for MAST task.

(2) Only if the program module or the event processing is not called.

(3) Depends on the CPU.

The **Sections** directory under each task allows you to access the following services.

Directory	Services	Mode		
		Off-line	On-line Stop	On-line Run
Sections	**New Section**	Yes	Yes	Yes
	Create section activation conditions table	Yes	Yes	Yes
	Import	Yes	No	No
For each section	**Open**	Yes	Yes	Yes
	Delete	Yes	Yes	Yes
	Detach	Yes	Yes	Yes
	Export	Yes	No	No
	Import after	Yes	No	No
	Force to 0	No	Yes	Yes
	Force to 1	No	Yes	Yes
	Unforce	No	Yes	Yes
	Properties	Yes	Yes	Yes

The **Logic** directory under each task allows you to access the following services.

Directory	Services	Mode		
		Off-line	On-line Stop	On-line Run
Logic	**New Program Unit**	Yes	Yes	Yes
	New Section	Yes	Yes	Yes
	Create builded activation conditions table	Yes	Yes	Yes
	Import	Yes	No	No
For each Program Unit	**Analyze**	Yes	Yes	Yes
	Delete	Yes	Yes	Yes
	Detach	Yes	Yes	Yes
	Copy	Yes	Yes	Yes
	Paste After	Yes	Yes	Yes
	Paste Special after	Yes	Yes	Yes
	Import after	Yes	No	No
	Export	Yes	No	No
	Properties	Yes	Yes	Yes
Sections (under Program Unit)	**New Section**	Yes	Yes	Yes
	Import	Yes	No	No
For each section of the Program Unit	**Open**	Yes	Yes	Yes
	Delete	Yes	Yes	Yes
	Export	Yes	No	No
	Import after	Yes	No	No
	Force to 0	No	Yes	Yes
	Force to 1	No	Yes	Yes
	Unforce	No	Yes	Yes
	Properties	Yes	Yes	Yes
Animation tables (under Program Unit)	**New Animation Table**	Yes	Yes	Yes
	Paste	Yes	Yes	Yes
	Detach All	Yes	Yes	Yes
	Delete All	Yes	Yes	Yes
	Import	Yes	No	No

Directory	Services	Mode		
		Off-line	On-line Stop	On-line Run
	Export	Yes	No	No
	Make All Tables Permanent	Yes	Yes	Yes
	Purge Temporary Tables	Yes	Yes	Yes
	Open Forced Bits Table	No	Yes	Yes
For each animation table of the Program Unit	**Open**	Yes	Yes	Yes
	Copy	Yes	Yes	Yes
	Paste	Yes	Yes	Yes
	Export	Yes	No	No
	Delete	Yes	Yes	Yes
	Detach	Yes	Yes	Yes
	Properties	Yes	Yes	Yes
For each section (under Logic)	**Open**	Yes	Yes	Yes
	Delete	Yes	Yes	Yes
	Detach	Yes	Yes	Yes
	Export	Yes	No	No
	Import after	Yes	No	No
	Force to 0	No	Yes	Yes
	Force to 1	No	Yes	Yes
	Unforce	No	Yes	Yes
	Properties	Yes	Yes	Yes

Animation tables Directory

The **Animation tables** directory, page 217 allows you to access the following services.

Directory	Services	Mode		
		Off-line	On-line Stop	On-line Run
Animation tables	**New Animation Table**	Yes	Yes	Yes
	Paste	Yes	Yes	Yes
	Detach All	Yes	Yes	Yes
	Delete All	Yes	Yes	Yes
	Import	Yes	No	No
	Export	Yes	No	No
	Make All Tables Permanent	Yes	Yes	Yes
	Purge Temporary Tables	Yes	Yes	Yes
	Open Forced Bits Table	No	Yes	Yes
For each animation table	**Open**	Yes	Yes	Yes
	Copy	Yes	Yes	Yes
	Paste	Yes	Yes	Yes
	Export	Yes	No	No
	Delete	Yes	Yes	Yes
	Detach	Yes	Yes	Yes
	Properties	Yes	Yes	Yes

Operator Screens Directory

The **Operator Screens** directory, page 220 allows you to access the following services.

Directory	Services	Mode		
		Off-line	On-line Stop	On-line Run
Operator Screens	**New screen**	Yes	Yes	Yes
	New family	Yes	Yes	Yes
	Detach All	Yes	Yes	Yes
	Delete All	Yes	Yes	Yes
	Export	Yes	No	No
	Import	Yes	No	No
	Messages list	Yes	Yes	Yes
For each family of screens	**New screen**	Yes	Yes	Yes
	Copy	Yes	Yes	Yes
	Paste	Yes	Yes	Yes
	Delete	Yes	Yes	Yes
	Detach	Yes	Yes	Yes
	Export	Yes	No	No
	Family Properties	Yes	Yes	Yes
For each screen	**Open**	Yes	Yes	Yes
	Copy	Yes	Yes	Yes
	Paste	Yes	Yes	Yes
	Delete	Yes	Yes	Yes
	Detach	Yes	Yes	Yes
	Export	Yes	No	No
	Screen properties	Yes	Yes	Yes

Documentation Directory

The **Documentation** directory, page 222 allows you to access the following services.

Directory	Services	Mode		
		Off-line	On-line Stop	On-line Run
Documentation	**Open**	Yes	Yes	Yes
Title Page	**Print Setup**	Yes	Yes	Yes
General Information	**Open**	Yes	Yes	Yes

Functional modules

What's in This Chapter

Introduction to the functional view 237
The functional modules and their associated utilities 251

Aim of this Chapter

This chapter introduces the functional modules incorporated in a Control Expert project.

Introduction to the functional view

About this sub-chapter

This sub-chapter provides general information on the functional modules and their associated utilities.

Functional modules

Definition

A functional module is a group of program elements intended to perform a PLC function.

A functional module is comprised:

- Program Units (only for Modicon M580 and M340)
- program or event processing sections,
- animation tables associated with the functional module,
- runtime screens associated with the functional module,
- a comment (max. 1024 characters),
- interleaved functional modules, with these modules, in relation to the main function, performing one or more PLC sub-functions.

  NOTE: a functional module does not necessarily have a program section, an animation table or runtime screens.

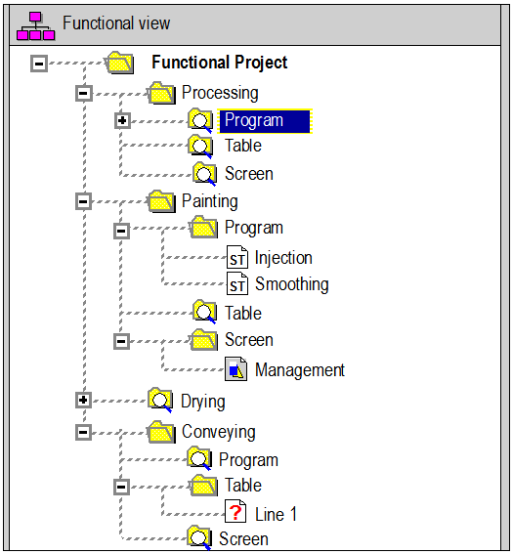
The other elements that make up a project are not incorporated into the functional modules:

- the configuration,
- the global variables of a project,
- the DFB and DDT types, which are global to a project,
- the SR program modules, which are global to a task,
- the documentation.

At a glance

All the functional modules of the project can be found in the **Functional Project** directory of the function view.

The following illustration shows the **Functional Project** directory:



NOTE: When Program Units are allowed in the application (Modicon M580 and M340), the **Program** folder is replaced by the **Logic** folder in each functional module directory.

As the default the browser displays the first level of the directory tree. To access the other levels, you need to expand the directories.

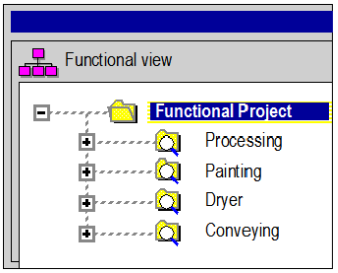
NOTE: This breakdown does not take into account the order of execution of the program by the PLC.

Project directory of the functional view

At a Glance

The **Functional Project** directory of the functional view allows you to access all the functional modules of the project and the associated services.

The following illustration shows the **Functional Project** directory:



Associated services

The **Functional Project** directory allows you to access the following services, which can be reached via the contextual menu:

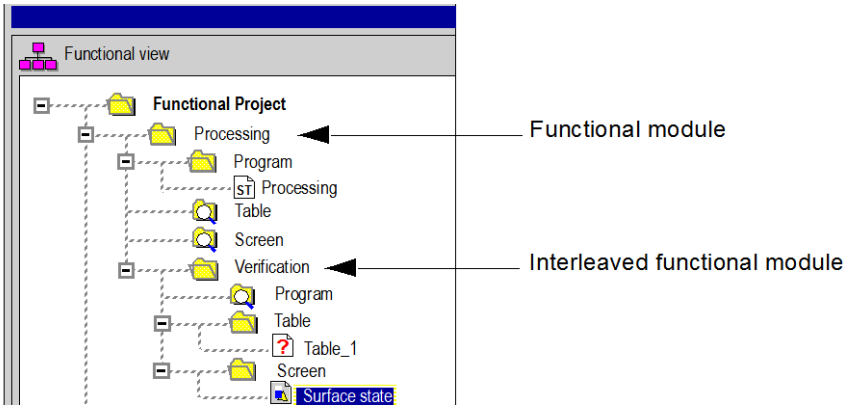
Directory	Services
Functional Project	**New Functional Module**: allows you to create a new functional module. **Detach all**: allows you to detach all elements included in the functional modules (sections and animation tables) without deleting them. **Export**: allows you to access export of the global project, page 1720. **Import**: allows you to access import of a functional module, page 1719. **Project Settings**: allows you to access the project specific settings, page 551. **Properties**: allows you to access the properties of the global project.
For each functional module	allows you to access all the elements of the functional module: program, animation tables, operator screens, nested functional modules.

Functional Module Directory

At a Glance

The **Functional Module** directory of the functional view allows you to access all the elements of a functional module (program, animation tables and operator screens) and the nested modules.

The following illustration shows an example of a **Functional Module** directory:



Associated services

The **functional module** directory allows you to access the following services, which can be reached via the contextual menu:

Directory	Services
Functional module: Processing	**New Functional Module...**: allows you to create a new nested functional module. **Create...**: allows you to create a new program element in the functional module. **Delete**: allows you to delete the functional module. **Detach all**: allows you to detach all elements included in the functional module (sections and animation tables) without deleting them. **Export**: allows you to access export of the functional module, page 1718. **Import**: allows you to access import of a functional module, page 1719. **Create builded Activation Conditions Table** (CTRL+T): allows you to initialize an animation table, with the activation conditions associated with the program elements of the functional module (sections, program modules and event processings). **Protection of Included Section(s)...**: allows you to define the protection of the program elements (sections, program modules and event processings) of the function module (no protection, write-protection, read/write-protection). **Properties**: allows you to access the functional module properties.
Program or **Logic**	allows you to access the functional module programs, page 241.
Table	allows you to access the animation tables of the functional module, page 242.
Screen	allows you to access the operator screens of the functional module, page 243.

Functional Module Program Directory

At a Glance

The program directory of a functional module (**Program** or **Logic**) allows you to access the program elements of this module: Program Units (when available), sections, program modules, and event processing.

Associated services

The program directory of a functional module allows you to access the following services, which can be reached via the contextual menu:

Directory	Services
Program or **Logic**	**Create...**: allows you to create a new program element in the functional module: Program Unit, section, program module or event processing. **Detach all**: allows you to detach all elements included in the functional module (Program Units, sections, and animation tables) without deleting them. **Import...**: allows you to import a program element to the functional module: Program unit, section, program module, event processing (see also topic on program import, page 1701). **Create builded Activation Conditions Table** (CTRL+T): allows you to initialize an animation table, with the activation conditions associated with the program elements of the functional module: sections. **Protection of Included Section(s)...**: allows you to define the protection of the program elements of the functional module: Program Units, sections, program modules and event processing.
Program Unit, Section, SR section, event processing	See **Programs** Directory, page 213 for a description of the associated services.

Functional Module Animation Tables Directory

At a Glance

The **Table** directory of a functional module allows you to access the animation tables of this module.

Associated services

The **Table** directory of a functional module allows you to access the following services, which can be reached via the contextual menu:

Directory	Services
Table	**New Animation Table**: allows you to create a new animation table. **Detach all**: allows you to detach all the animation tables from the functional module, without deleting them. **Import**: allows you to access import of the animation tables, page 1717 in the functional module.
For each animation table	**Open**: allows you to access the animation table. **Copy**: allows you to duplicate the animation table. **Paste**: allows you to add a copy of an animation table into the animation tables directory in the functional module and the project browser. **Export**: allows you to access export of the animation table, page 1716. **Delete**: allows you to delete the animation table. **Detach**: allows you to detach the animation table from the functional module, without deleting it. **Properties**: allows you to access the animation table properties.

Functional Module Operator Screens Directory

At a Glance

The **Screen** directory of a functional module allows you to access the operator screens of this module.

Associated services

The **Screen** directory of a functional module allows you to access the following services, which can be reached via the contextual menu:

Directory	Services
Screen	**Import**: allows you to access import of one or more screens, page 1715 and/or families of screens. **New family**: allows you to create a new family of screens. **New screen**: allows you to create a new operator screen. **Detach all**: allows you to detach all the screen elements from the functional module, without deleting them.
For each family of screens	See Operator Screens Directory, page 220 for a description of the screen families services.
For each operator screen	See Operator Screens Directory, page 220 for a description of the services available from the screen.

Summary of Services for the Functional Views

At a Glance

The following tables summarize the services associated with the functional view of the project and indicate the operating modes in which you can use them.

Functional Project Directory

The **Functional Project** directory, page 239 allows you to access these services.

Directory	Services	Mode	
		Off-line	On-line
Functional Project	**New Functional Module**	Yes	Yes
	Detach All	Yes	Yes
	Export	Yes	No
	Import	Yes	No
	Project Settings	Yes	Yes
	Properties	Yes	Yes

Functional Module Directory

The **Functional Module** directory, page 240 allows you to access these services.

Directory	Services	Mode	
		Off-line	On-line
For each functional module	**New Functional Module**	Yes	Yes
	Detach All	Yes	Yes
	Create	Yes	Yes
	Delete	Yes	Yes
	Export	Yes	No
	Import	Yes	No
	Protection of Included Sections	Yes	Yes
	Create builded Activation Conditions Table	Yes	Yes
	Properties	Yes	Yes

Functional Module Program Directory

The program directory, page 241 of a functional module allows you to access these services.

Directory	Services	Mode	
		Off-line	On-line
Program(1)	**Create**	Yes	Yes
or	**Import**	Yes	No
Logic(1)	**Detach All**	Yes	Yes
	Protection of Included Sections	Yes	Yes
	Create builded Activation Conditions Table	Yes	Yes
For each Program Unit	Refer to the table below for services availability according to the operating mode.		
For each section, event processing	**Open**	Yes	Yes
	Delete	Yes	Yes
	Detach	Yes	Yes
	Export	Yes	No
	Import After	Yes	No
	Force to 0	No	Yes
	Force to 1	No	Yes
	Unforce	No	Yes
	Properties	Yes	Yes

(1) Depends on the controller

The Program Unit directory, page 241 of a functional module allows you to access these services.

Directory	Services	Mode	
		Off-line	On-line
For each Program Unit	**Analyze**	Yes	Yes
	Delete	Yes	Yes
	Detach	Yes	Yes
	Copy	Yes	Yes
	Paste After	Yes	Yes
	Paste Special after	Yes	Yes
	Import After	Yes	No
	Export	Yes	No
	Force to 0	No	Yes
	Force to 1	No	Yes
	Unforce	No	Yes
	Properties	Yes	Yes
Sections (under a Program unit	**New Section**	Yes	Yes
	Import	Yes	No
For each section of the Program Unit	**Open**	Yes	Yes
	Delete	Yes	Yes
	Detach	Yes	Yes
	Import After	Yes	No
	Export	Yes	No
	Force to 0	No	Yes
	Force to 1	No	Yes
	Unforce	No	Yes
	Properties	Yes	Yes

Directory	Services	Mode	
		Off-line	On-line
Animation tables (under Program Unit)	**New Animation Table**	Yes	Yes
	Paste	Yes	Yes
	Detach All	Yes	Yes
	Delete All	Yes	Yes
	Import	Yes	No
	Export	Yes	No
	Make All Tables Permanent	Yes	Yes
	Purge Temporary Tables	Yes	Yes
	Open Forced Bits Table	No	Yes
For each animation table of the Program Unit	**Open**	Yes	Yes
	Copy	Yes	Yes
	Paste	Yes	Yes
	Export	Yes	No
	Delete	Yes	Yes
	Detach	Yes	Yes
	Properties	Yes	Yes

Functional Module Animation Tables Directory

The **Table** directory, page 242 of a functional module allows you to access these services.

Directory	Services	Mode	
		Off-line	On-line
Table	**New Animation Table**	Yes	Yes
	Import	Yes	No
	Detach All	Yes	Yes
For each animation table	**Open**	Yes	Yes
	Export	Yes	No
	Delete	Yes	Yes
	Copy	Yes	Yes
	Paste	Yes	Yes
	Properties	Yes	Yes
	Detach	Yes	Yes

Functional Module Operator Screens Directory

The **Screen** directory, page 243 of a functional module allows you to access these services.

Directory	Services	Mode	
		Off-line	On-line
Screen	**New family**	Yes	Yes
	New screen	Yes	No
	Import	Yes	No
	Detach All	Yes	Yes
For each family of screens	**New screen**	Yes	Yes
	Copy	Yes	Yes
	Paste	Yes	Yes
	Delete	Yes	Yes
	Export	Yes	No
	Detach	Yes	Yes
	Family Properties	Yes	Yes
For each operator screen	**Open**	Yes	Yes
	Copy	Yes	Yes
	Paste	Yes	Yes
	Export	Yes	No
	Detach	Yes	Yes
	Delete	Yes	Yes
	Screen properties	Yes	Yes

The functional modules and their associated utilities

About this sub-chapter

This sub-chapter is a guide on how to use the utilities associated with the functional modules.

Properties of a functional module

Properties to define

List of different properties to define:

- the name comprising 32 characters. This name must be unique for modules positioned on the same level.
- the comment comprising 256 characters.

NOTE: The activation condition and the protection of the program elements (Program Units, sections, program modules and event processing) are defined under program element properties.

Displaying or editing properties

Carry out the following steps:

Step	Action
1	In the functional view, right-click the functional module, in the **Functional Project** directory.
2	Click **Properties**.
3	Perform the editing.
4	Confirm with **OK**.

NOTE: The **Apply** button confirms the editing without closing the window.

Creating a functional module

At a Glance

A functional module can be created off-line, with the PLC at **Stop** or in **Run**.

It can be created at the Project level or at the level of each existing functional module.

Creating a new functional module

Carry out the following actions:

Step	Action
1	In the functional view, right-click the **Functional Project** directory.
2	Select **New Functional Module**.
3	Enter the name and comment and confirm with **OK**.

Creating a lower level functional module

Carry out the following actions:

Step	Action
1	In the functional view, right-click the module "above", in the **Functional Project** directory.
2	Select **New Functional Module**.
3	Enter the name and comment and confirm with **OK**.

Moving a functional module

A functional module can be moved off-line, with the PLC at **Stop** or in **Run** (this has no effect on the execution of the project). The movement corresponds only to a modification of the functional architecture of the project (a module is directly attached to the **Functional Project** directory level or to another functional module).

Moving a functional module:

Step	Action
1	Left-click on the module to be moved (holding down the mouse button),
2	Move the module to the desired position.

Programming a functional module

Introduction to programming a functional module

A functional module has a program directory (**Program** or **Logic**) that may contain program elements:

- Program Unit (only for Modicon M580 an M340)
- LD, ST, FBD, IL sections
- event processing
- SFC sections

For more detailed information, refer to chapter *Application Program structure* (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).

Various cases may arise when programming a functional module:

- **Case 1**: the program element already exists in the structural view
- **Case 2**: creating the program element using the functional view
- **Case 3**: importing a program element using the functional view

Case 1: the program element already exists in the structural view

The program element has already been created in the structural view:

Step	Action
1	Select the program element.
2	Move the program element to the functional module.

Case 2: creating the program element using the functional view

The following table shows how to create a program element:

Step	Action
1	In the **Functional view**, right-click the **Program** or **Logic** directory in the functional module directory.
2	Select the **Create ...** command from the contextual menu.
3	Select either **New Program Unit ...**, **New Section ...**, **New Timer Event ...** or **New IO Event ...**.
4	Enter the various headings in exactly the same way as for creating a program element from the structural view. The name of the functional module is shown again at the structural view level.

Case 3: importing a program element using the functional view

The following table shows how to import a program element:

Step	Action
1	In the **Functional view**, right-click the **Program** or **Logic** directory in the functional module directory.
2	Select the **Import ...** command from the contextual menu.
3	In the **Import** dialog box, select the file to be imported.
4	Click on the **Import** button to finish the import.

When using the functional view to import a program element that does not have its task in the importing application, the program element is attached to the **MAST** task.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

Verify that an imported program element operates correctly while attached to the **MAST** task if it was originally running in a non-MAST task.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Rules

Take into account the following notes:

1	An SFC section can only be created off-line and then only in the **MAST** task.
2	A macro-step or an event can only be created off-line.
3	The other actions are authorized in off-line mode, with the controller at **Stop** or in **Run**.
4	The protection of a module applies to all the program elements attached to the functional module and to the lower level modules.

Protecting program elements of the functional module

At a Glance

In a functional module you are able to define the protection, page 192 of the program elements contained in the module.

Protection

Procedure:

Select from	Then...
the functional module	• with the mouse right-click the functional module in the **Functional Project** directory of the functional view, • select **Protection of Included Section(s) ...**, • select **Read only** or **No read & write**. **Result**: All the program elements of the module are protected.
from the **Program** or **Logic** directory	• with the mouse right-click on the **Program** or **Logic** directory in a functional module of the functional view, • select **Protection of Included Section(s) ...**, • select **Read only** or **No read & write**. **Result**: All the program elements of the module are protected.

Debugging a functional module

At a Glance

The organization of a functional module, and the distribution of the Program Units, sections, event processing and SFC sections in the various modules has no impact on the execution

of the program. The program is executed in the order shown in the structural view. To help you debug a functional view, the following are available:

- basic debugging functions,
- the animation table initialization function.

Initializing an animation table

This action allows you to initialize an animation table, with the activation conditions associated with the program elements of the functional module.

Select from	Then...
the functional module	• right-click with the mouse on the directory of a functional module in the functional view. • select **Create builded Activation Conditions Table**.
from the **Program** or **Logic** directory	• with the mouse right-click on the **Program** or **Logic** directory in a functional module of the functional view. • select **Create builded Activation Conditions Table**.

NOTE: All the activation conditions that are displayed in the animation table are the conditions used during the last build (partial or total).

Detach or Delete a Functional Module

At a Glance

To detach elements from a functional module, cut the links between the module and the associated elements (program elements, animation tables, screens):

- The Program Units and sections contained in the module are not deleted but detached.
- The animation tables associated with the module are not deleted but detached from the module.
- The screens associated with the module are not deleted but detached from the module.

NOTE:

- These actions are authorized off-line and on-line when the controller is in Stop or Run.
- To detach functional modules and the associated directory elements, refer to the description of services associated with the functional view, page 244.

Deleting Functional Modules

There are several ways to delete functional modules:

- Delete one or more functional modules without deleting the program elements, animation tables, and screens.
- Delete one functional module while deleting the program elements, animation tables, and screens.
- Delete all functional modules while deleting the program elements, animation tables, and screens.
- Delete a section, a macro-step, or an event processing in a functional module.

Delete Functional Modules while Retaining Program Elements, Animation Tables, and Screens

Delete a functional module without deleting the program elements and animation tables:

Step	Action
1	From the functional module, detach the module with the **Detach all** command in the contextual menu. **Result**: All the elements of the module and the lower level modules are detached.
2	Delete the module(s) by right-clicking (contextual menu) and selecting the **Delete** command. **Note:** You must be in **off-line** mode.

Delete a Functional Module while Deleting Modules, Animation Tables, and Screens

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

Before you delete a module, confirm that the consequential deletion of these elements has no adverse consequence on the behavior of the application:

- The Program Units contained in the module
- The sub-modules contained in the module
- The sections contained in the module
- The associated animation tables
- The associated screens

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Delete a functional module while deleting modules, animation tables, and screens:

Step	Action
1	Select the module.
2	Select **Delete**.
3	Confirm with **Yes**.

NOTE: This action is authorized for the SFC sections and the events in off-line mode, with the controller at **Stop**. The other elements may be deleted in **Run**.

Delete a Program Unit, Section, or Event in a Functional Module

When deleting a Program Unit, or a section, or an event in a functional module, they are then deleted in the associated task.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

Before you delete a Program Unit, section, or event in a functional module, confirm that the deletion has no adverse consequence on the behavior of the application.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Delete a Program Unit, section, or event:

Step	Action
1	Select the desired Program Unit, or section, or event.
2	Select **Delete**.
3	Confirm with **Yes**.

Rules

Observe these guidelines:

- To retain the contents of the module, detach its elements before you delete it.
- You can delete an SFC section or an event in off-line mode only.

Functional module export

Exporting a functional module

You can export, page 1718 a functional module.

Functional module import

Importing a functional module

You can import, page 1719 a functional module.

Creating, Deleting, Locating, Dragging-and-Dropping an Animation Table in a Functional Module

At a Glance

In a functional module you are able to create, delete, locate and drag-and-drop an animation table.

Creating an Animation Table

Procedure:

If the table	Then...
already exists	• select the animation table in the **Functional Project** directory of the functional view or the **Animation tables** directory of the structural view, • drop the animation table to a functional module in the **Table** directory level.
needs to be created from the functional view	• with the mouse right-click the **Table** directory of the module, • select **New Animation Table**.

Accessing the Properties of an Animation Table

Carry out the following actions:

Step	Action
1	Select the table.
2	Select **Properties** from the contextual menu.
3	You can edit the name or the comment of a table and its assignment to a module.

Copying an Animation Table.

Procedure:

If the table	Then...
needs to be copied from the functional view.	• with the mouse right-click the source table in the **Functional Project** directory of the functional view, • select **Copy**, • select the source table in the **Functional Project** directory of the functional view, • select **Paste**. **Result**: The copy of the table is created in the source table directory.

Opening an Animation Table

Carry out the following actions:

Step	Action
1	Select the table.
2	Select **Open** from the contextual menu.

Deleting an Animation Table

Carry out the following actions:

Step	Action
1	Select the table.
2	Select **Delete** from the contextual menu.

Moving an Animation Table

Procedure:

If using	Then...
the drag-and-drop function	• left-click on the table to be moved (holding down the mouse button), • drop the table at the desired position. Note: The insertion point of the table is represented by a line.
Properties from the contextual menu	• select the table, • select **Properties** from the contextual menu, • select the name of the module in the **Functional Module** zone.

Detach all Animation Tables

Procedure:

If using the contextual menu	Then...
Detach all	• select the **Table** directory of the functional module, • select **Detach All** from the contextual menu, • confirm with **Yes**.

Detaching an Animation Table

Procedure:

If using the contextual menu	Then...
Detach	• select the animation table, • select **Detach** from the contextual menu, • confirm with **Yes**.
Properties	• select the table, • select **Properties** from the contextual menu, • select "none" in the **Functional Module** zone.

Creating, Deleting, Locating and Dragging-and-Dropping an Operator Screen in a Functional Module

At a Glance

In a functional module you are able to create, delete, locate and drag-and-drop an operator screen.

Creating an Operator Screen

Procedure:

If the screen	Then...
already exists	• select the operator screen in the **Functional Project** directory of the functional view or the **Operator Screens** directory of the structural view, • in any language editor window, select **Open Operator Screen.** in the function block contextual menu. • drop the operator screen in a functional module at the **Screen** directory level. **NOTE:** With the **Open Operator Screen** menu if multiple operator screens exist with the same name, the editor opens the first one found.
needs to be created from the functional view	• with the mouse right-click the **Screen** directory of the functional module, • select **New screen**.

Accessing the Operator Screen Properties

Procedure:

If selected at the level of the	Then...
a family of screens	• select the family of screens in the **Screen** directory of the structural view, • Select **Family Properties** from the contextual menu.
a specific screen	• select a screen in the **Screen** directory of the structural view, • Select **Screen Properties** from the contextual menu.

Creating a Family of Operator Screens

Procedure:

If the family	Then...
already exists	• select the family of operator screens in the **Functional Project** directory of the functional view or the **Operator Screen** directory of the structural view, • drop the family of operator screens in a functional module at the **Screen** directory level.
needs to be created from the functional view	• with the mouse right-click the **Screen** directory of the functional module, • select **New family**.

Opening an Operator Screen

Carry out the following actions:

Step	Action
1	Select the operator screen in the **Functional Project** directory of the functional view or the **Operator Screens** directory of the structural view.
2	Select **Open** from the contextual menu.

Deleting an Operator Screen

Procedure:

If the screen	Then...
does not belong to a family of screens	• with the mouse right click the operator screen in the **Operator Screen** directory, • select **Delete**.
belongs to a family of screens	• from the **Screen** directory, right-click the operator screen belonging to the family, • select **Delete**.

Moving an Operator Screen

Procedure:

If using	Then...
the drag-and-drop function	• left-click on the operator screen to be moved (holding down the mouse button), • move the operator screen to the desired location. **NOTE:** The insertion point of the operator screen is represented by a line.
Screen Properties from the contextual menu	• select the operator screen, • select **Screen Properties** from the contextual menu, • select the name of the module in the **Location** zone.

Detaching All Screens and Families of Screens

Procedure:

If you have chosen to detach	Then...
the operator screens and the families of screens	• select the **Screen** directory of the functional module, • select **Detach All** from the contextual menu, • confirm with **Yes**.
the family of screens	• in the **Screen** directory of the functional module, select the family of screens, • select **Detach** from the contextual menu,
a specific screen	• select the operator screen in the **Screen** directory of the functional module, • select **Detach** from the contextual menu,

Accessing the Messages List

The following table shows the procedure for accessing the operator screens messages list.

Step	Action
1	Select the **Operator Screens** directory from the structural view.
2	Select **Messages list** from the contextual menu.

FDT Container

What's in This Chapter

FDT Container 267
DTM Hardware Catalog 270
DTM Browser 283
DTM Properties Tabs 309

Object of the Chapter

This chapter introduces the FDT Container and explains how it is used.

FDT Container

Object of this Section

This section is an introduction to using DTMs in Control Expert.

Introduction to the Control Expert FDT Container

At a Glance

The FDT/DTM (**Field Device Tool**/**Device Type Manager**), an open-standard, in Control Expert integrates the configuration tools needed to build projects, with a mix of Schneider and Third Party intelligent devices, on heterogeneous networks.

Control Expert is now a **FDT Container** (Field Device Tool Container) for DTMs and, therefore, is a Generic Fieldbus Intergrater.

NOTE: Schneider-Electric is not responsible for the correct functioning of non-Schneider-Electric DTMs under Control Expert.

FDT Concept

The FDT defines the interfaces between device-specific software components (DTMs) and Control Expert. The focus of the current version of FDT is on engineering, commissioning, diagnostics and documentation of fieldbus based control systems.

The device-specific software component is called a DTM (Device Type Manager), which are supplied by the device manufacturer together with the device. Because the FDT technology used in Control Expert is based on an open standard, Control Expert accepts compliant DTM products from any DTM/device vendor.

NOTE: The DTMs are installed/uninstalled on the PC (like most other Windows® programs) that runs Control Expert and information about the DTMs is kept in the Windows Registry.

NOTE: The DTM Browser displays the DTM connectivity configuration, but does not create any topological variables (%I or %Q) in the application.

Components

To create a Fieldbus network with Control Expert, its FDT Container uses:

- The DTM Hardware Catalog, page 271
- The DTM Browser, page 283
- The DTM Editor, which uses vendor supplied GUIs to configure the DTMs

  NOTE: The DTM vendor is responsible for the DTM Configuration on-line Help, it is not available through the Control Expert on-line Help.

FDT Container User Rights

At a Glance

With 2 exceptions, a **FDT Container** user has unlimited access to the installed DTMs.

User Rights

The **FDT Container** user has all rights of access for the DTM connectivity tree (add, delete, configure, save DTMs) unless:

1. The Control Expert profile is set to Read Only for the user
2. The user has opened Control Expert in the Read Only mode

Control Expert Online/Offline and Device Connections

For the user to load and store DTM device parameters from/to its physical device, the device must be connected to its DTM through the Host PC.

This connection:

- Requires that the DTM be installed on the Host PC.
- Is **independent** of the Control Expert - PLC connection status. A DTM can connect to its device and transfer configuration information over the network in either direction under both conditions: Control Expert off-line or Control Expert on-line.

Limitation

If the user opens a project in Read Only mode, the DTM user right remains in Read-Only as long as the project is still opened (even if the user removes the Read-Only opened mode by performing the service Save As).

DTM Hardware Catalog

Object of the Section

This section explains the DTM **Hardware Catalog**.

Introduction to the Hardware Catalog

Introduction

The Control Expert **Hardware Catalog** contains a list of modules and devices that you can add to a Control Expert project. EtherNet/IP and Modbus TCP devices are located in the **DTM Catalog** tab at the bottom of the **Hardware Catalog**. Each module or device in the catalog is represented by a DTM that defines its parameters.

EDS Files

Not all devices in today's market offer device-specific DTMs. Some devices are defined by device-specific EDS files. Control Expert displays EDS files in the form of a DTM. In this way, you can use Control Expert to configure devices that are defined by an EDS file in the same way you would configure a device defined by its DTM.

Other devices lack both a DTM and an EDS file. Configure those devices by using the generic DTM on the **DTM Catalog** page.

View the Hardware Catalog

Open the Control Expert **Hardware Catalog**:

Step	Action
1	Open Control Expert.
2	Find the **PLC bus** in the **Project Browser**.
3	Use one method to open the catalog: • Use the pull-down menu (**Tools** > **Hardware Catalog**). • Double-click an empty slot in the **PLC bus**.

Description of DTM Hardware Catalog

At a Glance

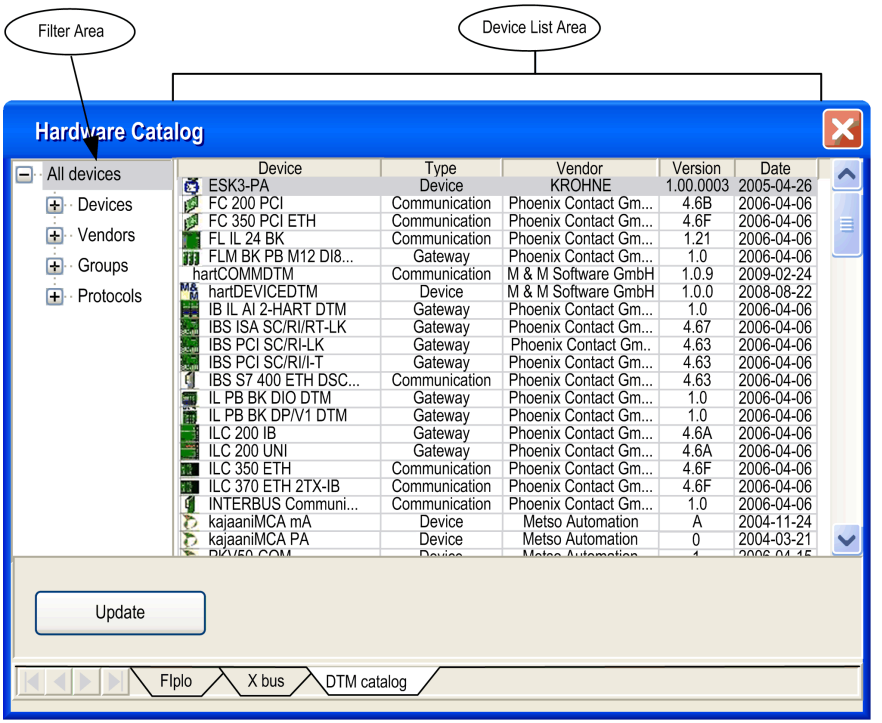
The DTM **Hardware Catalog** displays all the DTMs installed on the PC.

The DTM tab of the **Hardware Catalog** is divided into 3 parts:

- The Filter area
- The Device List Area
- The Update button

The DTM Catalog Tab

The following illustration is an example of the DTM **Hardware Catalog**:



Add DTMs to Connectivity Tree

There are 2 ways to add DTMs to a DTM connectivity tree:

- By using the Add device, page 297 procedure in the DTM Browser
- By using the mouse to drag a DTM from the Hardware Catalog and dropping it on an appropriate DTM in the connectivity tree

The Filter Area

The browser in the **Filter Area** has 4 types of filters with sub-filters. The actual sub-filters available depend on the type of DTMs installed on the PC:

- Devices

  There are 3 device type sub-filters:

  - Communication
  - Gateway or Modular
  - (Other) Devices
- Vendors

  Selecting a vendor name displays only the installed DTMs from that vendor.
- Groups

  Selecting a group type displays only the installed DTMs corresponding to that type of DTM.
- Protocols

  Selecting a protocol displays only the installed DTMs corresponding to that type of protocol.

  When a protocol is selected as the filter, a DTM is only shown if this protocol is required and/or supported by the DTM.

When a filter is selected, only the DTMs corresponding to that filter are displayed in the **Device List Area**.

The Device List Area

The **Device List Area** columns give information about each DTM. They are:

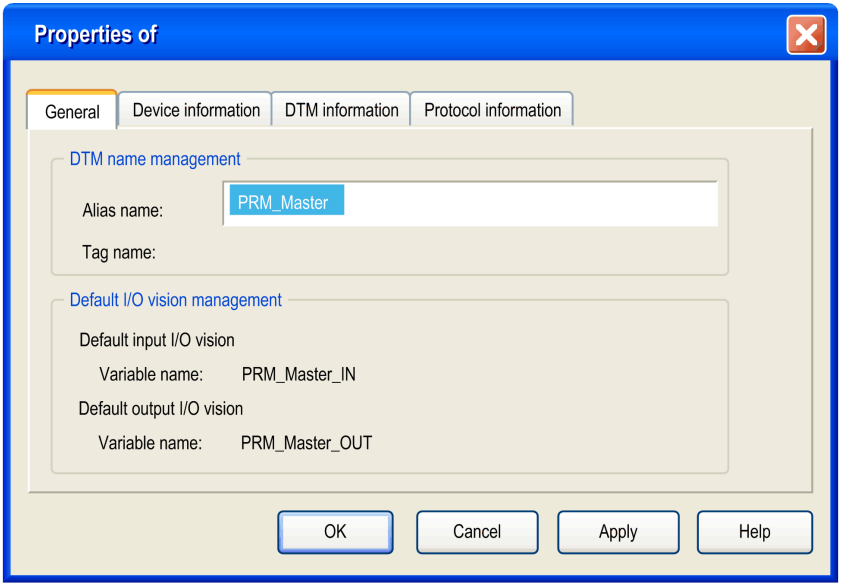
- Device (name)
- Type
- Vendor (name)
- Version

- Date

Clicking on a column heading sorts the DTMs alphabetically according the contents of the column.

DTM Properties

To display the properties of a DTM from the **Device List Area**:

Step	Action
1	Click anywhere on the line of the DTM. Result: The DTM is selected.
2	Right-click on the DTM. Result: The Properties menu opens.
3	Use the tabs, page 309 to see additional information about the DTM: 
4	When finished with the window, click on **OK** or **Cancel**.

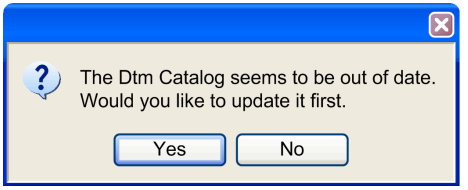
NOTE: When installing or modifying a DTM, update the Hardware Catalog, page 274 to update its properties information.

Update Hardware Catalog

The DTM catalog is fully embedded in the Control Expert software and requires to be up to date whenever a DTM is installed or unistalled. It can be done:

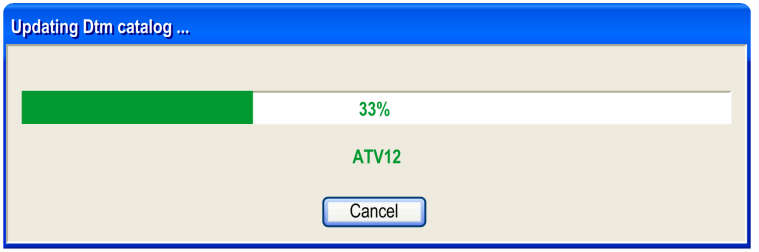
- automatically when the Control Expert software is launched in closed project mode
- or manually, from the **Hardware Catalog** window, anytime the customer requests an update.

When the Control Expert software is launched, in closed project mode, and DTMs are installed or uninstalled on/from the PC, the DTM catalog must be updated using the following procedure:

Step	Action
1	When DTMs are installed or uninstalled and the Control Expert software is launched. **Results:** The dialog box update opens: 
2	Click on the **Yes** button. **Results:** When the update is completed, a message box appears:

NOTE: The detection service is available as of FDT/DTM version 1.2.1.

The DTM catalog can be updated manually update can be done using the following procedure:

Step	Action
1	From the hardware Catalog, page 274, click on the **Update** button. **Results:** The dialog box update opens:
2	Click on the **Yes** button. **Results:** The standard Control Expert progress control dialog is displayed: 

Updating the **Hardware Catalog** does not modify the build status of the DTM. It creates an internal catalog file based on the installed DTMs existing in the Windows® registry.

The time necessary for an update depends on the number for DTMs installed and their sizes.

DTM Version Compatibility

The compatibility of DTM versions is managed by the vendor. They cannot be managed in Control Expert.

During an update no check is made if a DTM missing from the **Hardware Catalog** is used in an application.

There are 2 methods to check and display any inconsistencies:

- Build the whole Control Expert application
- Run the Check devices, page 293 service from the **HostPS** contextual menu

Adding a DTM to the Control Expert Hardware Catalog

A Manufacturer-Defined Process

Before a DTM can be used by the Control Expert **Hardware Catalog**, install the DTM on the host PC (the PC that is running Control Expert).

The installation process for the DTM is defined by the device manufacturer. Consult the documentation from the device manufacturer to install a device DTM on your PC.

NOTE: After a device DTM is successfully installed on your PC, update the Control Expert Hardware Catalog to see the new DTM in the catalog. The DTM can then be added to a Control Expert project.

Adding an EDS File to the Hardware Catalog

Introduction

You may want to use an EtherNet/IP device for which no DTM is in the catalog. In that case, use these instructions to import the EDS files into the catalog to create a corresponding DTM.

Control Expert includes a wizard you can use to add one or more EDS files to the Control Expert **Hardware Catalog**. The wizard presents instruction screens to execute these commands:

- Simplify the addition of EDS files to the **Hardware Catalog**.
- Provide a redundancy check when you add duplicate EDS files to the **Hardware Catalog**.

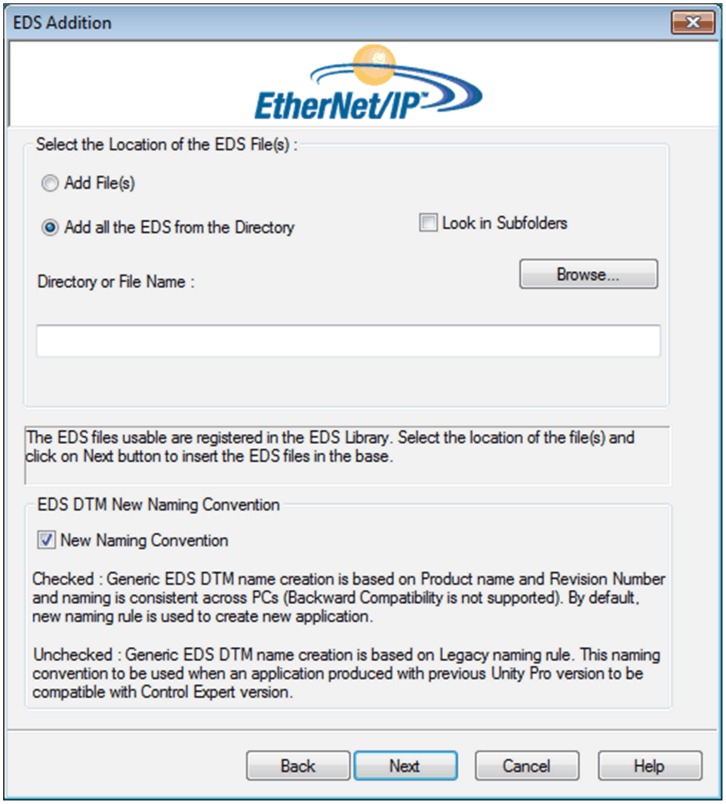
NOTE: The Control Expert **Hardware Catalog** displays a partial collection of DTMs and EDS files that are registered with the ODVA. This library includes DTMs and EDS files for products that are not manufactured or sold by Schneider Electric. The non-Schneider Electric EDS files are identified by vendor in the catalog. Please contact the identified device's manufacturer for inquiries regarding the corresponding non-Schneider Electric EDS files.

Adding EDS Files

Open the **EDS Addition** dialog box:

Step	Action
1	Open a Control Expert project that includes an Ethernet communication module.
2	Open the **DTM Browser** (**Tools > DTM Browser**).
3	In the **DTM Browser**, select a communication module.
4	Right-click on the communication module and scroll to **Device menu > Additional functions > Add EDS to library**.
5	In the **EDS Addition** window, click **Next**.

You can now see this page:



Add one or more EDS files to the library:

Step	Action
1	Use these commands in the **Select the Location of the EDS File(s)** area of the **EDS Addition** dialog box to identify the location of the EDS files: • **Add File(s)**: Add one or more EDS files that are individually selected. • **Add all the EDS from the Directory**: Add all files from a selected folder. (Check **Look in Subfolders** to add EDS files from the folders within the selected folder.)
2	Click **Browse** to open a navigation dialog box.
3	Select the location of the EDS file(s): • Navigate to at least one EDS file. • Navigate to a folder that contains EDS files. **NOTE:** Keep the location selected (highlighted).
4	Click **Select** to close the navigation window. **NOTE:** Your selection appears in the **Directory or File Name** field.

Step	Action
5	Choose the naming convention rule for the EDS DTM name creation. The new naming convention is based on Model Name / Product Name and Revision. A random character is automatically suffixed when Model Name / Product Name and Revision of an EDS file in the library is identical. The new naming convention is irrespective of the order in which EDS files are added to device library. By default, the **New Naming Convention** check box is selected and the new naming rule applies. **NOTE:** To keep backward compatibility with Unity Pro/Control Expert versions, unchecked the **New Naming Convention** check box and the naming rule is based on Model Name / Product Name.
6	Click **Next** to compare the selected EDS files to the files in the library. **NOTE:** If one or more selected EDS files is a duplicate, a **File Already Exists** message appears. Click **Close** to hide the message.
7	The next page of the **EDS Addition** wizard opens. It indicates the status of each device you attempted to add: • check mark (green): The EDS file can be added. • informational icon (blue): There is a redundant file. • exclamation point (red): There is an invalid EDS file. **NOTE:** You can click **View Selected File** to open and view the selected file.
8	Click **Next** to add the non-duplicate files. **Result**: The next page of the **EDS Addition** wizard opens to indicate that the action is complete.
9	Click **Finish** to close the wizard. **Result**: The hardware catalog automatically updates.

Removing an EDS File from the Hardware Catalog

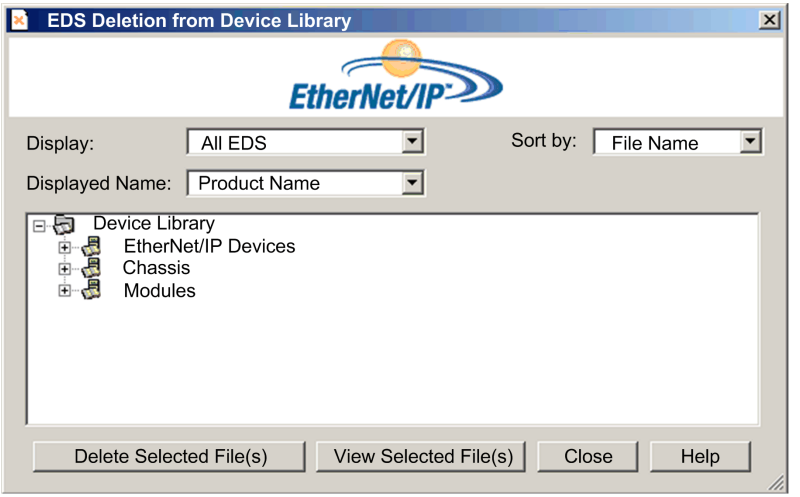
Introduction

You can remove a module or device from the list of available devices in the Control Expert **Hardware Catalog** by removing its **EDS** file from the library.

When you remove an EDS file from the library, the device or module disappears from the **DTM Catalog**. However, removing the file from the library does not delete the file from its stored location, so you can import the file again later.

Removing an EDS File from the Catalog

Use these steps to remove an EDS file from the catalog:

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td>1</td><td colspan="2">Open the Control Expert DTM Browser (Tools > DTM Browser).</td></tr>
<tr><td>2</td><td colspan="2">In the DTM Browser, select an Ethernet communication module.</td></tr>
<tr><td>3</td><td colspan="2">Right-click the module and scroll to Device menu > Additional functions > Remove EDS from library to open the EDS Deletion from Device Library window:
 
</td></tr>
<tr><td rowspan="4">4</td><td colspan="2">Use the selection lists in the heading of this window to specify how EDS files are displayed:</td></tr>
<tr><td>Display</td><td>Choose criteria to filter the list of EDS files:
• All EDS (no filtering)
• Only Devices
• Only Chassis
• Only Modules</td></tr>
<tr><td>Sort by</td><td>Choose criteria to sort the list of displayed EDS files:
• File Name
• Manufacturer
• Category
• Device Name</td></tr>
<tr><td>Displayed Name</td><td>Choose the identifier for each device:
• Catalog Name
• Product Name</td></tr>
</table>

Step	Action
5	Expand (+) the **Device Library** navigation tree and select the EDS file you want to remove. **NOTE:** Click **View Selected File** to see the read-only contents of the selected EDS file.
6	Click the **Delete Selected File(s)** button to open the **DeleteEDS** dialog box.
7	Click **Yes** to remove the selected EDS file from the list.
8	Repeat these steps for each EDS file you want to delete.
9	Click **Finish** to close the wizard. **Result**: The hardware catalog automatically updates.

Export / Import EDS Library

Introduction

To use the same project on two Control Expert installations (for example a source, and a target Host PCs), you may have to update the DTM **Hardware Catalog** of the target Host PC.

Instead of adding one by one the missing EDS files in the target Host PC, you can update the DTM **Hardware Catalog** in two steps:

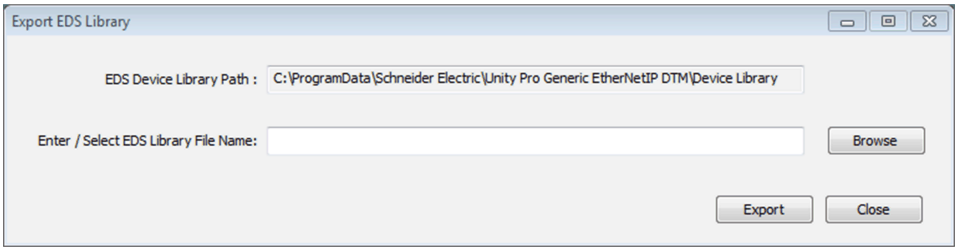
- Exporting the EDS library from the source Host PC.
- Importing the EDS library in the target Host PC.

NOTE: When you export the EDS library, the software generates an **.DLB** file which contains all the DTM created form EDS files.

Exporting EDS Library

Open the **Export EDS Library** dialog box:

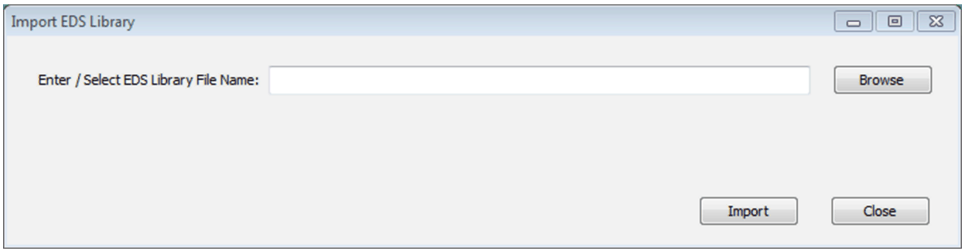
Step	Action
1	Open a Control Expert project that includes an Ethernet communication module.
2	Open the **DTM Browser** (**Tools** > **DTM Browser**).
3	In the **DTM Browser**, select a communication module.

Step	Action
4	Right-click on the communication module and scroll to **Device menu > Additional functions > Export EDS library** to open the **Export EDS library** window: 
5	For the archived EDS library you want to create: • Enter the full folder path along with the file name in the **Enter / Select EDS Library File Name** field, or • Click **Browse** to open a navigation dialog box: ◦ Select the location, and ◦ Enter the file name, and ◦ Click **Save** to close the navigation window and your selection appears in the **Enter / Select EDS Library File Name** field.
6	Click **Export** to create the archived EDS library. **Result:** A new wizard opens to indicate that the export is complete. Click **Ok** to close the wizard.
7	In the **Export EDS library** window, click **Close**.

Importing EDS Library

Use these steps to import an archived EDS library:

Step	Action
1	Open the Control Expert **DTM Browser** (**Tools > DTM Browser**).
2	In the **DTM Browser**, select an Ethernet communication module.

Step	Action
3	Right-click the module and scroll to **Device menu** > **Additional functions** > **Import EDS library** to open the **Import EDS library** window: 
4	For the archived EDS library you want to import: • Enter the full folder path along with the file name in the **Enter / Select EDS Library File Name** field, or • Click **Browse** to open a navigation dialog box: ◦ Select the location, and ◦ Enter the file name, and ◦ Click **Save** to close the navigation window and your selection appears in the **Enter / Select EDS Library File Name** field.
5	Click **Import**. **Result:** A new wizard opens to indicate that the export is complete. Click **Ok** to close the wizard.
6	In the **Import EDS library** window, click **Close**.

DTM Browser

About this Section

This section describes the DTM Browser, which enables the user to select the different DTMs that make up the DTM connectivity tree on the host PC.

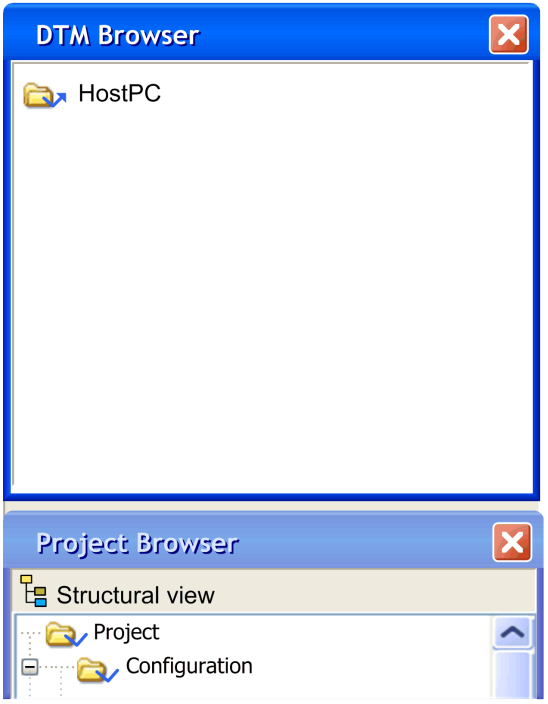
DTM Browser

At a Glance

The DTM Browser is the main component of the Control Expert FDT Container. The DTM devices are managed from the browser.

DTM Browser

To open the DTM Browser use the **Tools** menu -> DTM Browser:



The root of the DTM Browser is **HostPC**, which is the PC platform for the Control Expert project.

DTM Browser DTM Types

The DTM Browser displays the configured DTMs added to the connectivity tree **HostPC** with hierarchical levels (nodes).

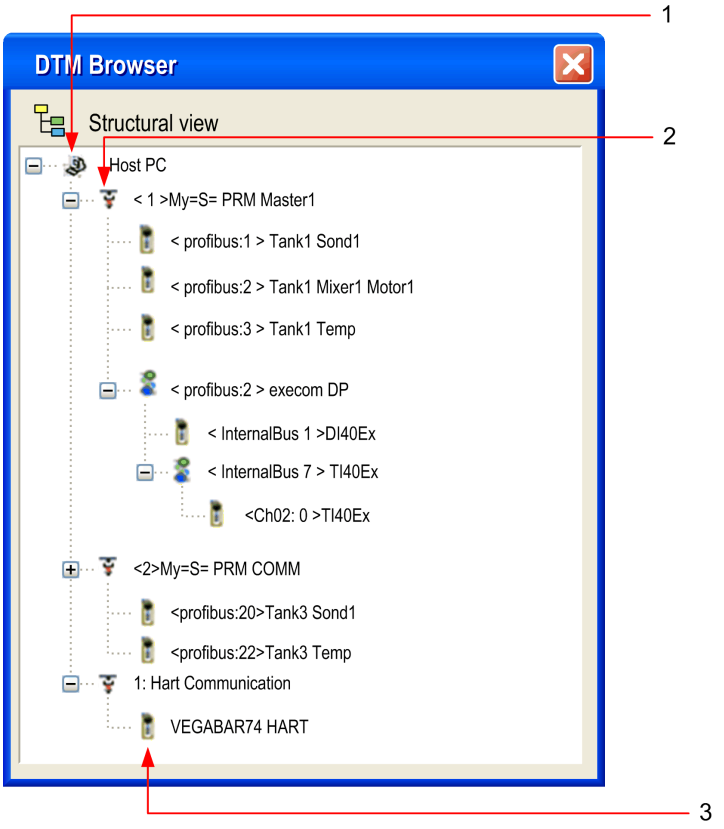
There are 3 types of DTMs:

- Communication DTMs (drivers)
  - Any COM DTM can be plugged directly under the root node (HostPC) and must be at the 1st level
  - A COM DTM can support Gateway DTMs or Device DTMs as children if their protocols are compatible
- Gateway DTMs

  A Gateway DTM can support other Gateway DTMs or Device DTMs as children if their protocols are compatible.
- Device DTMs

  A Device DTM does not support any child DTMs

The following illustration is a connectivity tree:



The following table describes the different hierarchical nodes of the connected DTMs in the above example connectivity tree:

Number	Description of Level
1	The First Level is the root node, HostPC, which is where the PC is connected to the physical network.
2	The Second Level devices are communication DTMs (drivers) that connect to their physical communication devices on the network.
3	The Third Level devices are connected to a communication device.

Copy Paste feature

Each device DTM configured can be copied and pasted to easily create configuration.

NOTE: Only Device DTMs can be copy and pasted.

Operating mode to make a Copy/Paste:

- Make a right click on the Device DTM you want to Copy/Paste
- Select **Copy**
- Make a right click on the Communication DTM or Gateway DTM in which you want to paste the Device DTM
- Select **Paste**.

  NOTE: If protocols are matching only device DTM can be paste under Communication DTM, only if protocols are matching. Communication DTM has to be in off-line mode.

  NOTE: If protocols are matching, only device DTM can be paste under Gateway DTM, only if protocols are matching. Gateway DTM has to be in off-line mode.

  NOTE: When a DTM is pasted in the connectivity Tree, the General tab, page 310 is displayed. By default, the copied DTM alias is based on a valid DTM alias name.

  NOTE: A default address is automatically created after the paste of a device DTM. The address can be modified when entering in the father DTM configuration screen. Copy/Paste feature has limitation, it can only be done inside an application and can not be done between two applications.

Tree View Naming

Each DTM has a default name when inserted into the browser. The syntax is the following: <Channel: Bus Address> User Name:

- Channel

  This is the name of the **channel communication media** where the DTM is plugged in. This name is read from the DTM and is set by the device vendor.

- Bus Address of the **device**. It can be either:
  - The connection point on its parent gateway network
  - The slot number in the modular device parent internal bus

  This information is supplied by the parent DTM and can be modified in Control Expert if the DTM supports this modification.

- User Defined Name

  This name is set by default to the vendors name for the DTM name (which is the default vendor name), but can be reset, page 306.

Examples:

- For a communication DTM

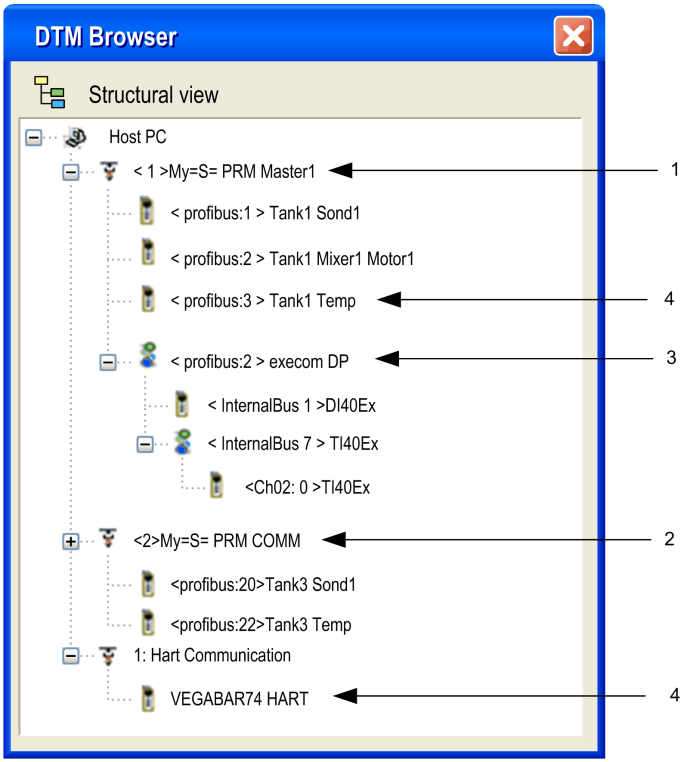
  <1> My =S= PRM COMM, where 1 is the references number.

- For a Device or a Gateway

  <Profibus:1> Tank1Sond1 means that the DTM is connected to the Profibus channel of its “parent” at connection point 1.

Example Connectivity Tree

The following illustration is a connectivity tree:



Number	Node Level Description
1	This Second Level PRM Comm Master communication DTM connects to the HostPC for communication with a PRM device via the Ethernet for DPV0 and DPV1 exchanges.
2	This Second Level PRM Comm communication DTM connects to the HostPC for communication with a PRM device via the Ethernet for DPV1 exchanges.
3	This Third Level Profibus Gateway DTM connects simple or Gateway devices to the higher communication DTM.
4	The Simple or Compact DTMs are connected to communication DTMs that allow them to communicate with their physical devices on the network. These tree leaves do not allow any sub-connectively.

Ethernet Ready Equipment

Definition

An Ethernet Ready equipment is an Ethernet equipment that provides additional services compared to standard EIP or Modbus equipment.

Applicability

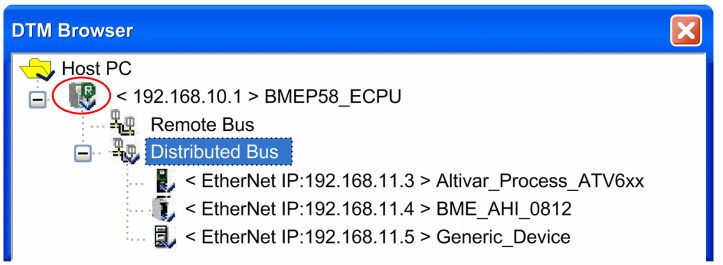
This concept of Ready equipment applies to:

- Standalone devices connected to the Ethernet I/O
- Modules plugged on the Ethernet backplane

NOTE: The only Ready module that exists is the BME CXM 0100.

Representation

In Control Expert, such devices are identified with a specific green label Ⓡ in the DTM browser:



Services

Ready services are related to the software integration of the products into Control Expert. It is based on DTM technology.

This table presents the services available for Ethernet Ready equipment:

Service	Description
Single parameter entry	Any device parameter that you enter is shared into the entire configuration environment.
Bus editor declaration	Declaration is done from the Ethernet I/O bus editor or PLC bus editor in the Control Expert browser. It uses products that belong natively to the Control Expert Hardware Catalog.
System transfer	You can transfer all the system in a single operation: the Ready device configuration is transferred to the PLC with the PLC application. **NOTE:** The PLC acts as a configuration server for all the Ready devices.
Scanning capacity	Scanned in a deterministic way (from the CPU RIO scanner(1)): you can attach the Ready equipment to any PLC task. It is refreshed within the task scan. **NOTE:** This Ready equipment is called remote equipment to differentiate it from distributed equipment, which is scanned from the DIO scanner (RPI can be changed and is not linked to the task period).
Alert message	Inform you when a modification done from the device DTM impacts the PLC application and can lead to stop the PLC.
Online help integration	–
Shared user rights	User rights are automatically shared between Control Expert and the device DTM.
(1) Scanners that accept Ready devices (M580 CPU) are also identified with this green label Ⓡ.	

Ready, Built, Connected and Installed DTMs

At a Glance

The status of a DTM is indicated in the DTM Browser: Ready, Not-Built, Connected and Installed.

Ready

Ready devices or modules are devices or modules that are used by DTMs and fully integrated in the Control Expert software, they mainly bring two particularities:

- They use new Control Expert mechanisms to be integrated into the Control Expert hardware catalog.
- They are managed by Control Expert Ethernet I/O bus editors as for any X80 I/O modules.

A green “R” icon indicates that the device or module is a **Ready** one.

Built or Not-Built DTMs

Build status:

- **Not-Built**: A blue check marks indicated that a node or one of its sub-nodes have changed, the information stored in the physical device is no longer consistent with the local project.
- **Built**: After building the Control Expert project, the blue check mark is no longer in front of browser nodes that were changed before the build.

Connected or Non-Connected DTMs

Connecting a DTM to its physical device automatically connects all higher level parent nodes up to the root node (all DTMs that the DTM being connected depends on.

Disconnecting a DTM from its physical device automatically disconnects all its lower level child nodes (all DTMs that depend on the DTM being disconnected).

Connection status:

- **Connected**: Connected nodes are represented by **bold** text in the browser. A node can be connected without all its sub-nodes being connected.
- **Non-Connected**: Non-connected nodes are represented by **normal** text.

  NOTE: Connecting a fieldbus or device is not linked the Control Expert connection to a PLC (Off-line or On-line). DTMs can be connected/disconnected while Control Expert is either Off-line or On-line.

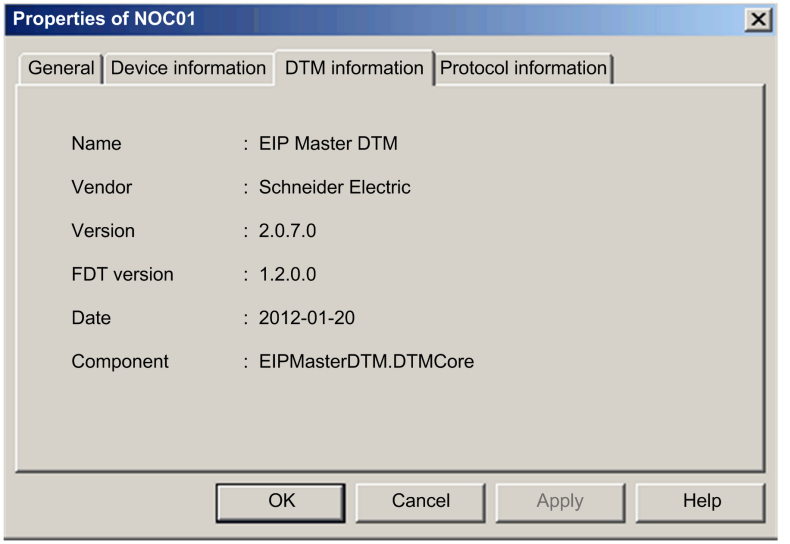
Installed DTMs

Installation status:

A red X on the icon of a DTM indicates that the DTM software is not installed on the PC.

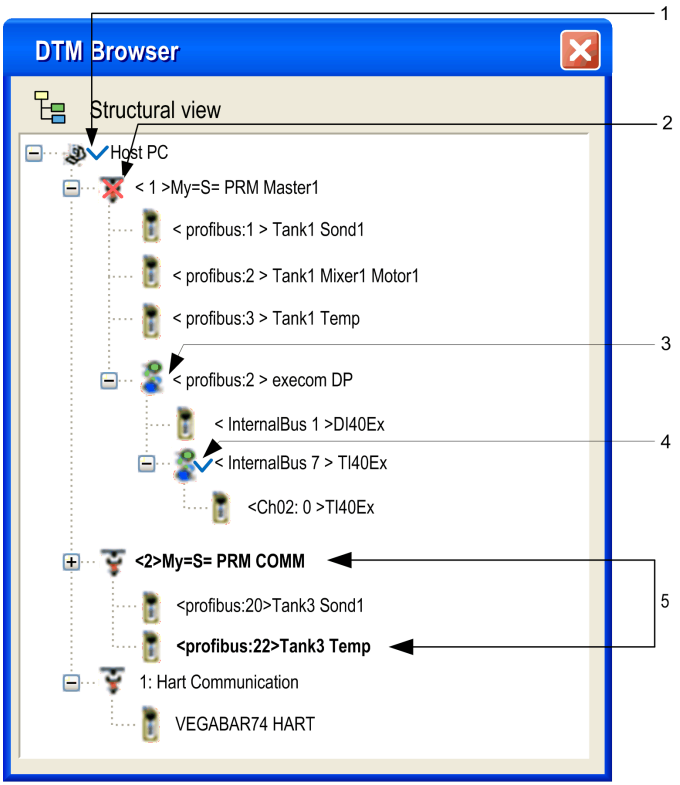
- This status is updated (red X **added**) when devices are detected but no DTM software is installed on the Host PC when the user:
  - Updates the **Hardware Catalog**
  - Tries to build the application
  - Tries to use a service in one of the 2 right-click contextual menus in the DTM Browser
  - Tries to use the Open service by left-clicking twice on a DTM in the DTM Browser
- This status is updated (red X **removed**) when the user updates the **Hardware Catalog** with the missing DTM.

A right-click on an invalid or non-installed DTM opens a 2-item contextual menu:

Function	Description
Delete	This removes the selected DTM (and its sub-nodes) from the DTM project.
Properties	To help identify the non-installed DTMs, this function displays one tab (Device information) in the Properties of dialog box: 

DTM Browser

The following illustration gives examples of built/non-built, connected/non-connected and non-installed statuses:



The following table explains how a DTM Browser indicates if the DTMs are built, connected and non-installed:

Number	Status	Number	Status
1	Not-Built	4	Not-Built
2	Non-installed	5	Connected
3	Built		

DTM Browser Contextual Menus

At a Glance

The (right-click) contextual menu of the DTM Browser offers functions that depend on item clicked on.

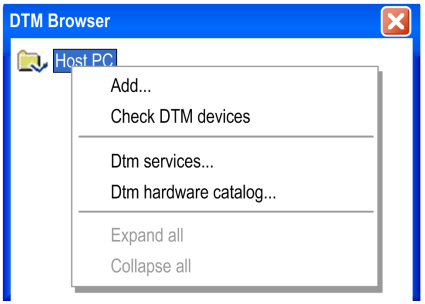
NOTE: The contextual menu items are also available via the **Edit** and **View** menus.

The DTM Browser Contextual Menu accesses:

- Adding and deleting DTMs
- Connecting and disconnecting DTMs to their physical devices
- Displaying and printing the properties of a DTM
- Transferring DTM configuration information to and from the physical device
- DTM-specific functions (via the Device menu function)
- Field Bus Discovery used to scan the physical devices in a fieldbus network

Host PC

The **Host PC** contextual menu:



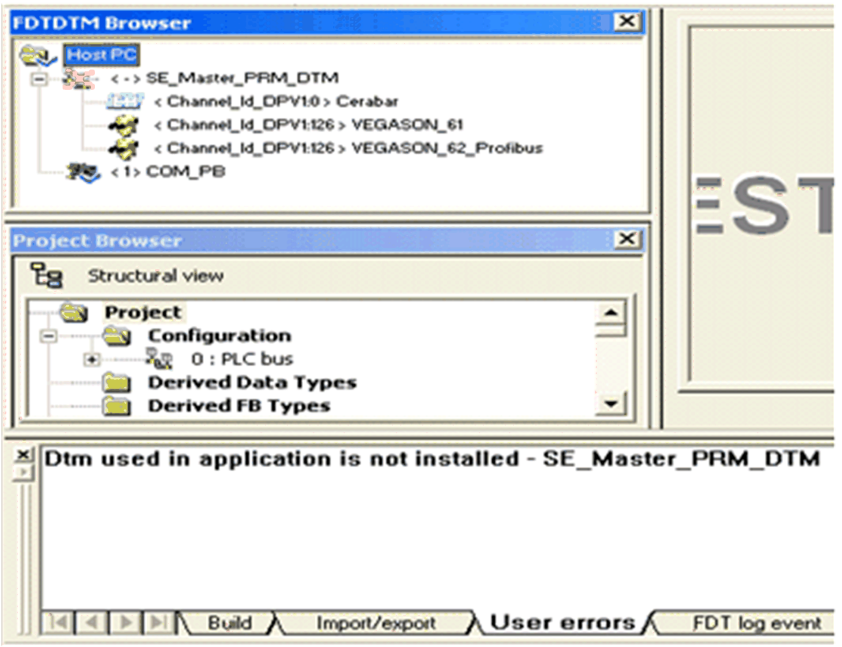
The **HostPC** contextual menu has the following items:

Name	Description
Add...(1)	The **Add** dialog box opens, a version of the **Hardware Catalog**, allowing the selection of a Communication DTM.
Check DTM devices(1)	This function checks the current project for invalid or non-PC-installed DTMs. Refer to Check DTM devices, page 294 below.
DTM Services(1)	This function allows the users to select/unselect all the DTMs added in the project and performed actions on the selected DTMs. Refer to DTM Services, page 295 below.

Name	Description
DTM Hardware Catalog (1)	This function open the **DTM catalog** tab in the **Hardware Catalog**.
Expand all(2)	All DTMs in the project are shown.
Collapse all(2)	Only the Commutation DTMs in the project are shown.
(1) This function is available via the **Edit** menu. **(2)** This function is similar to Project Browser Expand/Collapse all, page 168.	

Check DTM Devices

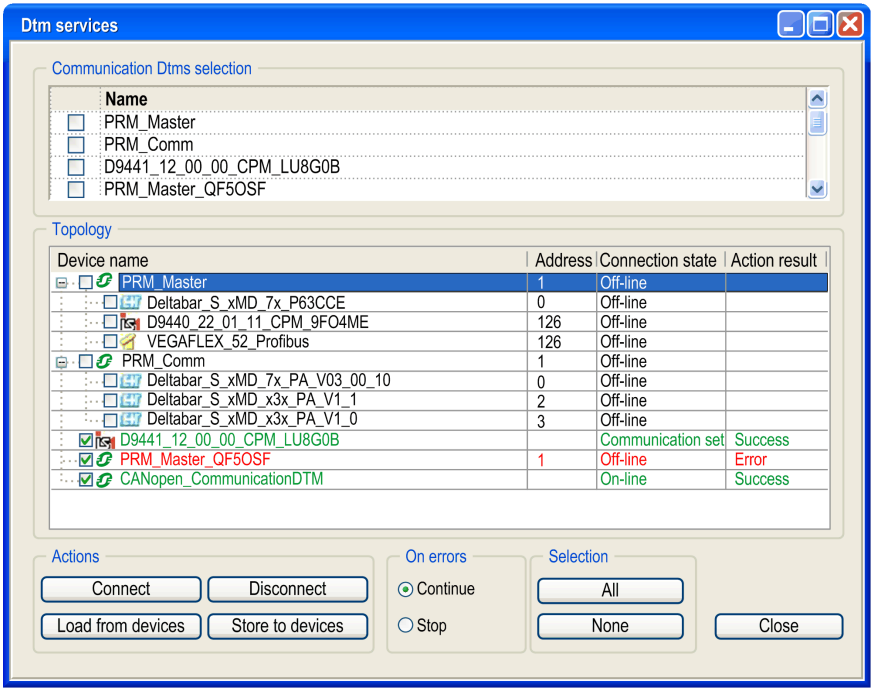
If the results of the check include invalid or non-installed DTMs, they are displayed in the **User errors** tab in the bottom information window and a **Red X** is placed over their icons in the DTM Browser:



NOTE: Check DTM devices only detects installed DTMs, for GSDs verify the GSD library.

DTM Services

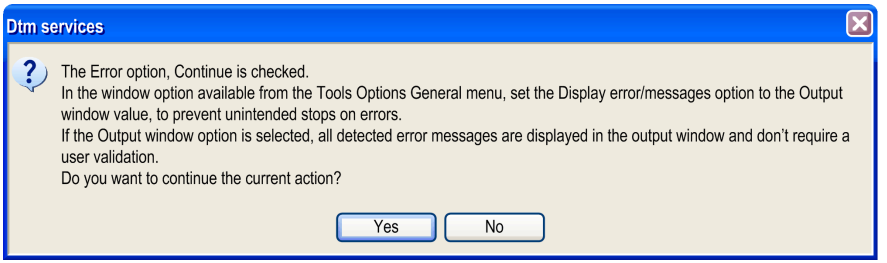
The users can select/unselect DTMs added in the project and performed actions on the selected DTMs:



The **DTMs services** dialog box menu contains the following group boxes:

Name	Description
Communication Dtms selection	The user can select or unselect all communication DTMs and attached DTMs currently added in the project.
Topology	The **Topology** area lists all DTMs added in the project. The set of selected DTMs can be modified by selecting or unselecting the DTMs in the Structure tree control.
Actions	The user can perform actions for each DTMs contained in the set of DTMs previously selected in the Structure tree control.
On errors	The user can select to **Continue** or **Stop** the process if an error is detected. A dialog box is displayed whatever options is selected.
Selection	To select either all DTMs added in the project or none.

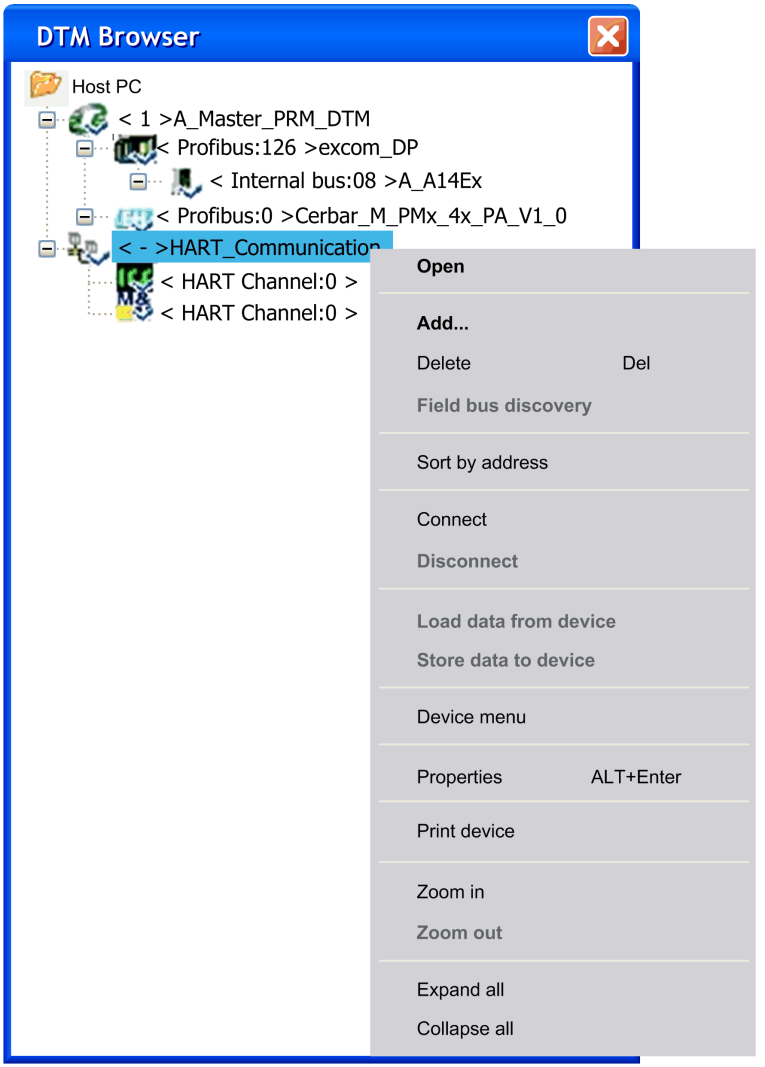
The dialog box below is an example when the user selects either **Continue** or **Stop** in the **On errors** group box:



NOTE: The entire process can be cancelled any time by using control button Cancel.

Communication and Other DTMs

The available DTM Browser contextual menu items differ according to the DTM selected:



The **DTM** contextual menu has the following items:

Name	Description
Open(1)	This opens the “<device name> - DTM Configuration” window with the Configuration and Device Description information supplied by the device vendor. A double-click on the DTM in the DTM Browser also opens this window.
Add...(1)	This opens the **Add** dialog box, a version of the **Hardware Catalog**, allowing the instantiation of a DTM. The DTMs are filtered, that is, the **Add** window only displays DTMs compatible with the DTM selected in the DTM Browser. To add a device refer to Add device, page 300 below
Delete(1)	If the selected DTM allows this function, this deletes the selected DTM and its sub-node DTMs from the DTM connectivity tree. Deletion from the DTM connectivity tree does not effect the DTMs link to the I/O Scanning table. **NOTE:** If the DTM you want to delete is a Gateway, and is not installed on the Host PC: • It is possible to delete the DTM only if no devices are configured under the Gateway. • If devices are configured under the Gateway, it is not possible to delete these devices nor the DTM Gateway.
Field Bus Discovery	This scans the connected physical devices to create the corresponding field bus connection topology. This item is only available in the contextual menu if a device connected. Refer to Field Bus Discovery, page 301 service. **NOTE:** The DTM to scan has to be installed in the PC and the **Hardware Catalog** must be up to date.
Sort by address	This displays the DTMs sorted under their parent DTMs according to the bus address of its physical device in ascending order.
Connect(1)	This connects the DTM to its physical device on the network. This connection does not depend on the PLC online/offline status of the Control Expert project application. Connecting a Gateway- or Device-DTM implicitly connects its parent DTM. While the device is connected, **Connect** is not available.
Disconnect(1)	This disconnects the DTM from its physical device. This disconnection does not depend on the PLC online/offline status of the Control Expert project application. Disconnecting a DTM implicitly disconnects all its child DTMs. While the device is disconnected, **Disconnect** is not available.
Load data from device(1)	This uploads the configuration parameters from the physical device to the local DTM, but not from the application (binary) in the PLC. While the device is disconnected, **Load data from device** is not available.

Name	Description
Store data to device(1)	This downloads the local DTM parameters to the physical device. Each DTM must be stored manually (one by one). While the device is disconnected, **Store data to device** is not available. To make an application in the PLC consistent with the stored DTM, the user must build the Control Expert application and download the new version of the application to the PLC
Device Menu	The functions in this menu are vendor dependent, such as: • Compare (the online and offline parameters) • Diagnostic (displays a diagnostic screen) • Print (HTML) • Observe • Additional functions available when selecting the communication DTM: ◦ EDS file (add, or remove) ◦ EDS library (export, or import) ◦ Online Action ◦ Explicit Message (Ethernet or Modbus) ◦ Store configuration of devices to the FDR server ◦ About ◦ Advanced Mode(4) When configuring a DTM, the changes are validated in Control Expert in 2 ways, depending on the changes: • Automatically as the changes are made. • You must click on a **OK**, **Apply** or **Validate** button after making the changes.
Properties	Refer to the DTM Properties Tabs, page 309 section.
Print device	If this optional function is supported by a DTM, it opens the vendor device documentation in Internet Explorer, which can then be printed. Only one DTM information can be printed out at time. A print of the whole network is not available.
Zoom in(1),(2)	This displays only the selected DTM node and its child DTMs. While zoomed-in, **Zoom in** is not available.
Zoom out(1),(2)	This returns to the full display of the DTM connectivity tree. While there is a full display, **Zoom out** is not available.
Expand all(3)	This displays all DTMs below the selected DTM.

Name	Description
Collapse all(3)	This displays only the selected DTM.
(1) This function is available via the **Edit** menu.	
(2) This function is similar to Project Browser Zoom in/out, page 166.	
(3) This function is similar to Project Browser Expand/Collapse all, page 168.	
(4) Advanced option are available DTM configuration screen	

Add a DTM Device

To add a new DTM to the connectivity tree in the DTM Browser:

Step	Action
1	Make sure that the DTM appears in the **Hardware Catalog**. If it is not in the catalog, perform an update, page 274 of the **Hardware Catalog** using the **Update** button. If it still does not appear in the **Hardware Catalog**, the DTM must be installed on the PC, page 272.
2	In the DTM Browser select the node where the DTM is to be added.
3	Right-click and select **Add...**
4	In the **Add** window select the DTM to be added to the DTM connectivity tree.
5	Either: • Click on the **Add DTM** button • Double-click on the selected DTM Results: The 4-tab DTM **Properties of** dialog opens.
6	Using the 4 tabs, verify that the correct DTM has been added to the DTM connectivity tree and verify its configuration.
7	If the DTM Alias name does not meet the Control Expert (or the user application) naming rules, it must be changed before closing the dialog.
8	Click on the **OK** button to finish adding the DTM to the network tree or click on the **Cancel** button to close the dialog without adding anything.

Field Bus Discovery Service

At a Glance

The Field Bus Discovery service is available for Communication and Gateway DTMs that:

- Support device scanning
- Are connected to their physical devices

Only the first level devices below the Communication and Gateway DTMs are detected (there is no recursive scanning).

Field Bus Discovery Use

The results of the scanning process is compared to the registered DTMs in the DTM catalog of the computer. If a match is found in the DTM catalog for a scanned device, the results are accompanied with a matching type that gives the accuracy of the match.

The three available matching types are:

- Exact Match:

  All identification attributes are matching. The correct device type was found.

- Generic Match:

  At least the Manufacturer ID and Device Type ID attributes match. The support level of DTM is “Generic Support”.

- Uncertain Match:

  At least the Manufacturer ID and Device Type ID attributes match. The support level of DTM is **not** “Generic Support”.

The following procedure explains how to use the Field Bus Discovery service:

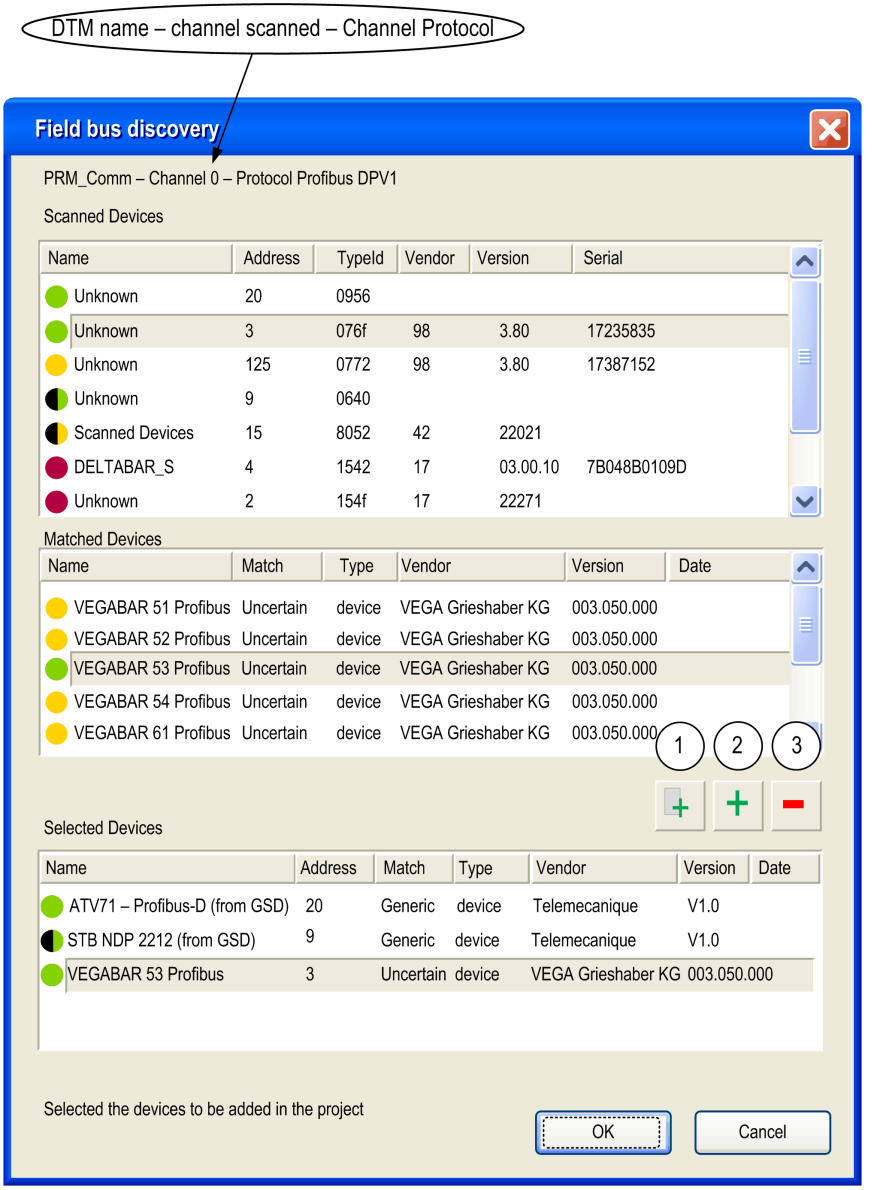
Step	Action
1	In the DTM Browser select an appropriate DTM.
2	Right click to open the Contextual menu **Field Bus Discovery**.
3	If necessary, select the channel and protocol: • If the DTM has more than one channel • If the channel supports more than one protocol
4	Click on **OK** in the message box to have this service detect all devices on the selected channel.
5	If at least one matched device has been found, the Field Bus Discovery dialog, page 303 is displayed listing the scanned and matched devices. Select the matched devices to be created in the Control Expert project.

NOTE: a device that is connected on the bus can be detected if:

- Its DTM is installed on the PC.
- Its GSD is added into the GSD library (if not a DTM device).
- The DTM hardware catalog is up-to-date (with the DTM installed and the GSD in the library).

Field Bus Discovery Dialog

If at least one matched device has been found, the Field Bus Discovery dialog box is displayed listing the scanned and matched devices. Select the matched devices to be created in the Control Expert project (which then shows up in the **Selected Devices** list:



This dialog has three lists:

List	Description
Scanned Devices	This list displays all the devices (matched and unmatched) found during the scan.
Matched Devices	This list displays the matched DTM found in the workstation DTM catalogue for the selected device in the **Scanned Devices** list. Each time a scanned device is selected in the **Scanned Devices** list, the contents of the **Matched Devices** list is updated to display the matched device DTM found for the selected scanned device. The matching process can yield one or more matched devices for a given scanned device. In this case, select one among the matched DTMs found.
Selected Devices	This list displays the device DTMs selected to be added to the Control Expert project.

The lists use colored icons:

Color	Description
Green	Indicates that the device has been selected
Yellow	Indicates that the device has been matched
Red	Indicates that the device has **not** been matched
Black	Indicates information about the address of the scanned device: • In the **Scanned Devices** list, the device has an address identical to one of the DTMs in the Control Expert project • In the **Matched Devices** list, the device will be assigned an address identical to one of the DTMs in the Control Expert project

This dialog has five buttons:

Button	Use this button to...
1	automatically add the best matched (according to the matching types listed above) device DTM for each found device in the **Matched Devices** list to the **Selected Devices** list.
2	add the matched device DTM selected in the **Matched Devices** list.
3	remove one or more devices from the **Selected Devices** list.
OK	insert the device DTMs in the **Selected Devices** list into the Control Expert project. If there are one or more devices in the **Selected Devices** list that have the same address in the Control Expert project, a message box opens asking if you want to continue. If you click **OK**, all devices in the Control Expert project that have identical addresses as the selected devices is **deleted** and **replaced** by the DTMs selected in the **Selected Devices** list.
Cancel	cancel the Field Bus Discovery scan and do nothing. All information in the three lists is lost

Store Device Configurations

Introduction

After configuring or modifying a configuration of a DTM, the changes must be downloaded to its physical device using Store data to device, page 293.

Connectivity Tree

For a connectivity tree it is necessary to:

- Store the configuration for each DTM into the corresponding device so that the device works correctly
- **Build** or **Rebuild all** the EcoStruxure Control Expert application and download it to the controller be able to upload the complete application including all DTM configurations from the controller binary

The total number of operations equals the number of DTMs stored + 1 download of rebuilt application.

Recovering Connectivity from the Controller Application

If the complete application has been stored in the controller (ETS option checked), it includes the connectivity of the DTMs.

Uploading the application from the controller provides the connectivity stored in the controller. For this, the PC used for uploading must have the corresponding DTMs, page 286 installed.

Verify that the corresponding configuration is stored in the devices to avoid de-synchronization of the project..

De-synchronization of Project

If a DTM configuration is changed and stored to its device, the application running in the controller is not updated.

To synchronize the controller application with the device configuration, rebuild the EcoStruxure Control Expert application to take into account the changes and download it to the controller.

Uploading a binary from a controller overwrites the configurations of all DTMs in EcoStruxure Control Expert applications retrieved from the controller. Any modifications of the DTM configurations that are not stored in the controller binary are overwritten by the configuration in the controller binary and lost.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

Do not upload a project from the controller if any changes have been made in any devices connected to the controller since the last build and download of the project.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Renaming DTMs

At a Glance

According to the FDT standard, the official name of a DTM is its Tag name assigned by the DTM vendor.

Unfortunately, some DTMs have Tag names that do not:

- Allow the Tag name to be changed
- Meet the Control Expert naming rules

The solution is to choose the DTM Alias name that meets these rules (and any special user defined application naming rules).

Alias Name

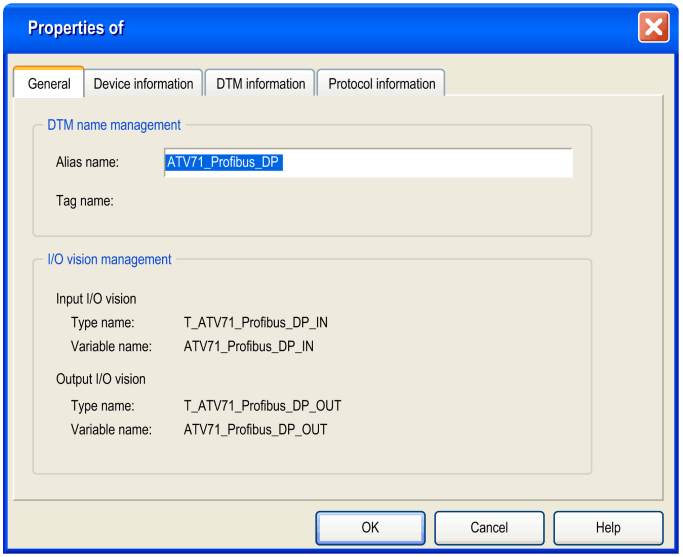
The Alias name is the DTM reference name in Control Expert and must be unique in an application.

If possible, the default, the Alias name is the same as the DTM Tag name. But if necessary, the Alias name can be changed in 3 places:

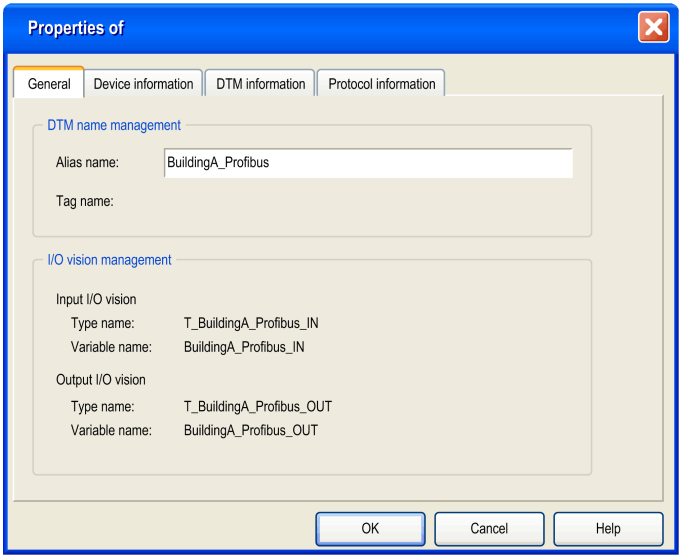
- Directly in the General tab, page 310
- In the DTM Browser by clicking on the selected DTM or using the F2 shortcut with the selected DTM.

I/O Vision in Communication DTMs

For device DTMs attached to certain Schneider Electric Communication DTMs (such as the SE_Master_PRM_DTM), the default names of the DTM Input/Output Types (a structure) and the Variable instances of the Type are based on the default DTM Alias/Tag name:



Changing the Alias name automatically changes the names of the Input/Output Types and Variables:



Profibus DTMs

Configuration of Profibus Field Devices

Using vendor-supplied Profibus DTMs, users are able to remotely:

- Configure Profibus devices associated with the DTM
- Adjust the parameters of these devices while they are **off-line**
- Adjust the parameters of these devices while they are **on-line**

PRM Master Bus Module and its DTMs

Control Expert uses a Profibus Remote Master (PRM) Gateway module (the bus master physical device) and its PRM configuration tools (software):

- The PRM **Master** DTM that allows the PRM Gateway to carry out:
  - Explicit DPV0 functions (control and diagnostics of the Profibus slaves on the network).
  - Communication functions (implicit DPV1 exchanges with the Profibus slaves). It also provides access to the Profibus slaves from the Control network to HMI, SCADA or ASSET management tools.
- The PRM **Comm** DTM that only allows the PRM Gateway module to carry out communication of implicit DPV1 exchanges and access to management tools.

The Modbus TCP I/O scanning link between the PLC and the (PRM) Gateway module connected to the Profibus network is provided by a Quantum, Premium, M340, or M580 module with an Ethernet port.

DTM Properties Tabs

About this Section

This section explains the different versions of the **Properties of** dialogs.

DTM Properties Dialog

At a Glance

The number of tabs in a **Properties** dialog varies according where the dialog is called from.

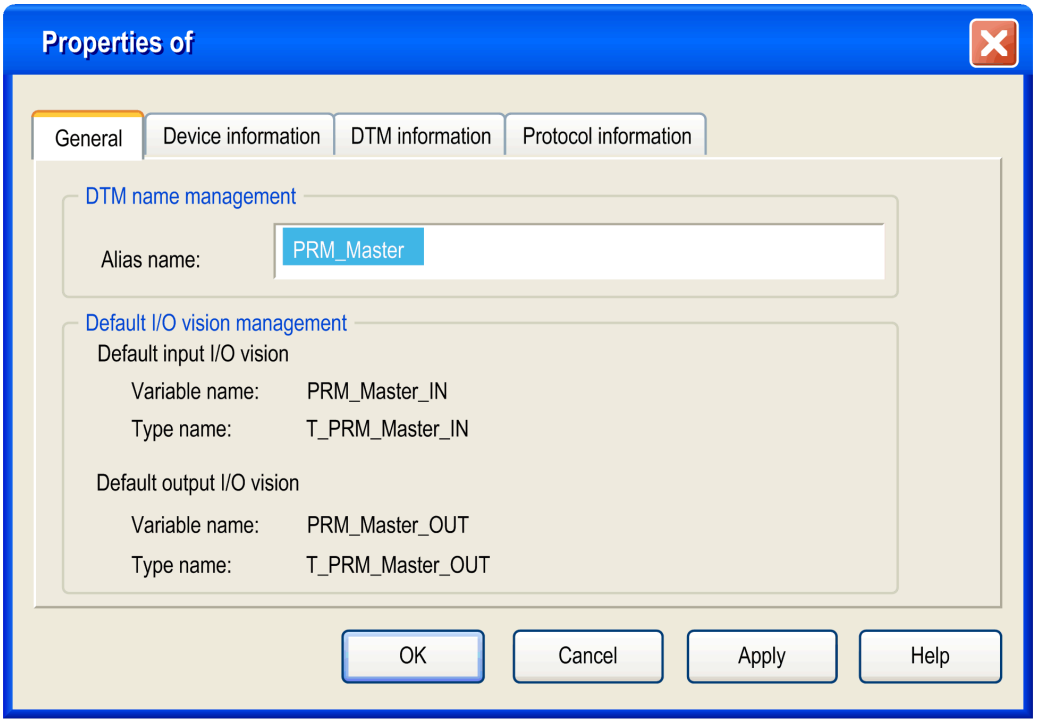
Properties Tabs

The **Properties** dialog is called when a DTM is selected:

- To be added to connectivity tree in the DTM Browser from the **Hardware Catalog**:
- For the Contextual Menu **Properties** service in the **Hardware Catalog**:
- It opens with 4 tabs:
  - General
  - Device Information
  - DTM Information
  - Protocol Information

General

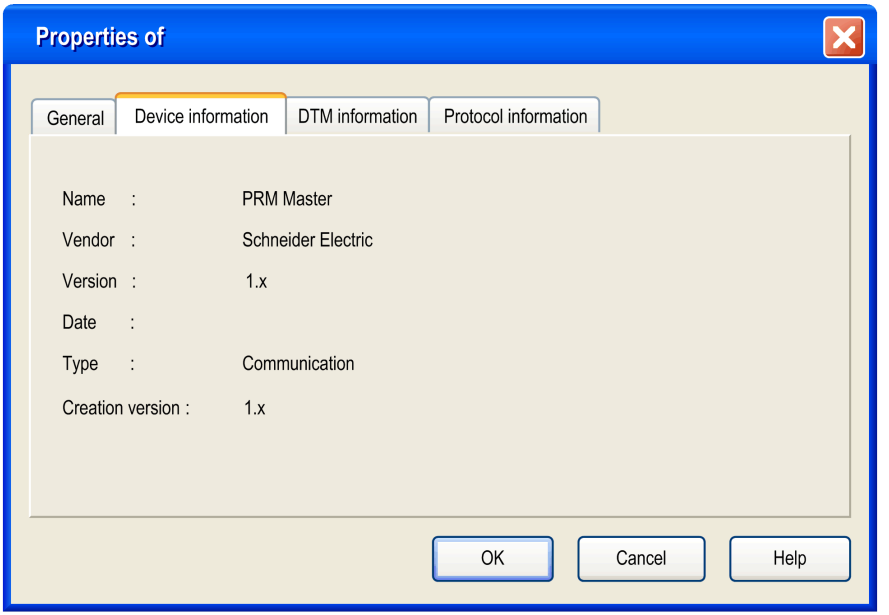
The General tab:



NOTE: The Alias name (and the DTM Tag name if the vendor allows it) can be changed, page 306 here or in the DTM Browser.

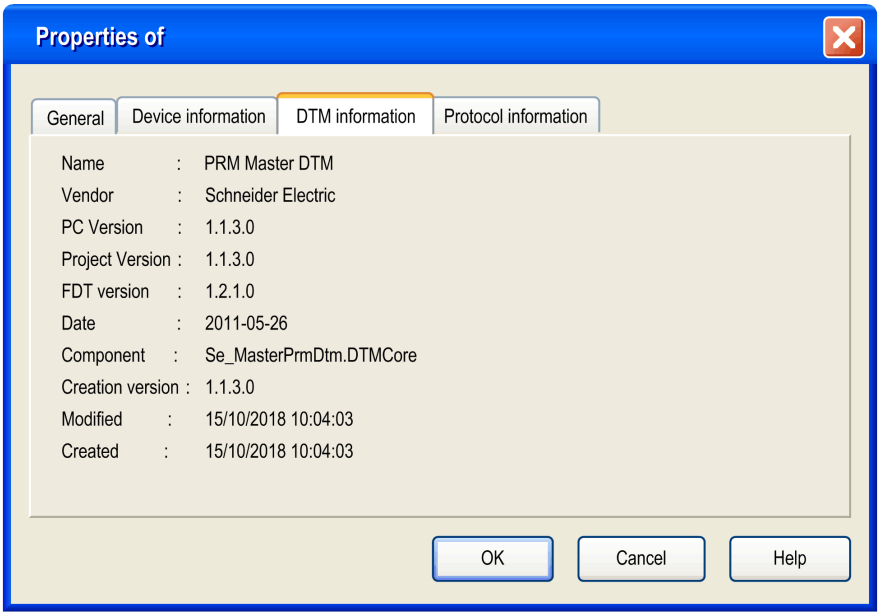
Device Information

The Device Information tab:



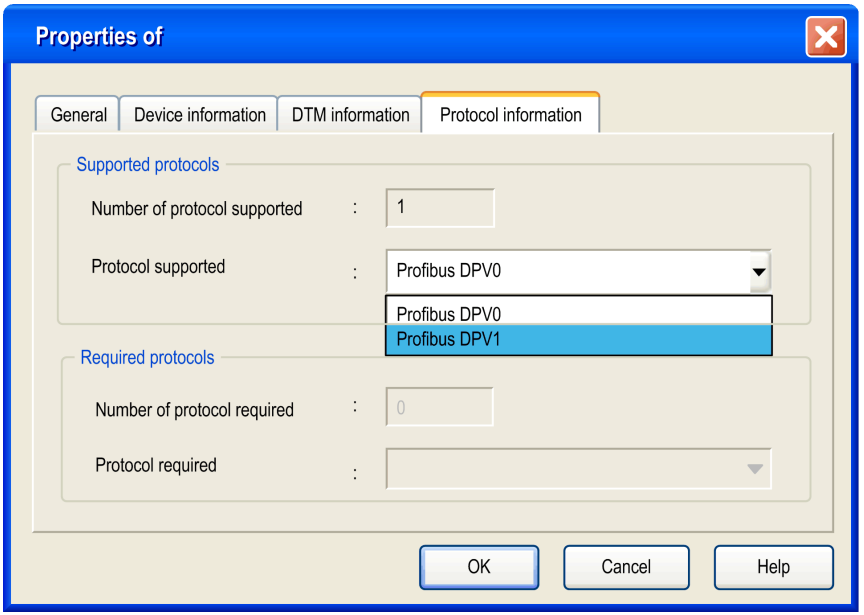
DTM Information

The DTM Information tab:



Protocol information

The Protocol Information tab:



Item	Description
Supported protocols	These are the protocols available for the child nodes of this DTM.
Required protocols	These are the protocols that must be supported by the parent DTM of this DTM.

Type Library Manager

What's in This Chapter

Libset (Set of Libraries) 314
Overview of the Type Library Manager..... 317
Type Library Manager 318
Description of the Type Library Manager..... 321
Loading an object from the library into the project 327
Transferring an object from the project into the library 330
Creating a new library and family..... 332
Creating an installable family 334
Updating a family of a library..... 336
Consistency Check Assistant..... 338
Library Version Management 341
Using the type library of an older Unity Pro/Control Expert Version 345

Object of this section

This chapter describes the main functions of the Type Library Manager.

Libset (Set of Libraries)

At a Glance

Libset (set of libraries) comprises all libraries, families, functions and variables (I/O data structures) that can be used to develop an automation project.

You can only modify the **Libset** version in the **Open** project dialog.

Libset Versions

As of Libset version 10.0, the library set is automatically installed with Control Expert.

To install previous **Libset** versions, start **Programs > EcoStruxure Control Expert > Install libset** tool.

Creating a New Project

Creating a new project, the latest **Libset** version is automatically selected for this project.

To modify the **Libset** version of your project, you have to save your project and open it again with the version of **Libset** you want to use.

Opening a Project

In general, opening an existing project (STU file) or an archived project (STA file), you can select the **Libset** version.

NOTE: At opening time, if the **Libset** version is missing, a pop-up message appears. You can use the **Installer libset** tool to setup one or several Libsets.

Importing a ZEF or XEF file, the latest **Libset** version is always selected.

The following table describes the default **Libset** version used to open a project according to the file type.

File Type	Project before Unity Pro 4.0	Project from Unity Pro 4.0 to Unity Pro 5.1	Project from Unity Pro 6.0
STU file	Not applicable	The **Libset** version used to build the project is automatically selected. If the respective **Libset** version is not installed, the latest version will be used.	
STA file	The **Libset** version used to generate the STA file is automatically selected. If the respective **Libset** version is not installed, the latest version will be used.	The **Libset** version used to generate the STA file is automatically selected. If the respective **Libset** version is not installed, the latest version will be used.	
XEF file	The latest **Libset** version is selected.	The latest **Libset** version is selected.	
ZEF file	Not applicable	Not applicable	The latest **Libset** version is selected.

Modifying the Libset Version

Opening an existing project, page 77 (STU file) or an archived project (STA file) you can select the **Libset** version.

The **Open** dialog provides the following options.

Element	Description
Keep project version	Select this option to keep the **Libset** version used to build the project. If the respective **Libset** is not installed, the latest **Libset** version will be used.
Update with latest version available	Select this option to update the project with the latest **Libset** version.
Select a specific target version	Select this option to choose a specific **Libset** version. You can select the version by means of the **Version** box below.

If the **Libset** version is modified, the project is in modified state.

Upgrade/Downgrade

- Upgrading the **Libset** does not update the project with the newly chosen libset automatically.

  EF/EFB differences will be displayed during opening the project in the **LIBRARY Version Management** dialog, where you can update the project.

  You can update the project later as well, using the contextual menu command in the **Type Library Manager** dialog, page 341.

  If you have updated the project, you have to start a **Rebuild All**. A **Rebuild All** needs to stop the PLC and to perform a global download.
- You can also downgrade the **Libset** version.

  There is no consistency check when downgrading the **Libset** while opening a project.

  So you may not be able to rebuild the project (using a block created in an earlier **Libset** version, not available in the latest **Libset** version).

  If the rebuild is unsuccessful, you have to save your project and to open it again with an appropriate **Libset**.
- If you want to add an FFB which is not yet contained in the current version of the **Libset**, you have to upgrade the whole **Libset**.
- A change of **Libset** version in the **Open** dialog is memorized in the project.

Libset Information

If no project is opened, information is displayed on the latest **Libset**.

If a project is opened, information is displayed on the **Libset** used for the project.

Information on the **Libset** version used for your current project is provided in the **General** tab of the **Properties of Project** dialog.

In the **Project Browser** of Control Expert right-click **Project** and select **Properties** from the context menu. In the **General** tab you find the **FFB Library version**.

Further on the following dialogs provide information on the **Libset** version (e.g. LibsetV3.0):

- **Types Library Manager**
- **Types Library Browser**
- **Data Selection**
- **Data Editor: Variable Type Selection**
- **Function Input Assistant: FFB Type Selection**
- **Data Properties**
- **PLCScreen > Information tab > APPLICATION > IDENTIFICATION**

Types Library Manager

In the **Types Library Manager** click the **Information** button to get the following information:

- **Current library**
  - **Library status**
  - **Diagnostic**
- **Library contents**
  - **Families**
  - **Libraries**
  - **FFBs**

For the **Libset Information** window please refer to information, page 320.

Overview of the Type Library Manager

Principle

The Control Expert Library contains all available objects to develop an automation project. These objects may be the following functions or variables:

- EF (Functions),
- EFB (Function blocks),
- DFB (user function blocks),
- DDT (Variables)
- ...

The Type Library Manager provides functions to edit the contents of the library and execute transfers between the library and the project you are working on.

Simple operations are automatically executed or you will be asked to confirm. An Assistant, page 338 will support complex operations. This Assistant becomes available during loading, transfers, or deletions.

List of available functions

The Control Expert context menus provide access to the library management functions below:

- Creating a family or library
- Deleting a family or library
- Cutting a family or object
- Inserting a family or object
- Showing and hiding a tree structure for an object, family, or library
- Renaming the current entry in the tree structure
- Displaying the Assistant to load an object
- Fit columns
- Deleting types not used in the project
- Version check: Consistency between different versions of the objects of the library and project Library Version Management, page 341.

Type Library Manager

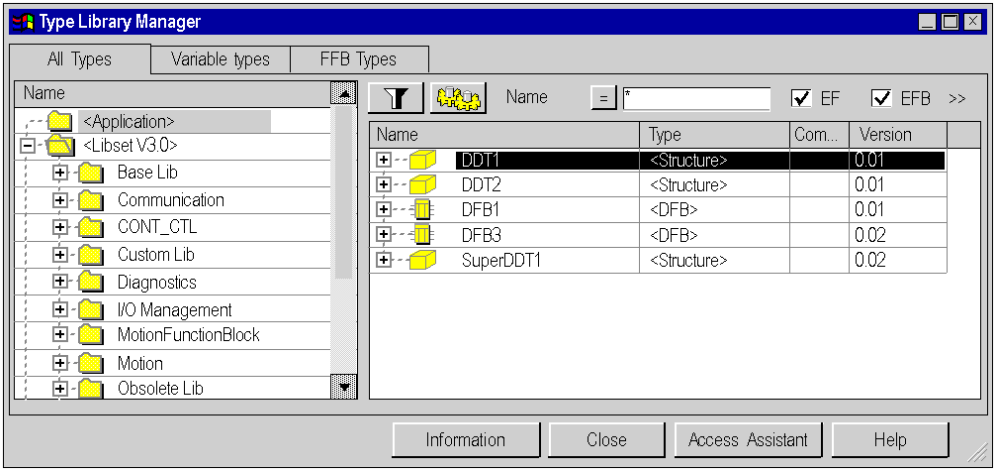
At a Glance

Control Expert software provides a manager to edit objects in the library providing access to all managing functions of the library.

- Loading an object into the project from the library, page 327.

- Transferring an object of the project into the library, page 330.
- Deleting a library object
- Updating and managing library versions
- ...

Accessing the Manager

Step	Action
1	In the **Tools** menu, select the **Type Library Manager** option. **Result**: The following window appears. 

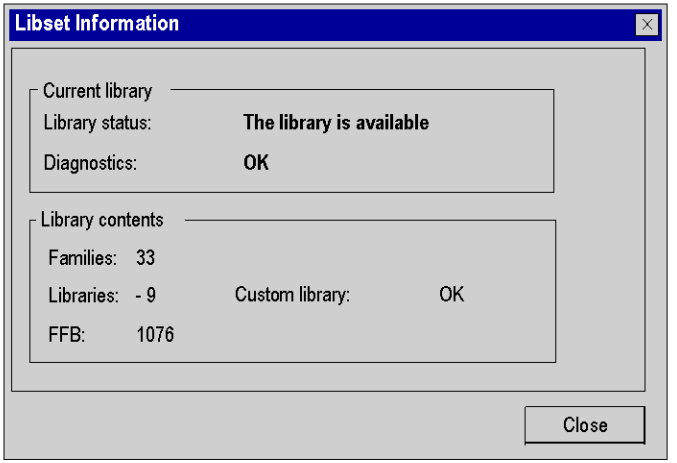
Areas of the Type Library Manager

Area	Description
Tabs	The tabs enable you to display the following according to your selection: • All Types • Variable types • FFB Types (Functions)
The left sub-window	His sub-window shows the tree structures of the objects of the project, library of functions (division of functions by family), and the IODDT.

Area	Description
The right sub-window	This sub-window displays the list of objects included in the selection made in the left sub-window. The check boxes above it are used to define this list.
Buttons	• Information Click this button to get information on the current library. See paragraph below. • Close Click this button to close the dialog box without accepting changes. • Access Assistant Click this button to open a window to copy objects into a project (Loading an object from the library into the project , page 327). • Help Click this button to call up the dialog help.

Information

Click the **Information** button to open the **Libset Information** window.



The table below describes the various functions offered.

Fields	Description
Current library	**Library status**: information on the status of the custom library • available • not available **Diagnostics**: information on the validity of the library • OK (valid) • not valid.
Library contents	**Families:** indicates the number of families available **Libraries:** indicates the number of libraries available **FFBs:** indicates the number of function blocks available **Custom Library:** indicates whether or not user libraries exist

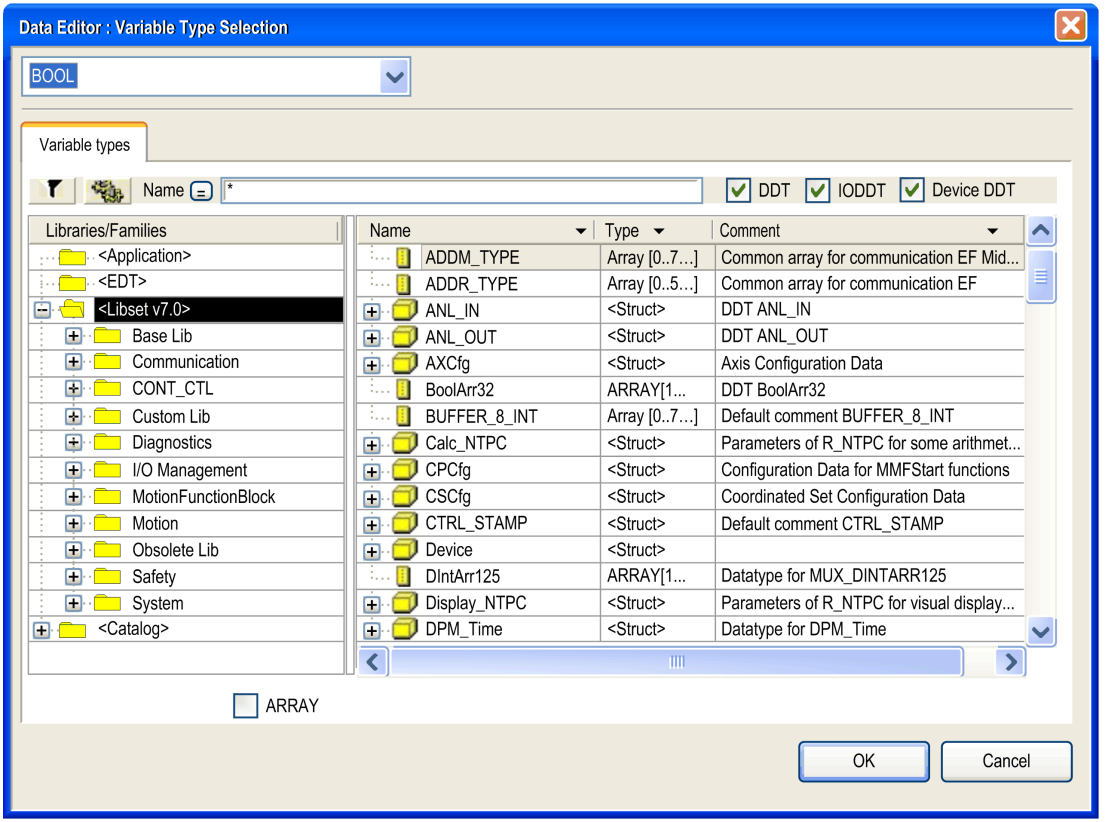
Description of the Type Library Manager

Dialog structure

The dialog box to select objects has three tabs:

- **All Types**
  Under this tab, you can select variables and FFBs.
- **Variable types**
  Under this tab, you can select the variables.
- **FFB Types**
  Under this tab, you can select FFBs.

Tab All Types tab



Elements of the **All Types** tab:

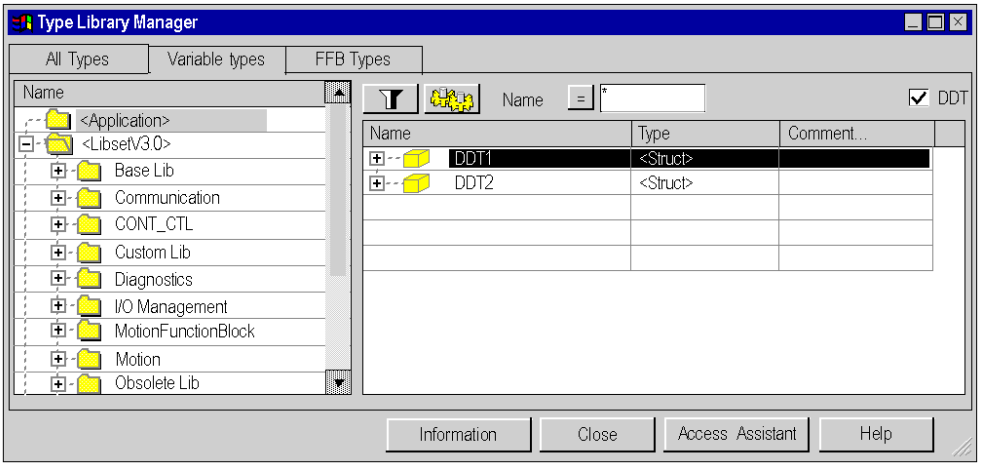
Element	Description
Button 	Click this button to update the display corresponding to the filter term defined in the **Name** field.
Button 	Click this button to open a dialog box, page 433 for defining the filters.

Element	Description
Button 	Click this button to invert the filter. The button changes from **=** to **<>** and vice versa.
Name (Text Box)	The name of the object to search for can be entered in this text box. You can also use the joker ***** and **?**.
EF	When you enable this check box, Elementary Function types (EFs) are displayed.
EFB	When you enable this checkbox, Elementary Function Block types (EFBs) are displayed.
DFB	When you enable this checkbox, Derived Function Blocks (DFBs) are displayed.
DDT	When you enable this checkbox, multi-element variables (derived variables) are displayed.
Device DDT	When you enable this checkbox, multi-element variables (derived variables) are displayed.
Name (Column)	This column displays the names of the function types and function block types and their formal parameters (inputs, outputs).
Type	This column displays the types of the function types and function block types and the data types of their formal parameters (inputs, outputs).
Comment	This column displays comments about the function types and function block types and their formal parameters (inputs, outputs).
For information on the buttons at the bottom of the dialog please refer to Areas of the Type Library Manager, page 319.	

NOTE: Help on Type:

Hold down the SHIFT+F1 keys and then click on the function block type for which you wish to access Help.

Variable Types tab

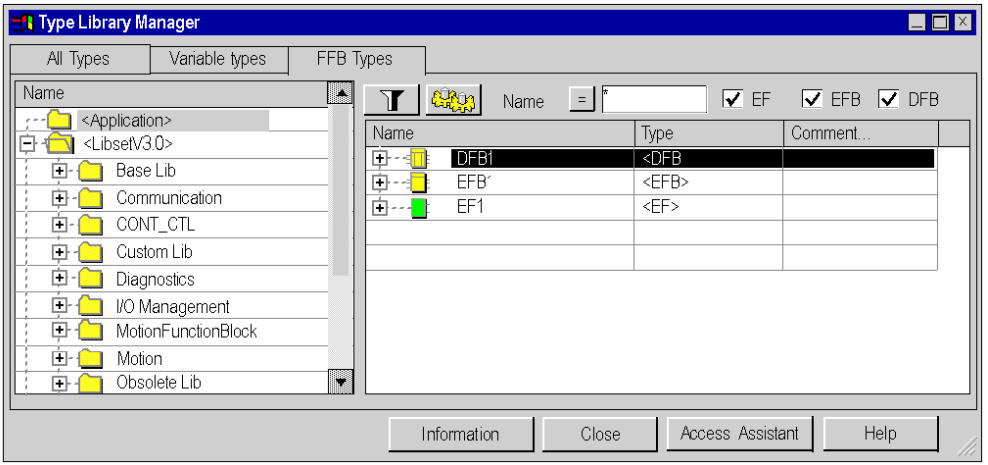


Elements of the **Variable Types** tab:

Element	Description
Button 	Click this button to update the display corresponding to the filter term defined in the **Name** field.
Button 	Click this button to open a dialog box, page 433 for defining the filters.
Button 	Click this button to invert the filter. The button changes from **=** to **<>** and vice versa.
Name (Text Box))	The name of the object to search for can be entered in this text box. You can also use the joker ***** and **?**.
DDT	When you enable this checkbox, multi-element variables (derived variables) are displayed.
Device DDT	When you enable this checkbox, multi-element variables (derived variables) are displayed.
Name (Column)	This column displays the names of the function types and function block types and their formal parameters (inputs, outputs).

Element	Description
Type	This column displays the types of the function types and function block types and the data types of their formal parameters (inputs, outputs).
Comment	This column displays comments about the function types and function block types and their formal parameters (inputs, outputs).
For information on the buttons at the bottom of the dialog please refer to Areas of the Type Library Manager, page 319.	

Tab FFB Types



Elements of the **FFB Types** tab:

Element	Description
Button 	Click this button to update the display corresponding to the filter term defined in the **Name** field.
Button 	Click this button to open a dialog box, page 433 for defining the filters.

Element	Description
Button 	Click this button to invert the filter. The button changes from **=** to **<>** and vice versa.
Name (Text Box)	The name of the object to search for can be entered in this text box. You can also use the joker ***** and **?**.
EF	When you enable this check box, Elementary Function types (EFs) are displayed.
EFB	When you enable this checkbox, Elementary Function Block types (EFBs) are displayed.
DFB	When you enable this checkbox, Derived Function Blocks (DFBs) are displayed.
Name (Column)	This column displays the names of the function types and function block types and their formal parameters (inputs, outputs).
Type	This column displays the types of the function types and function block types and the data types of their formal parameters (inputs, outputs).
Comment	This column displays comments about the function types and function block types and their formal parameters (inputs, outputs).
For information on the buttons at the bottom of the dialog please refer to Areas of the Type Library Manager, page 319.	

NOTE: Help on Type:

Hold down the SHIFT+F1 keys and then click on the function block type for which you wish to access Help.

Shortcut menu structure

Additional commands can be executed via the shortcut menu of the Type Library Manager.

Shortcut menu commands:

Menu command	Description
Delete	Use this menu command to delete a selected user-defined library.
Expand All	Use this menu command to show the directory structure of a selected library. The same function is also available via the **+** symbol.
Collapse	Use this menu command to hide the directory structure of a selected library. The same function is also available via the **-** symbol.
Create New Library	Use this menu command to create a new user-defined library. Compare also Creating a new library and family, page 332.

Menu command	Description
Rename	Use this menu command to rename a user-defined library.
Local Print	Use this menu command to print a list of the FFBs of a selected library.
Display Get Assistant	Use this menu command to open a window to copy objects into a project. (Loading an object from the library into the project , page 327).
Customize Columns...	Using this menu command you can: • define the columns, page 1235 to be displayed and • define the sequence, page 1235 for the columns shown.
Purge unused Types	Use the menu command to delete all unused FFBs from the project. **NOTE:** When a FFB is not available for a kind of PLC (after a section import for instance) but still in the application (removed from the section but still in the list of the Application FFBs), the purge feature should be used before building the application.
Compare Project with Library	This menu command enables you to compare all FFB types of a project with the FFB types of the libraries and update them, if necessary. Compare to Library Version Management, page 341

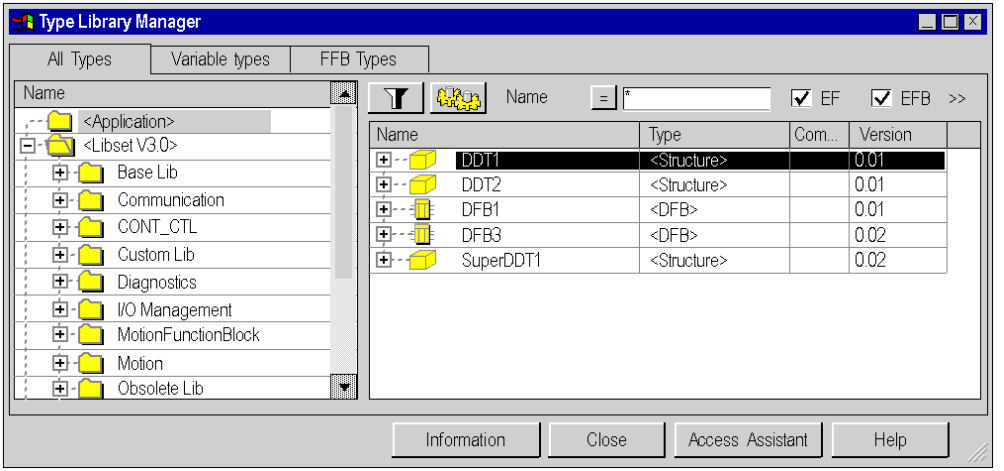
Loading an object from the library into the project

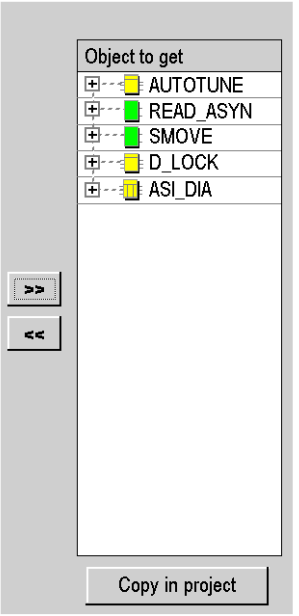
At a Glance

Loading objects from the library into the projects makes the objects used in a project available quicker without having to run the list of all other objects in the library.

Loading is automatically executed during the first use of a Logic section. You can also load the objects manually from the Type Library Manager.

Loading an object into the project

Step	Action
1	In the **Tools** menu, select the **Type Library Manager** option. **Result**: The following window appears. 
2	Click on **Access Assistant** to open the Help window for loading.

Step	Action
	Result: The following area is added on the right of the previously displayed window: 
3	Use this area to create a list of objects to be loaded by selecting them in sequence beginning in the two sub-windows of Step 1 and then transfer the objects by clicking on the button (double arrows).
4	Click on **Copy to project** to execute the loading process. **Result**: The objects are now loaded into the project and will be displayed in the tree structure for the project.

Loading requirements

If the name of the object in the project ...	then...
does not exist,	the object will be transferred automatically.
exists, but the types are different.	**the transfer will be stopped and not performed, and a message listing the causes will be displayed.** **Note**: To load this object you must rename it and restart the loading process.
exists and the types are identical.	loading will be executed after you confirm. **Note**: The Consistency Check Assistant, page 338 will ask you to confirm before loading.

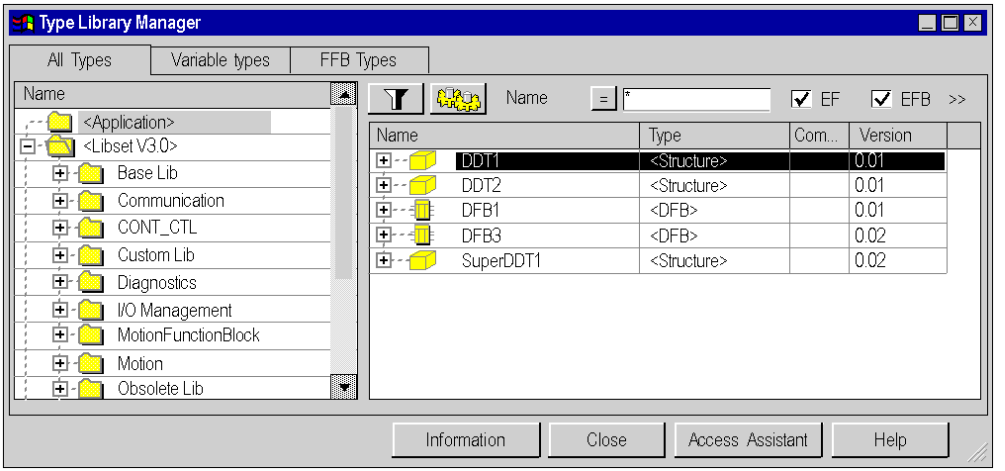
Transferring an object from the project into the library

At a Glance

The transfer of an object from the project into the library enables the use of the objects of a project with another project without having to import or export between applications.

Objects transferred to the library can be accessed from all projects.

Transferring an object to the library

Step	Action
1	In the **Tools** menu, select the **Type Library Manager** option. **Result**: The following window appears. 
2	Select the object of the project to be transferred.
3	Click the right mouse button to open the context menu and select the **Copy to Library** option.
4	Select the library and target family and click on **OK** to confirm your selection. **Result**: The transfer will be performed automatically. **Note**: If the transfer is not performed automatically, a help window will be displayed to help you correct the problem.

Transfer requirements from

If the name of the object in the library ...	then...
does not exist,	the object will be transferred if you have specified the family and the library to which the object is to be transferred.
exists, but the types are different.	**the transfer will be stopped and not performed, and a message listing the causes will be displayed.** **Note**: To load this object you must rename it and restart the loading process.

If the name of the object in the library ...	then...
exists and the objects are identical.	no action is required because the objects are identical.
exists and only the codes of the objects are different.	the transfer will only be executed if the version of the object to be transferred matches or is higher than the version of the object that currently exists in the library. **Note**: The Consistency Check Assistant, page 338 will ask for a confirmation before executing the transfer.
exists and only the prototypes of the objects are different.	**the transfer will be stopped and not performed, and a message listing the causes will be displayed.** **Note**: To load this object you must rename it and restart the loading process.

Creating a new library and family

At a Glance

You can create a user-defined library. It contains objects that are available for the development of automation projects, such as:

- Derived Data Types (DDT),
- Function Block Types (EFB/DFB),
- Elementary Functions (EF).

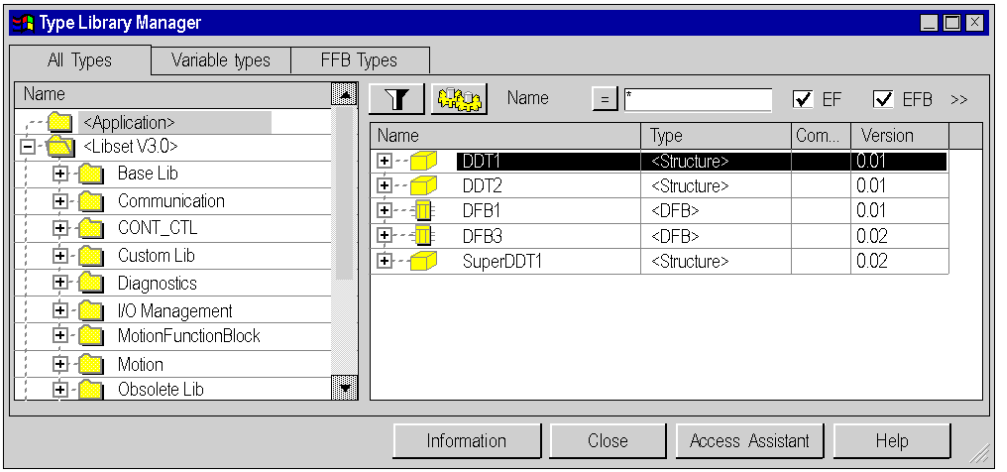
The library is linked to a programming device. This library can access all projects created with this programming device. You can save the Family of the Library, page 334 to reuse, page 336 it on another programming device.

The following steps are required:

- Creating a library,
- Creating one or more families in the new library,
- Linking objects (EF, EFB, DFB, DDT) to each family.

Control Expert provides a **User-defined Library**. This library contains the family **custom FAMILY**; however, it does not contain objects at this time.

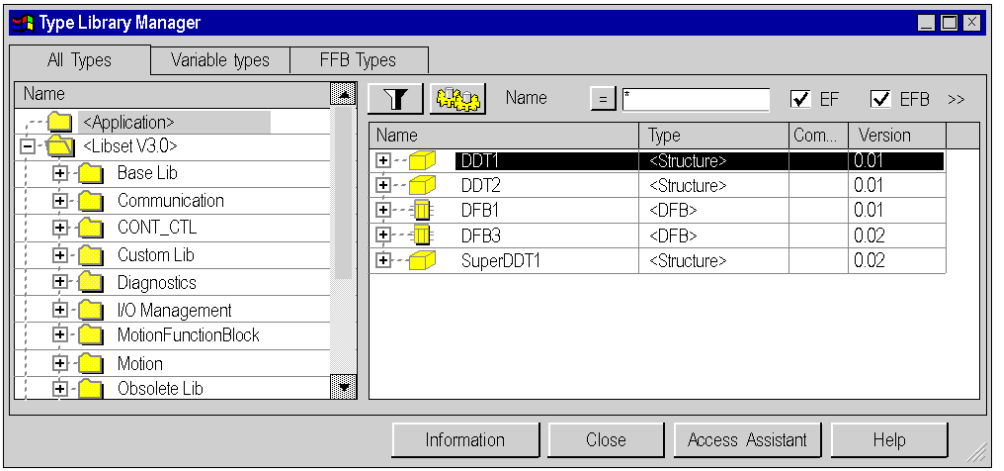
Create a new library

Step	Action
1	In the **Tools** menu, select the **Type Library Manager** option. **Result**: The following window will be displayed. 
2	Under **Name** select the **Libset** directory.
3	Select the **Create New Library** command in the context menu.
4	Enter the name and confirm with **Enter**.

Create a new family

Step	Action
1	In the **Tools** menu, select the **Type Library Manager** option.
2	Under **Name** select **User-defined Library** or a self-created library.
3	Select the **Create New Family** command in the context menu.
4	Enter the name and confirm with **Enter**.

Link the object to the family

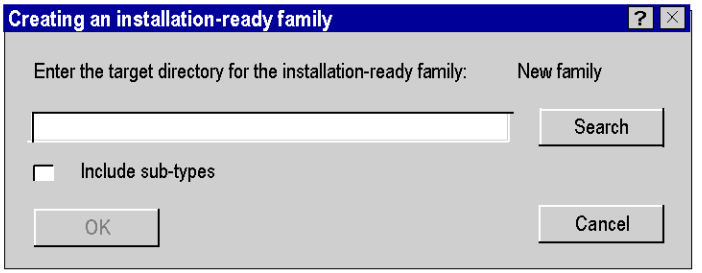
Step	Action
1	In the **Tools** menu, select the **Type Library Manager** option. **Result**: The following window will be displayed. 
2	Under **Name** select **User-defined Library** or a self-created library.
3	In the user-defined library, select **Custom FAMILY** or a self-created family.
4	Select the object on the right side in the **Name** area.
5	Drag the object onto **Custom FAMILY** or a self-created family. **Result**: The selected object is located in **Custom FAMILY** or the new family.

Creating an installable family

Introduction

The user-defined library can be saved in a file. It can then be imported into another programming device.

Procedure

Step	Action
1	In the **Tools** menu, select the **Type Library Manager** option.
2	In the **Name** area, display the contents of the **Libset** directory.
3	Select a family in the **User-defined Library** or a self-created library.
4	Select the **Create Install-Ready Family**command from the shortcut menu. **Result**: The following window will be displayed.  **Note**: Enable the **Include Sub-Types** checkbox when the family contains sub-types that are not defined in the new project.
5	Select the directory where the family is to be stored (installed). Use **Search** to select the directory.
6	Click **OK**.

Notes for EF/EFBs

NOTE: When you try to create an installable family from a family that contains EF/EFBs, a warning will be displayed that the EF/EFBs will not be exported. For more information about using families with EF/EFBs in other Control Expert installations, refer to Updating a family of a library, page 336.

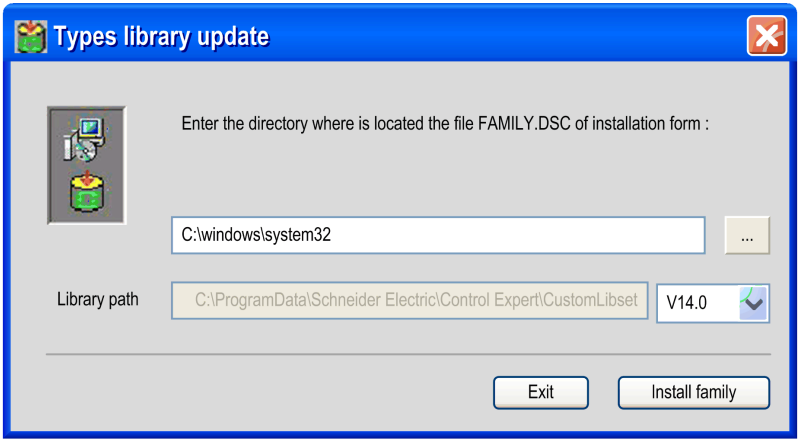
Updating a family of a library

Introduction

You can update the type library on a programming device. You can add new EF, EFB, DFB, DDT to use in your program.

NOTE: Close Control Expert before updating.

Procedure

Step	Action
1	Select **Start** > **Programs** > **EcoStruxure Control Expert** > **Types Library Update**. The **Types Library Update** window will be displayed. 
2	Select the directory where the family is stored using the **...** button.
3	Select the **Libset** version (e.g. V14.0) you want to update. (To change the **Library path** please refer to Library configuration options, page 597.)
4	Click on **Install Family**.
5	Answer **Yes** to the displayed question.
6	Answer **OK** to the displayed warning and click on **Exit**.

NOTE: If you want to use a family in several **Libset** versions of Control Expert, you must proceed as described above for each **Libset** version.

Family with EF/EFBs

You have created a user defined family (e.g. toolkit family) with the EFB tool kit and installed it in Control Expert.

You have also copied DFBs and/or DDTs into the family in Control Expert.

If you now want to create an installable family, the EF/EFBs from this family will **not** be exported.

Using families with EF/EFBs in other Control Expert installations

If you want to use user-defined families with EF/EFBs in other Control Expert installations, proceed as follows:

Step	Action
1	Use your user-defined family (e.g. ToolKit family) to create an **installable family**, page 334. **Result:** The DFB/DDTs of the family will be exported, the EF/EFBs will **not** be exported. A warning will advise you.
2	Install the user-defined family with the EF/EFBs you originally created with the EFB toolkit in the new Control Expert installation.
3	Install the family you exported in Step 1via **Start > Programs > EcoStruxure Contro lExpert > Types Library Update**. **Result:** Since the original family from the EFB toolkit and the **installable family** from Control Expert have the same name, a message will appear that this family already exists and you will be asked if you want to replace it.
4	Answer **No** to the displayed question. **Result:** Another question is displayed whether you want to add the additional types to the existing family instead.
5	Answer **Yes** to the displayed question. **Result:** All other types (DFB/DDTs) will be added to the family. No existing types will be replaced.

NOTE: If you want to use a family in several **Libset** versions of Control Expert, you must proceed as described above for each **Libset** version.

Consistency Check Assistant

At a Glance

An Assistant will automatically be displayed for complex or multiple operations to check the library or any objects of the projects for consistency to enable easy executing of required processes.

When nested objects are loaded or transferred the Assistant will help you select the objects to be transferred. It will show you the available operations, and which operations do or do not require preparation (renaming, updating, etc.).

NOTE: Depending on the complexity of your action, the Assistant will present the objects in a flat or hierarchical format. In some cases a partial list of objects may be selected for processing, and other cases require confirming or canceling the entire operation. The following sections list some examples and meanings.

NOTE: The illustrations below show the windows with different columns. This results from the option to configure the columns differently, just like in the Data Editor, page 428.

Unchecked elements

When using the Type Library Manager, some elements are not considered during the compatibility check of DFB and EFB types.

These elements are:

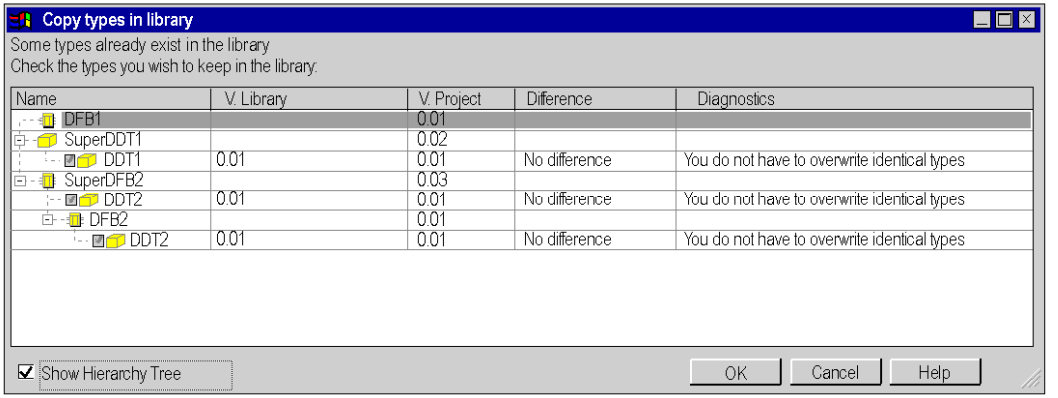
- Initial value elements that contain the initial value of data.
- The Save option showing whether a data element accepts or doesn’t accept the backup / restore mechanism.
- The RW program option showing whether a public variable can or cannot be written from the DFB and EFB program.
- The RW program instance option showing whether a public variable can or cannot be written from the program launcher.
- The Pin Number of the parameter. (**Note**: This check is only not considered with DFBs)

Libset Version 2.2

NOTE: Upgrading the **Libset** version of the project may lead to have a list of differences during the comparison of EFs, EFBs, DFBs used in the application with the library. In case of a project build with a **Libset** version 2.2, the version number (V1.0) may be identical with the selected **Libset**, even if differences are existing. Upgrading the project with the new **Libset** requires a rebuild of the project and a download in the PLC.

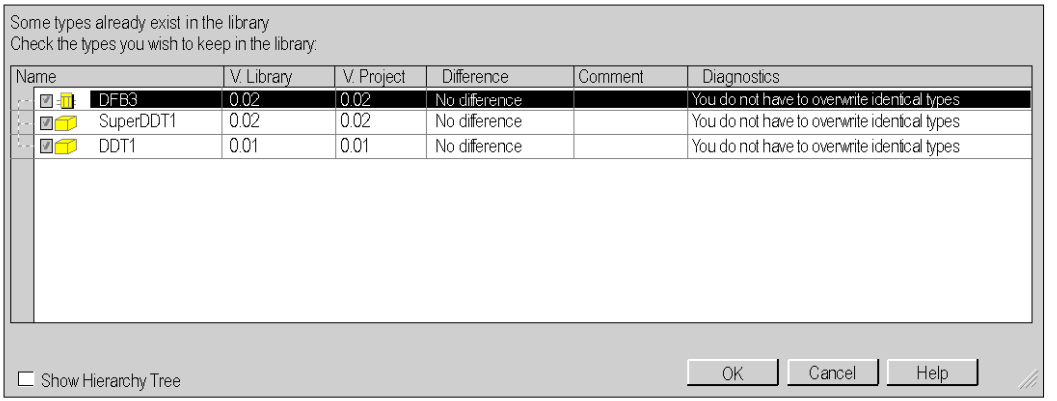
Example of a hierarchal illustration with global confirmation.

The following illustration shows an example of the assistant for a hierarchal view with required global confirmation. The window shows that certain object types already exist in the library and you are asked to confirm whether the types should remain in the library.



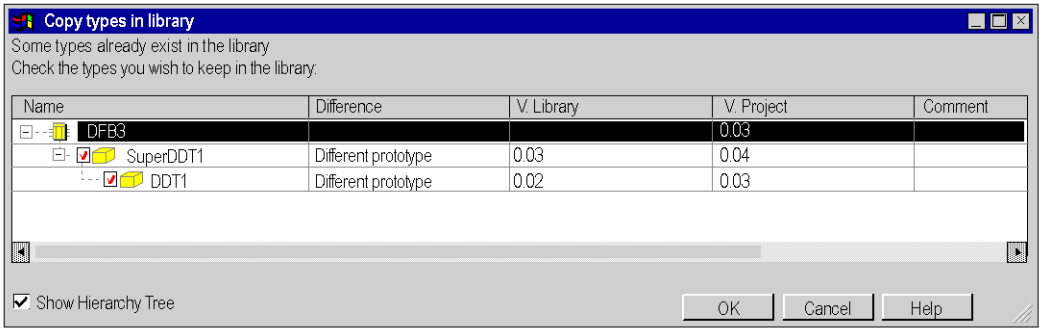
Example of a flat illustration with global confirmation.

The following illustration shows an example of the assistant for a flat (non-hierarchal) view with required global confirmation. The window shows that certain object types already exist in the library and you are asked to confirm whether the types should remain in the library.



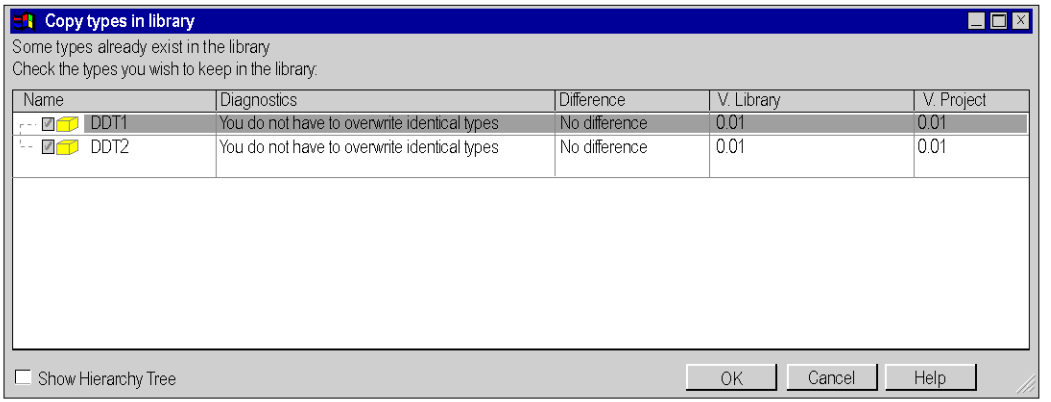
Example of a hierarchal illustration with selective confirmation.

The following illustration shows an example of the assistant for a hierarchal view with a selective confirmation option. The window shows that certain object types already exist in the library and you are asked to confirm whether certain types should remain in the library.



Example of a flat illustration with selective confirmation.

The following illustration shows an example of the assistant for a flat (non-hierarchal) view with a selective confirmation option. The window shows that certain object types already exist in the library and you are asked to confirm whether certain types should remain in the library.



Library Version Management

At a Glance

The Type Library Manager enables you to compare all FFB types of a project with the FFB types of the libraries.

If there are differences between the FFB types in the project and FFB types in the library, the **LIBRARY Version Management** dialog will be displayed showing the relevant information.

This dialog box has two buttons to updated the FFB types of the project with the FFB types of the libraries.

Compare project to library

The **Compare Project to Library command...** Begin via the context menu of the Type Library Manager (right-click).

LIBRARY Version Management tab.

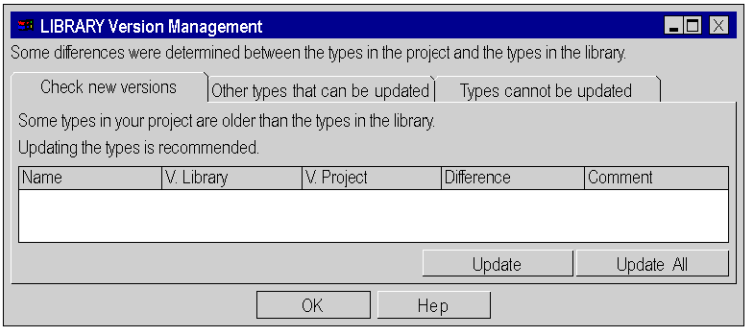
The **LIBRARY Version Management** consists of three tabs:

- **Check new versions**
- **Other types that can be updated**
- **Types cannot be updated**

The first two tabs have buttons to update the FFB types of the project with the FFB types of the libraries.

Register: Check new versions

This tab displays FFB types of the project that are older than the types of the libraries.



Elements of the **Check New Versions** tab:

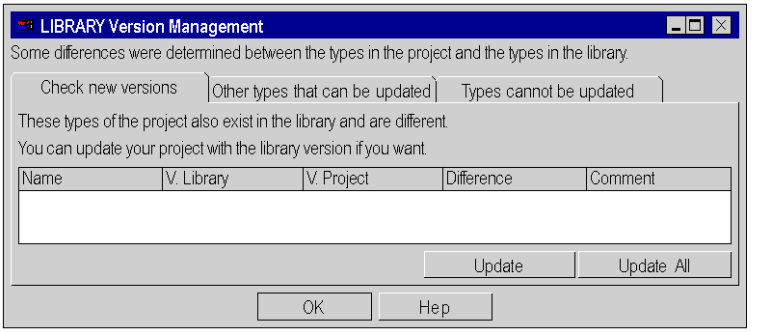
Element	Description
Name	Name of the FFB type
V. Library	Version of the FFB type in the library
V. Project	Version of the FFB type in the project
Difference	Determined difference between the two versions
Comment	Comment about the version difference
Button **Update**	Updates the selected FFB type with the FFB type from the library.
Button **Update All**	Updates all FFB types shown in this tab with the FFB types from the library.

Element	Description
Button **OK**	Closes the window.
Button **Help**	Opens Control Expert Help

NOTE: Updating FFB types leads to **Rebuild All Project**. The PLC must be stopped and a global download of the project must be performed.

Register: Other types that can be updated

This tab shows FFB types of the project that are different from those in the libraries, but can be updated.



Elements of the **Other types that can be updated** tab:

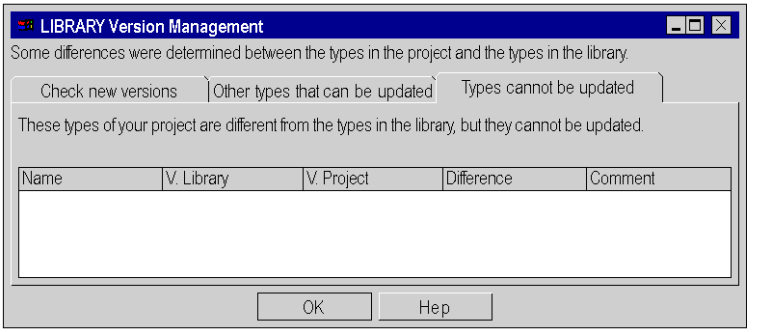
Element	Description
Name	Name of the FFB type
V. Library	Version of the FFB type in the library
V. Project	Version of the FFB type in the project
Difference	Determined difference between the two versions
Comment	Comment about the version difference
Button **Update**	Updates the selected FFB type with the FFB type from the library.
Button **Update All**	Updates all FFB types shown in this tab with the FFB types from the library.

Element	Description
Button **OK**	Closes the window.
Button **Help**	Opens Control Expert Help

NOTE: Updating FFB types leads to **Rebuild All Project**. The PLC must be stopped and a global download of the project must be performed.

Register: Types cannot be updated

This tab shows FFB types of the project with differences from those in the libraries that are so great that they cannot be updated. In this case you could updated FFB types only after you did an export/import of your project.



Elements of the **Types that cannot be updated** tab:

Element	Description
Name	Name of the FFB type
V. Library	Version of the FFB type in the library
V. Project	Version of the FFB type in the project
Difference	Determined difference between the two versions
Comment	Comment about the version difference

Element	Description
Button **OK**	Closes the window.
Button **Help**	Opens Control Expert Help

Using the type library of an older Unity Pro/Control Expert Version

At a Glance

NOTE:

Unity Pro is the former name of Control Expert for version 13.1 or earlier.

If you have already used an older version of Unity Pro/Control Expert, and if you have created your own type library and families to store DFBs or DDTs, you have to provide this own type library and family for the new version of Control Expert.

Providing the type library and families

Follow the steps below to install a new version of Control Expert:

Step	Action
1	Start the Unity Pro/Control Expert software.
2	Use **Tools** to open the **Type Library Manager**.
3	Create a ready-to-install family for each family containing your own DFBs or DDTs (via the context menu for the applicable family). Save these families on your hard drive (but not in the Schneider Electric directories).
4	Install the new version of Control Expert.
5	Go to **Programs > Schneider Electric > EcoStruxureControlExpert > Control Expert** and start the Tool **Type Library Update** and install all of your families. **Note:** If you want to use your families in several **Libset** versions, you must repeat this step for every **Libset** version.
6	Start the Control Expert software.

Step	Action
7	Make sure that all of your families have been installed.
8	Create an application.
9	Copy each DFB that belongs to your own type library from the library into your application.
10	Then copy each DFB from the application into the library family that it came from.
11	All of your libraries and families are now ready for the new Control Expert Version.

Data Editor

What's in This Part

Description of the Data Editor 349
Creating derived data types (DDT) 356
Creating data types of user function blocks (DFB) 378
Managing of instances of data belonging to the family of function blocks (EF) 380
Managing variable instances belonging to the EDT/DDT/IODDT/Device DDT families 390
General Functions 425

Object of this part

It guides the user through the general features available in the data editor in order to:

- Create data types,
- Create data type instances,
- search for data types or instances.

Description of the Data Editor

What's in This Chapter

Data Editor Access 349
Description of Data Editor tabs 351

Object of this chapter

This chapter gives an overview of the data editor.

Data Editor Access

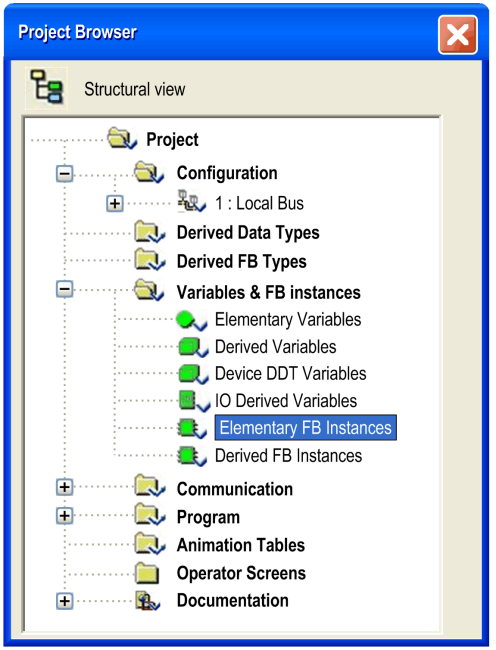
Introduction

The project can be accessed from the structural view in the data editor. It has features which support the following:

- Creating data types,
- Archiving or using function block data types in a library,
- Creating data type instances,
- Displaying data structures in hierarchical manner,
- Searching/sorting/filtering data.

Accessing the Data Editor

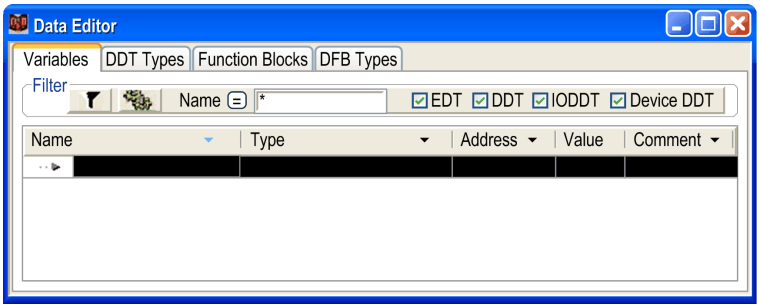
Structural view of a project:



Carry out the following steps:

Step	Action
1	With the right mouse-button click on the **Variables and FB instances** directory.
2	Select the **Open**command in the shortcut menu. **Result**: The data editor opens. The **Variables** tab is shown by default.

Data Editor:

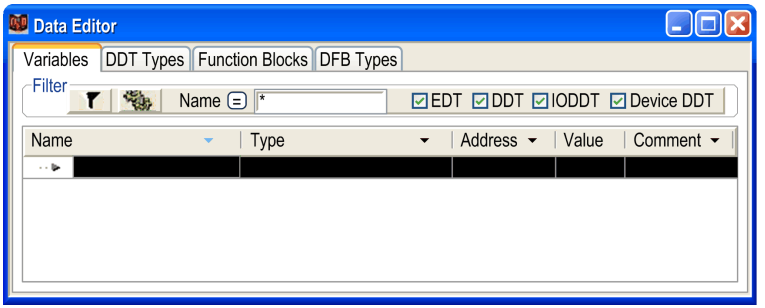


NOTE: You can open the editor directly via data instances which belong to a family defined by you. If you want to do this open the editor from one of the **subdirectories** in the **Variables and FB instances** directory. The variables to be displayed will be listed in the selected subdirectory.

Description of Data Editor tabs

Data Editor Elements

Appearance:



Description:

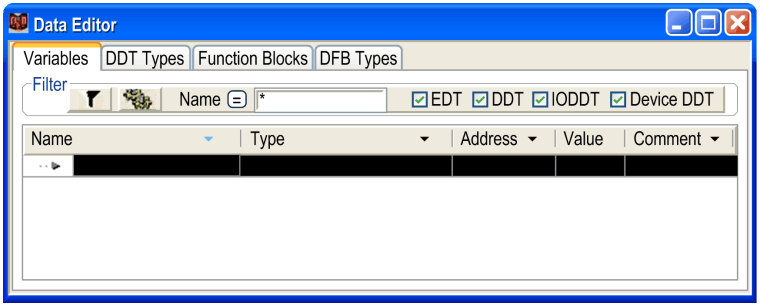
Element	Description
Filter	
Button 	Clicking on this button updates the display corresponding to the filter term defined in the **name** field.

Element	Description
Button 	Clicking on this button opens a dialog box, page 433 for defining the filters.
Button 	Clicking on this button inverts the filter. The button changes from **=** to **<>** and vice versa.
Name	Column where you enter the name (symbol) of the instance or the data type you want to display. You can use the place holders (***** or **? for this.**).
EDT	Displaying elementary data types
DDT	Displaying derived data types
IODDT	Display the derived data types (DDT) that reference inputs/outputs
Device DDT	Display the device derived data types
Properties	
Name	Column where you enter the name (symbol) of the instance or data type
Type	Column where you select the variable type.
Address	Column where you reference the variable instance with an address in the controller (located instance).
Value	Column where you initialize the variable.
Comment	Column where you enter the comment for the variables.

NOTE: The different fields can be assigned parameters, page 425.

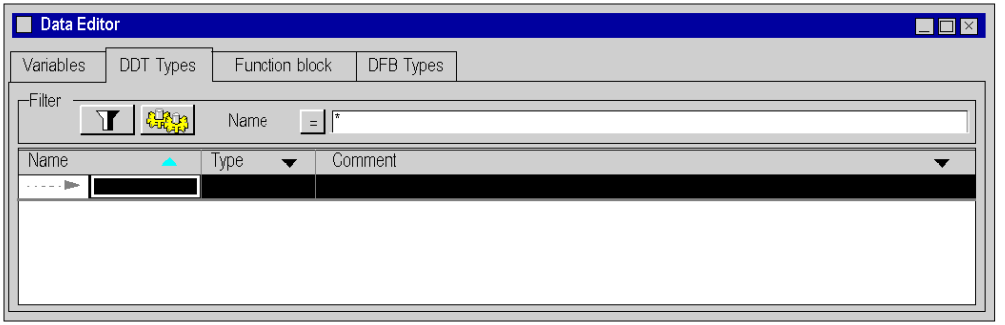
Tab: Variables

The **Variables** tab enables the administration of variable instances that belong to the families EDT/ DDT/ IODDT/ Device DDT



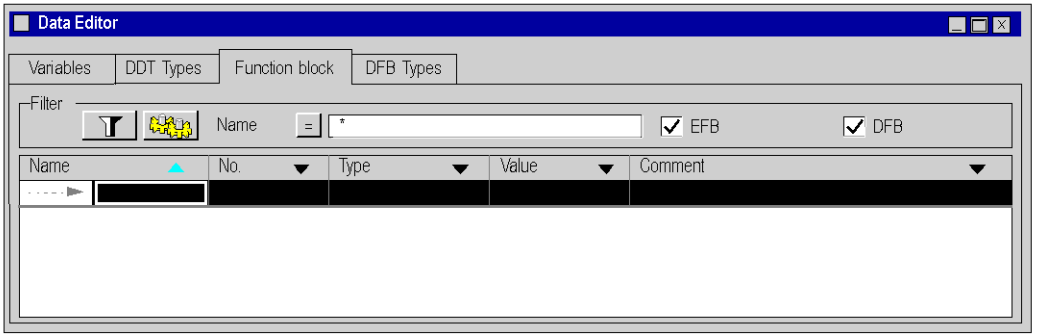
Tab: DDT Types

The **DDT Types** tab is used to manage derived data types (structures or arrays):



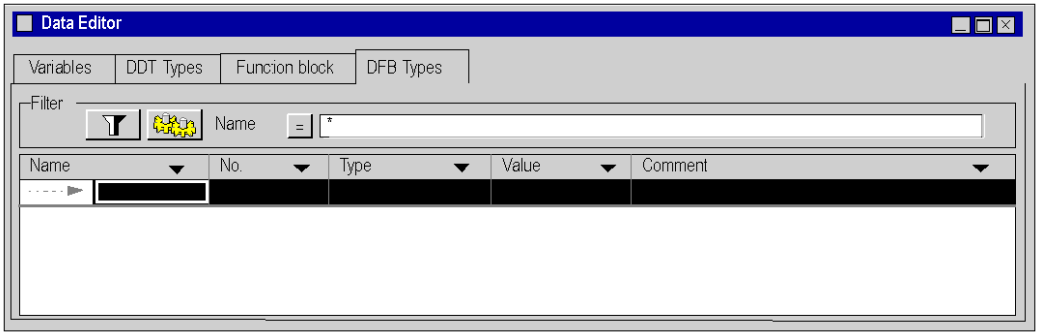
Tab: Function blocks

The **Function Blocks** tab enables the management of the variable instances of type EFB or DFB that belong to the function block family.



Tab: DFB Types

The **DFB Types** tab enables the administration of data types of derived function blocks (DFB).



Creating derived data types (DDT)

What's in This Chapter

Access to derived data types (DDT) 356
Creation of a derived data type (DDT)..... 357
Administration of derived data types and their elements 368
Creating Nested Derived Data Types..... 370
Save Derived Data Types (DDT) 375

Object of this chapter

This chapter guides the user through the creation and management of derived data types (DDT).

Access to derived data types (DDT)

Procedure

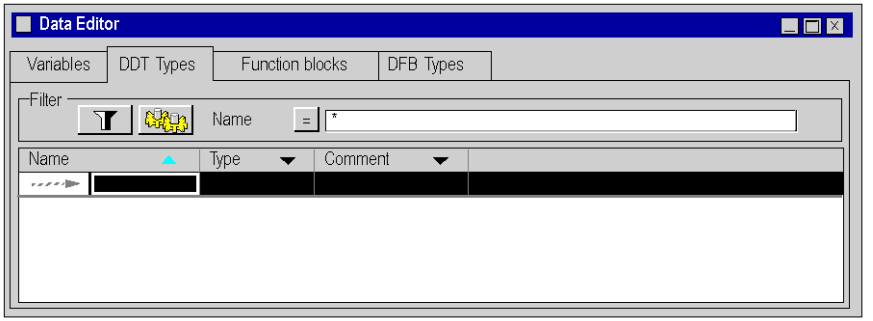
You have the following two options to access the Data Editor to create derived data types:

- from the project structure view via the **Variables and FB Instances** directory,
- from the project structure view via the **Derived Data Types** directory,

Access via the "Variables and FB Instances" directory.

Step	Action
1	In the project structure view, switch to the **Variables and FB Instances** directory.
2	Select the **Open** command in the shortcut menu and the Data Editor will open.
3	Select the **DDT Types** tab in the Data Editor.

Example:



Access via the "Derived Data Types" directory.

Step	Action
1	Switch to the **Derived Data Types** directory in the project structure view.
2	Select the **Open** command in the shortcut menu, which will open the Data Editor that is automatically shown in the **DDT Types** tab.

Creation of a derived data type (DDT)

Introduction

A derived data type corresponds with:

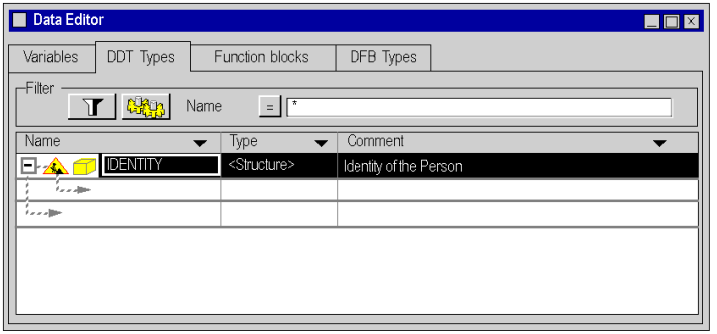
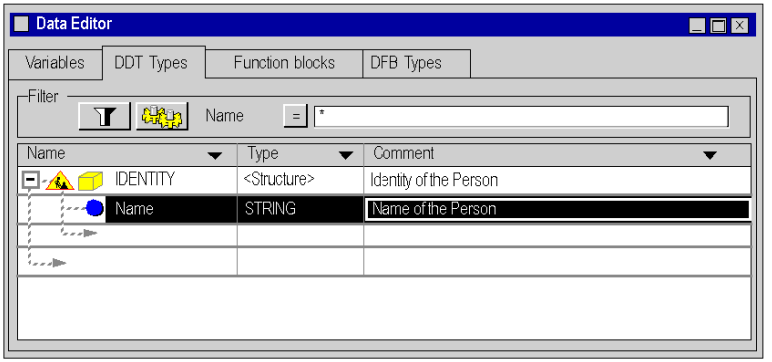
A derived data type is one of the following

- a structure type:
  - Structure for input/output data, the type is not created by the user in this case, but by the **manufacturer** (IODDT),
  - Structure for other data; in this case the type is **created by user**.
- an array type.
- a reference type.

Memory Structure

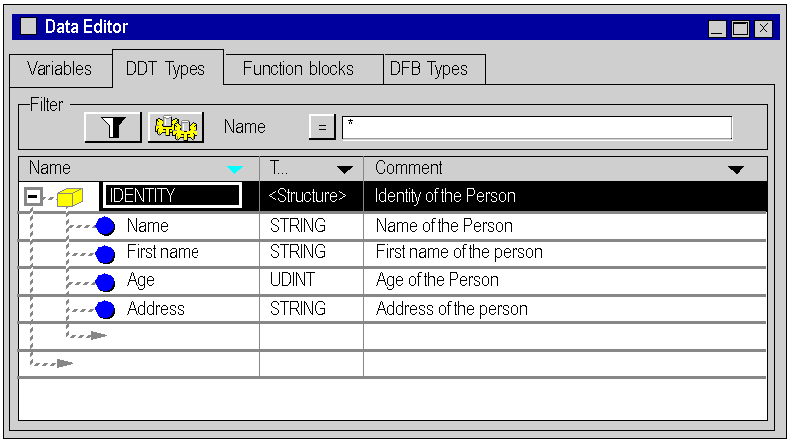
The different PLC families vary in memory structure.

Creating a Structure Type

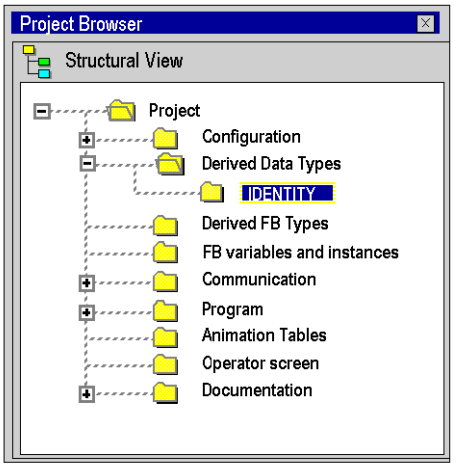
Step	Action
1	Double click in the Data Editor in the **DDT Types** tab on the empty field (marked by an arrow **Name**, and enter the name of the structure type (e.g. IDENTITY); the default type is **<Struct>**. Enter a comment (optional).
2	Call up the newly created structure by clicking on **+**. Example: 
3	Double-click on the corresponding **Name** field and enter the name of the first element of the structure (e.g. Name, Age) and its type and comment (optional). Example: 

Step	Action
4	Double-click on the next cell (marked with an arrow) to enter the name of the next element etc.
5	Go to the name of the structure type (`IDENTITY`) and select **Analyze Type** in the shortcut menu.

Result: The structure was successfully analyzed (the symbol in front of the structure type has changed):



Result: The `IDENTITY` structure is created in the structure view of the project:



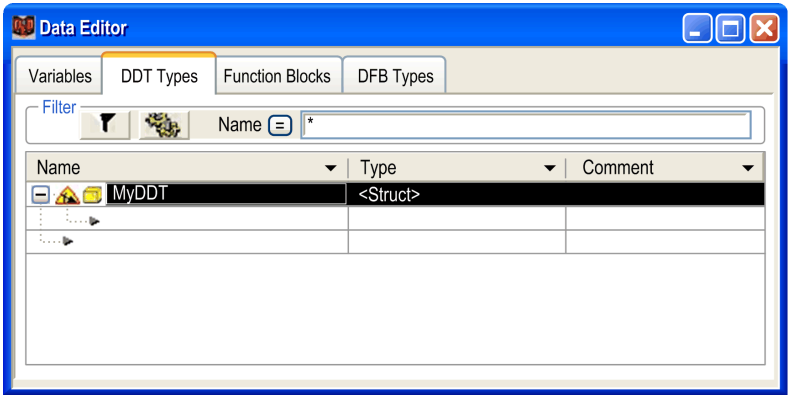
NOTE: Using the shortcut menu and the **Open** command you can access the `IDENTITY` structure.

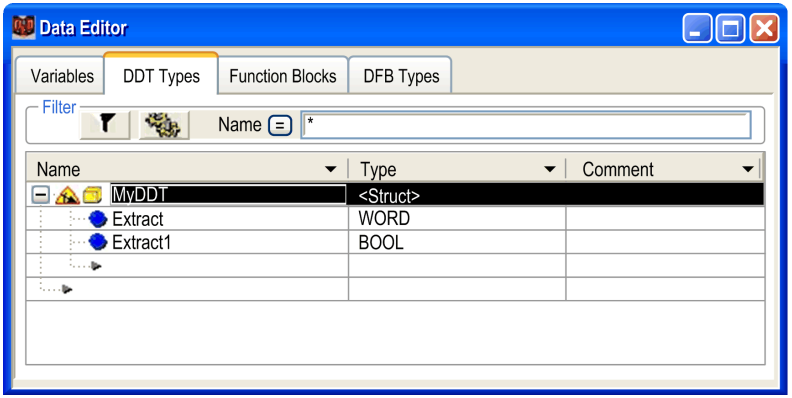
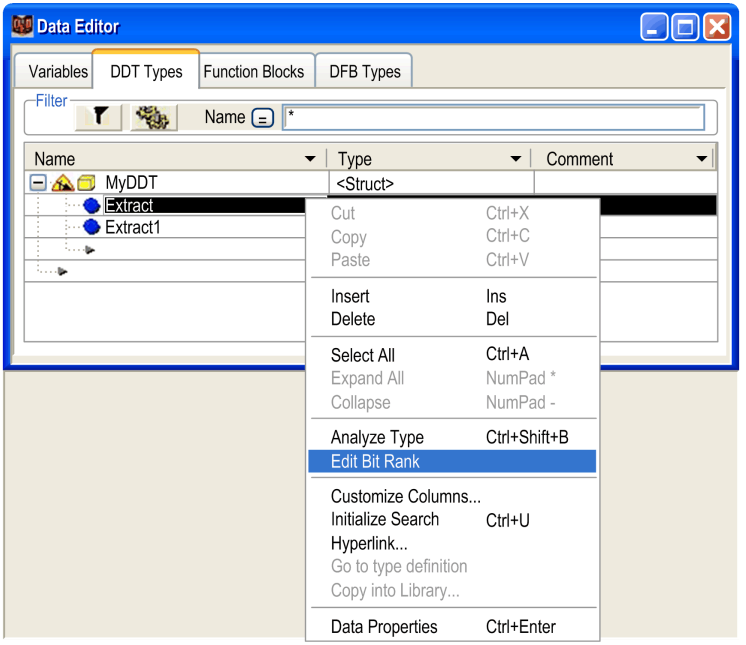
Creating a Structure Type with an Extracted Bit

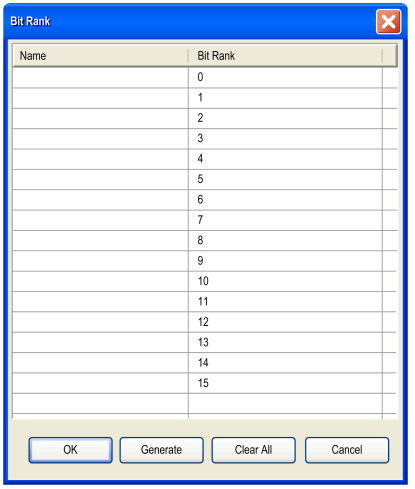
The user can create structure type using extracted bit.

The **Bit Rank** dialog box is accessible by right clicking on the instance or data type which type must be:

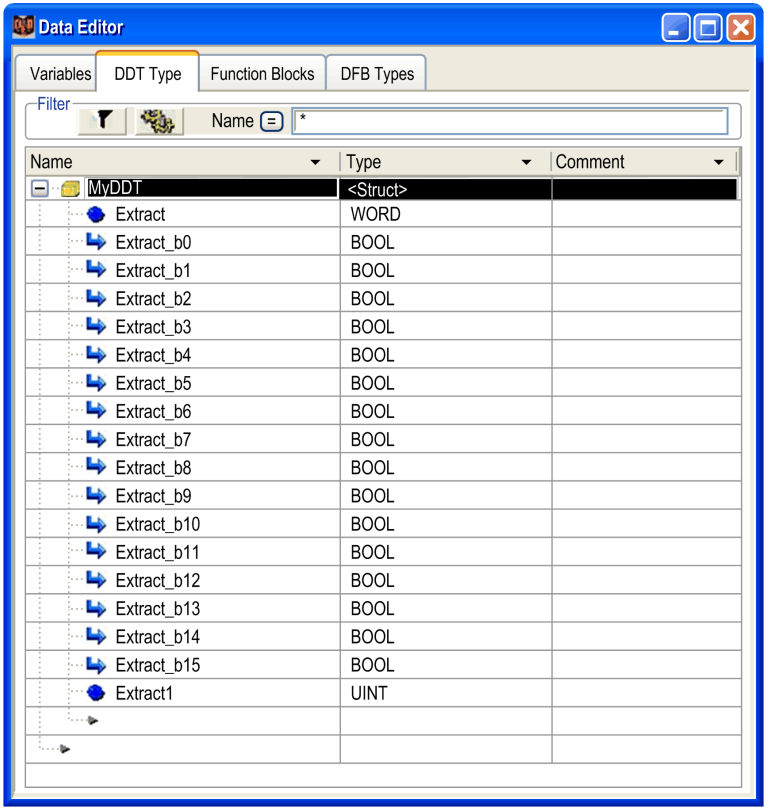
- WORD
- UINT
- INT
- BYTE
- an extracted bit with a compatible parent

Step	Action
1	Double click in the Data Editor in the **DDT Types** tab on the empty field (marked by an arrow **Name**, and enter the name of the structure type (for example, MyDDT); the default type is **<Struct>**. Enter a comment (optional).
2	Call up the newly created structure by clicking on **+**: 

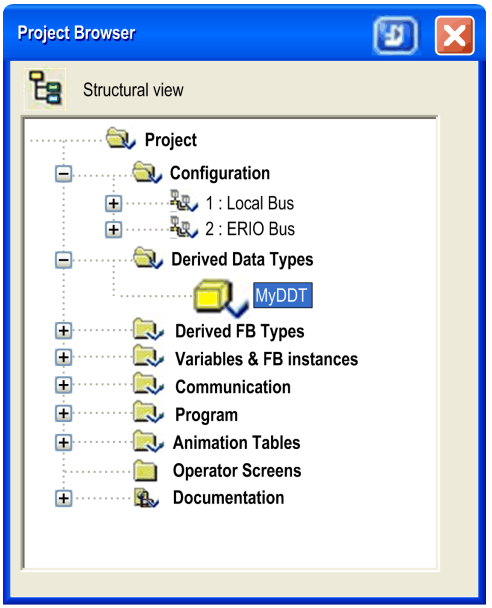
Step	Action
3	Double-click on the corresponding **Name** field and enter the name of the first element of the structure (e.g. Extract, Extract1) and its type and comment (optional): 
4	Double-click on the next cell (marked with an arrow) to enter the name of the next element etc.
5	Select the variable to create the extracted bit and right-click on the variable. The **Edit Bit Rank** is available in the contextual menu list: 

Step	Action
	Here below is the **Bit Rank** dialog box: 
6	Within the **Bit Rank** menu, the user can select to named each variable manually or automatically using the **Generate** button. **NOTE:** The Bit name can be edited manually directly in the **Data Editor**.
7	Go to the name of the structure type (`MyDDT`) and select **Analyze Type** in the shortcut menu.

Result: The structure was successfully analyzed (the symbol in front of the array type has changed):



Result: The `MyDDT` structure is created in the structure view of the project:

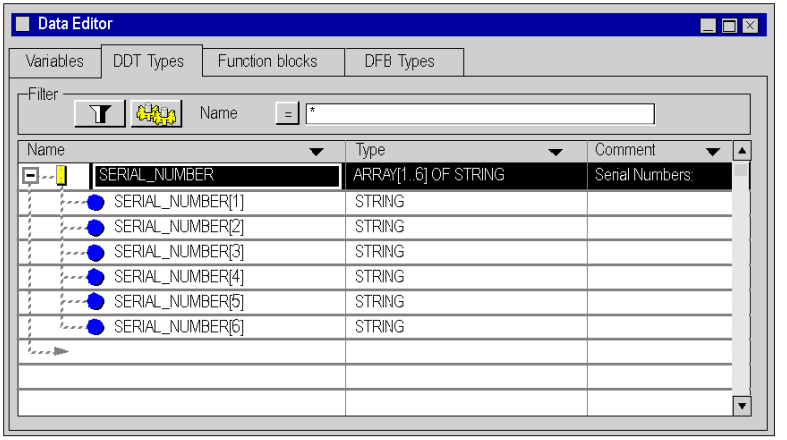


NOTE: Using the shortcut menu and the **Open** command you can access the `IDENTITY` structure.

Creating an Array Type

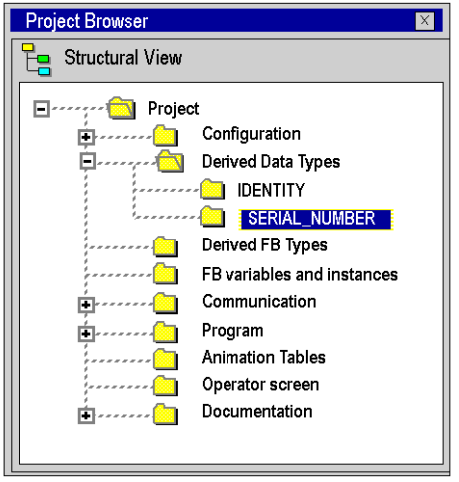
Use this procedure to create an array type:

Step	Action
1	Double click in the **Data Editor** in the **DDT Types** tab on the empty field (marked by an arrow **Name**, and enter the name of the array type (e.g. `SERIAL_NUMBER`); the default type is **<Struct>**. Enter a comment (optional).
2	Double click on the **Type** field, select **<Array>**, and confirm by pressing **Enter**. **Result**: The Variable Type Selection dialog appears.
3	The **ARRAY** checkbox is automatically checked: • Enter the size of the array in the corresponding field (for example, 1..6) • In the **libraries/families** area select the types that should be displayed: ◦ the **<Application>** directory for a derived type already in the application ◦ the **<EDT>** directory for an elementary type ◦ the **<GDT>** directory (displayed but not applicable here) ◦ the **<Libset>** directory for a derived type archived in a library

Step	Action
4	Confirm with **OK**.
5	Change to the name of the array type and go to the shortcut menu and select **Analyze Type**. **Result**: The array was successfully analyzed (the symbol in front of the array type has changed): 

NOTE: Comments cannot be entered for the sub elements of the array type.

The array type `SERIAL_NUMBERS` is created in the project structure view:



NOTE: You can access the array of the `SERIAL_NUMBER` type in the context menu using the **Open** command.

Creating a Reference Type

Step	Action
1	Double click in the **Data Editor** in the **DDT Types** tab on the empty field (marked by an arrow **Name**, and enter the name of the reference type (e.g. `MyRefInt`); the default type is **<Struct>**. Enter a comment (optional).
2	Double click on the **Type** field, select **<Reference>**, and confirm by pressing **Enter**. **Result**: The Variable Type Selection dialog appears.
3	The **REF_TO** checkbox is automatically checked. Select the reference data type, in this example: INT.
4	Confirm with **OK**.
5	In the *Data Editor*, right-click the name `MyRefInt` and select **Analyze Type**. **Result**: The reference was successfully analyzed (the symbol in front of the reference type has changed).

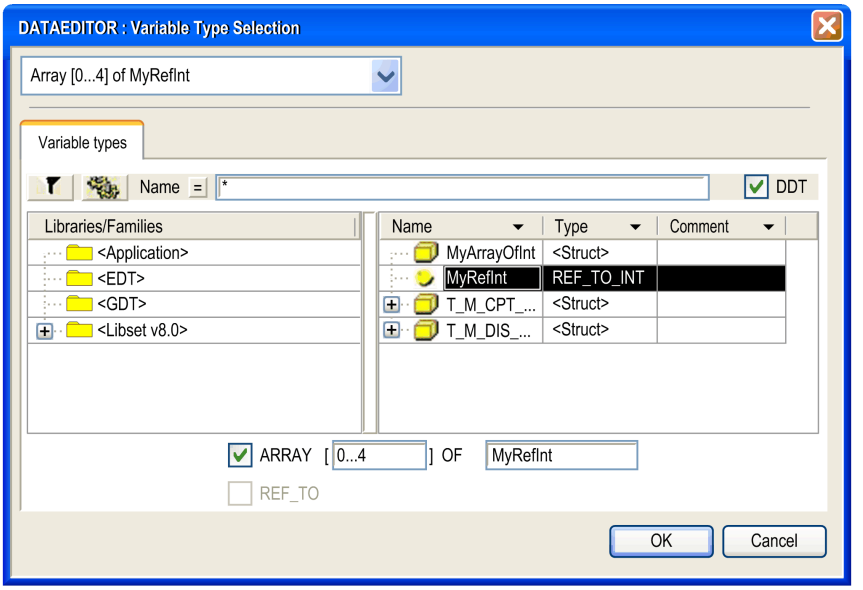
The reference type `MyRefInt` is created in the **Project Browser Structure view**.

NOTE: You can access the reference of the `MyRefInt` type using the contextual menu using the **Open** command.

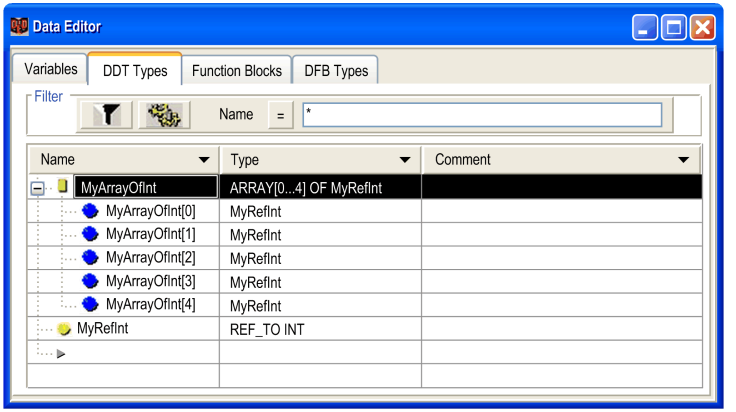
Array of References

You cannot make directly, for example, an Array of REF_TO INT. You must first make a reference DDT with type REF_TO INT as in the figure above. Then you make a new DDT type, an array of the references.

In **<Application>**, select `MyRefInt`:



After confirming with **OK** and performing an **Analyze Type**, you obtain an array of references:

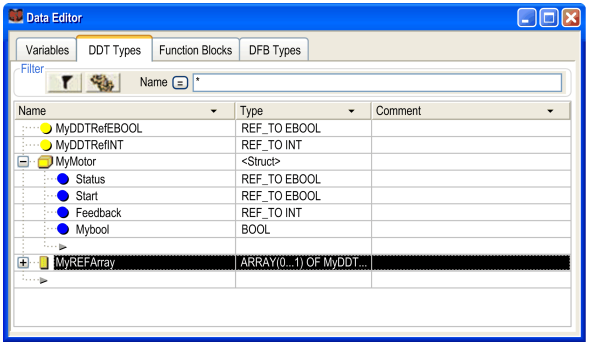


Structure with References

This example shows the creation of an array of reference type variables:

Step	Action
1	Double click **DDT Types** tab in the **Data Editor** on the empty field (marked by an arrow **Name**, enter the name of the structured type and press ENTER. The default type is **<Struct>**.
2	Double click **Types** and select **<Array>** in the drop-down menu.
3	Press ENTER to open the **Variable Type Selection** dialog.
4	Set the size of the array.
5	Select **<Application>**.
6	Select `MyDDTRefINT`.

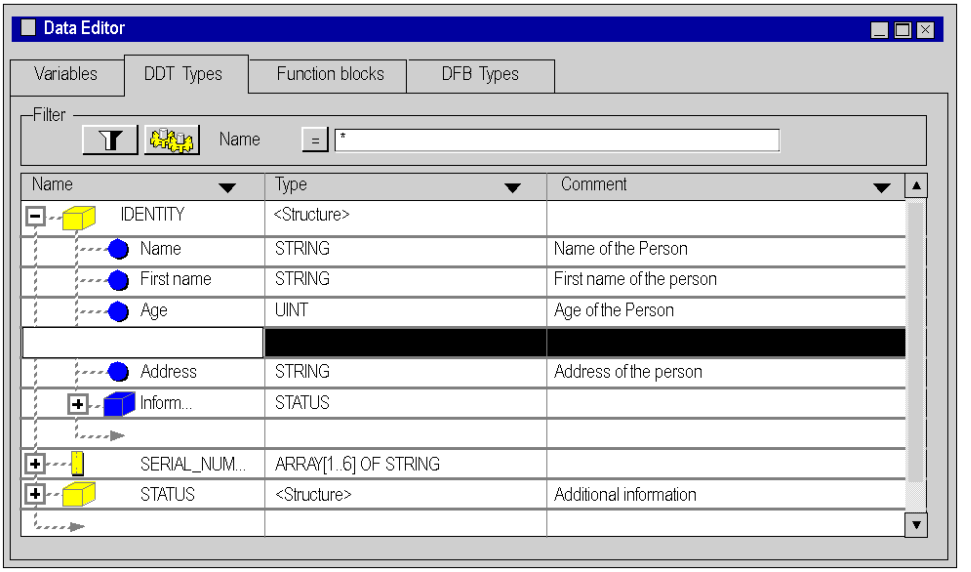
The array of reference types example:



Administration of derived data types and their elements

Inserting an element into a derived data type

Execute the following steps:

Steps	Action
1	Display the desired data type in the data editor in the **DDT Types** tab.
2	Place the cursor on the element in front of which you want to insert another element and select the **Insert new** command in the shortcut menu. Inserting an element into an address element 
3	Enter the attributes of the element into the newly created row.

Deleting an element belonging to a derived data type

Carry out the following steps:

Step	Action
1	Display the desired data type in the data editor in the **DDT Types** tab from which you wish to delete the element.
2	Select the desired element.
3	Use the shortcut menu to select the **Delete** command.

NOTE: Elements belonging to protected data types cannot be deleted.

Deleting a derived data type:

Carry out the following steps:

Step	Action
1	Select the desired data type in the data editor in the **DDT Types** tab.
2	Using the context menu select the **Delete** command.

NOTE: Data types that exist as an instance in the application cannot be deleted.

Creating Nested Derived Data Types

Introduction

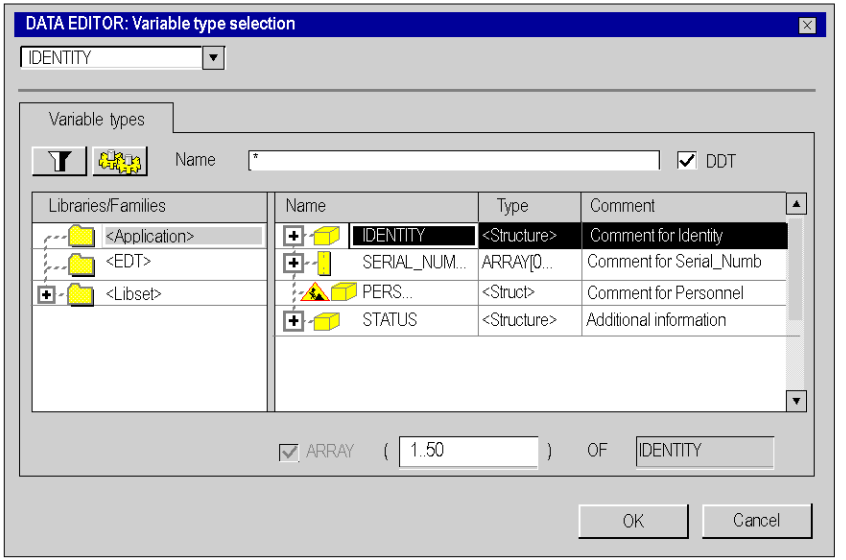
Data types that are part of the DDT family can be nested. The nesting can be organized as follows:

- in structures that contain arrays or structures
- in arrays that contain arrays or structures
- in a combination of these different nestings

Array, that contains structures

Carry out the following steps:

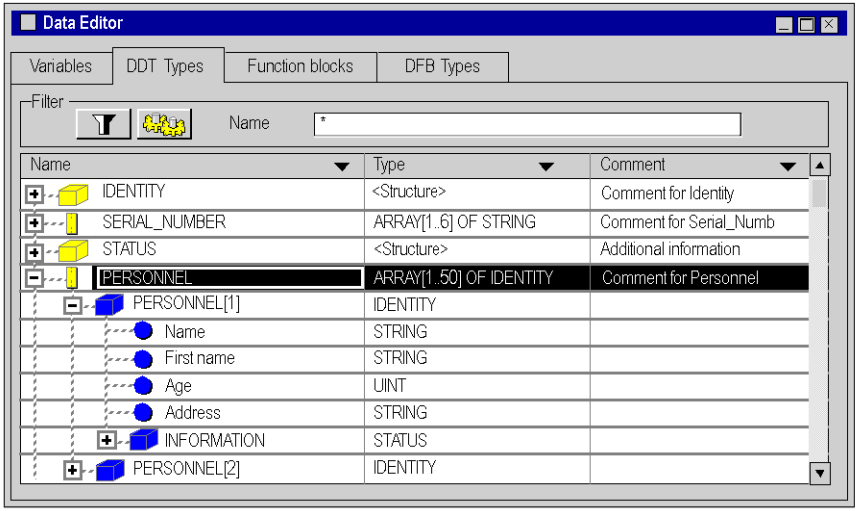
Step	Action
1	Double click in the Data Editor in the **DDT Types** tab on the empty field (marked by an arrow **Name**, and enter the name of the array type (e.g. PERSONNEL); the default type is **<Struct>**. Enter a comment (optional).
2	Double click on the **Type**field, select **<Array>**, and confirm by pressing **Enter**. **Result:** The **Variable Type Selection** appears.
3	The **ARRAY** checkbox is automatically checked: • enter the size of the array in the corresponding field

Step	Action
	• In the **libraries/families** area select the following: ◦ the **<EDT>** directory for an elementary type ◦ the **<Applications>** directory for a type already in the application ◦ the **<Library>** directory for an already archived type See the following window: 
4	Enable the **DDT** checkbox, which will display the derived data types.
5	Select the desired structure in the **Name** column and confirm with **OK**.
6	Change to the name of the array type and go to the context menu and select **Analyze Type**.

NOTE: Comments cannot be entered for the sub elements of the array type.

NOTE: Changing the boundaries of an anonymous array lead to the lost of the initial values of that array. Those values must be re-entered after the modification.

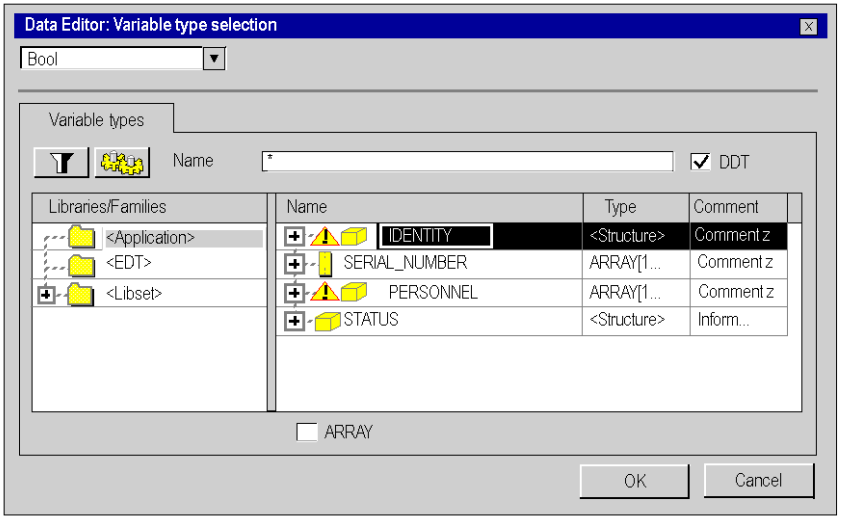
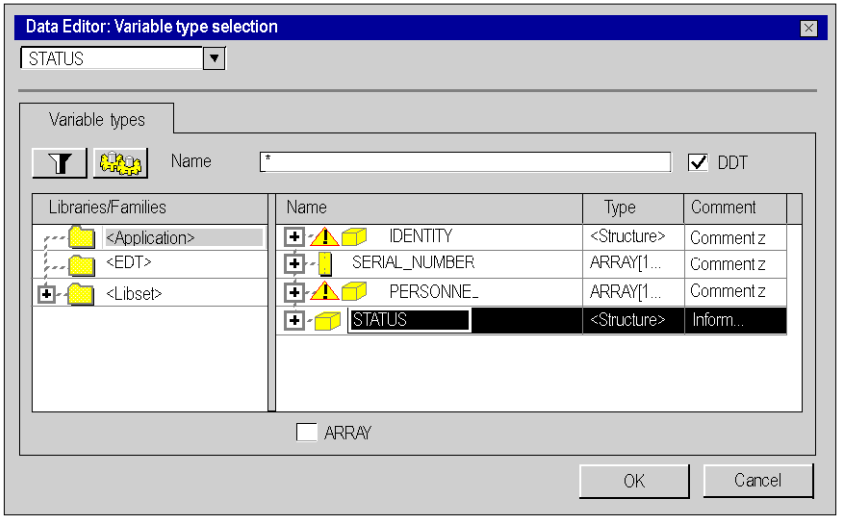
Array of the type **PERSONNEL** that contains structures of the type **IDENTITY**:



Structure containing a structure

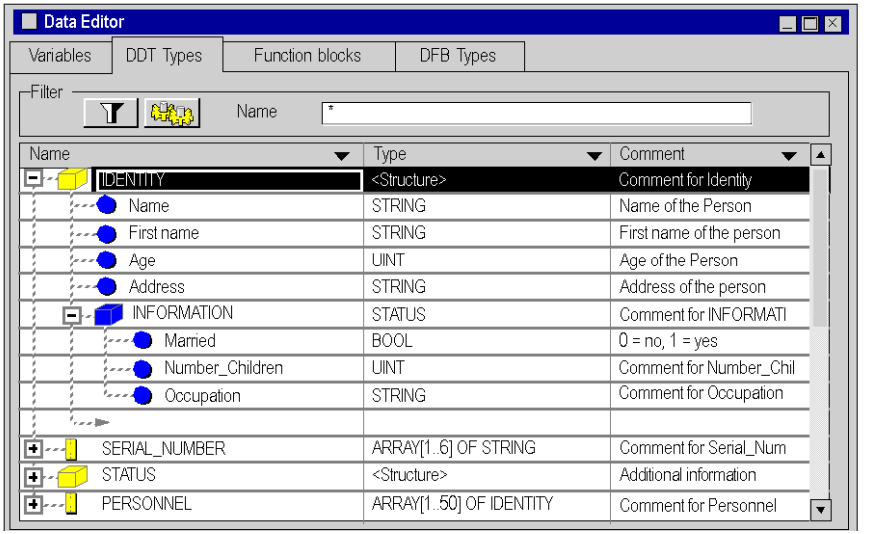
Execute the following actions (provided a structure exists):

Step	Action
1	Click on **+** to display the type of the structure in the **DDT Types** tab in the data editor, to which you want to add an element.
2	Double click on the empty **Name**field and enter the name of the new structure type element. Enter a comment (optional).
3	Double-click on the corresponding **Type** field and click on  .

Step	Action
	The following window appears. 
4	In the **libraries/families** area select: • the **<EDT>** index for an elementary type • the **<Applications>** directory for a type that already exists in the application • the **<Library>** for an already archived type See the following example: 

Step	Action
5	Enable the **DDT** checkbox, which will display the derived data types.
6	Select the desired structure in the **Name** column and confirm with **OK**.
7	Switch to the name of the structure type and select the **Analyze Type**entry in the context menu.

Example: **IDENTITY** structure that contains an **INFORMATION** structure of the **STATUS** type.

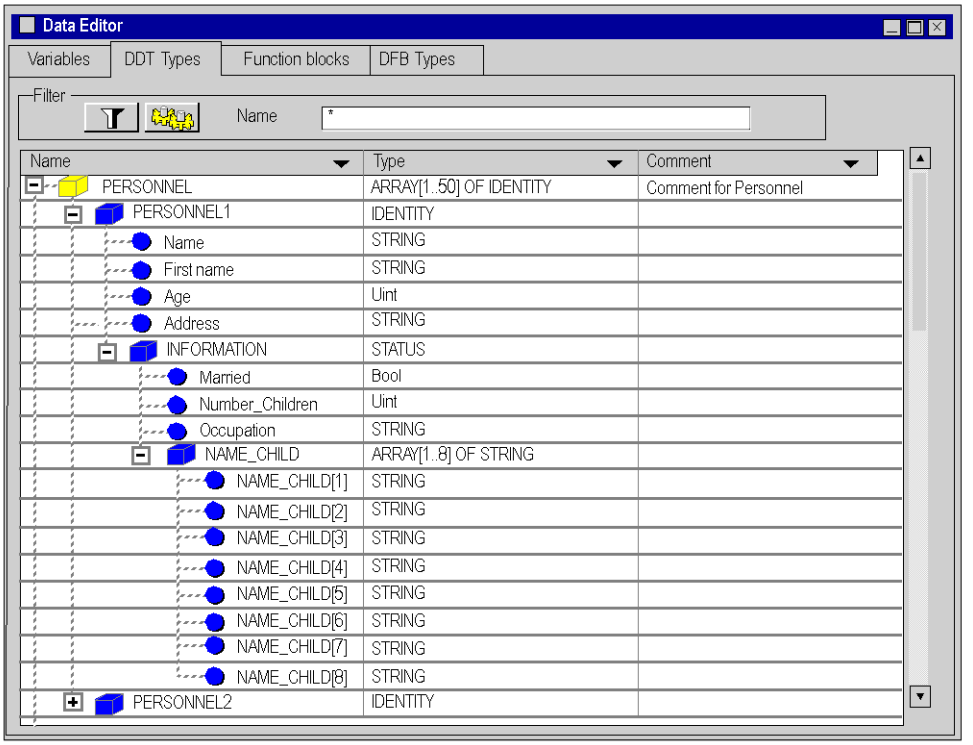


Combination of the different nestings.

The following example shows:

- a **PERSONNEL** array consisting of **IDENTITY** type structures
- **IDENTITY** structures containing **STATUS** type structures
- **STATUS** structures containing arrays of the **NAME_CHILD** type

Example:



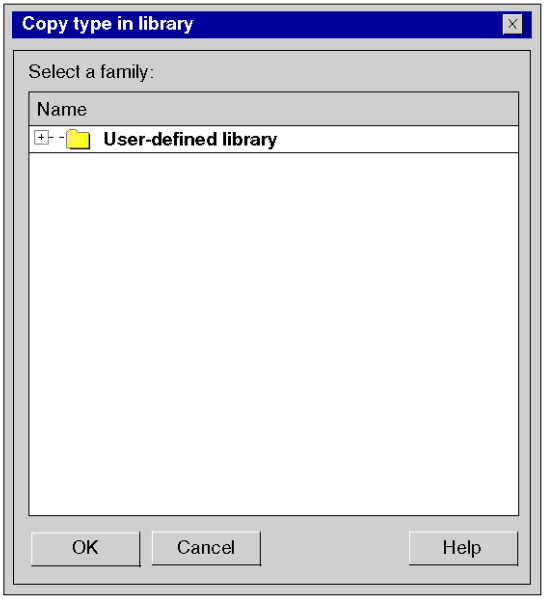
Save Derived Data Types (DDT)

Description of how to save

You can copy the data types you created into your library to reuse them in another project.

Filing a data type in the library

Carry out the following steps:

Step	Action
1	Select the **DDT Types** tab in the Data Editor.
2	Select the data type that you want to save in the library.
3	Using the shortcut menu select the **Copy into Library** command. The following dialog box opens: 
4	Show the desired family to select the directory where you want to archive the data type.
5	Confirm with **OK**.

NOTE: This procedure can also be started from the project browser in the **Derived Data Types** directory.

Restoring a data type in the library

See loading an object from the library., page 327

Creating data types of user function blocks (DFB)

What's in This Chapter

Creation and Management of Data Types and Function Blocks (DFB) 378

Object of this chapter

This chapter guides the user through the creation and management of user function block data types.

Creation and Management of Data Types and Function Blocks (DFB)

At a Glance

See Derived Function Blocks (DFB), page 1254.

This chapter explains the **DFB Types** tab modes of the Data Editor with the following descriptions:

- accessing the DFB Types
- creating a DFB Type
- programming a DFB Type
- managing the elements of a DFB Type
- nestling of the DFB Types
- saving DFB Types

Managing of instances of data belonging to the family of function blocks (EF)

What's in This Chapter

Creating / Deleting a variable instance of function blocks (EFB/DFB) 380
Displaying function block variable instances in the editor 383
Changing the attribute for variable instances in a function block 386

Object of this chapter

This chapter guides the user through the creation and management of data instances belonging to block function types such as:

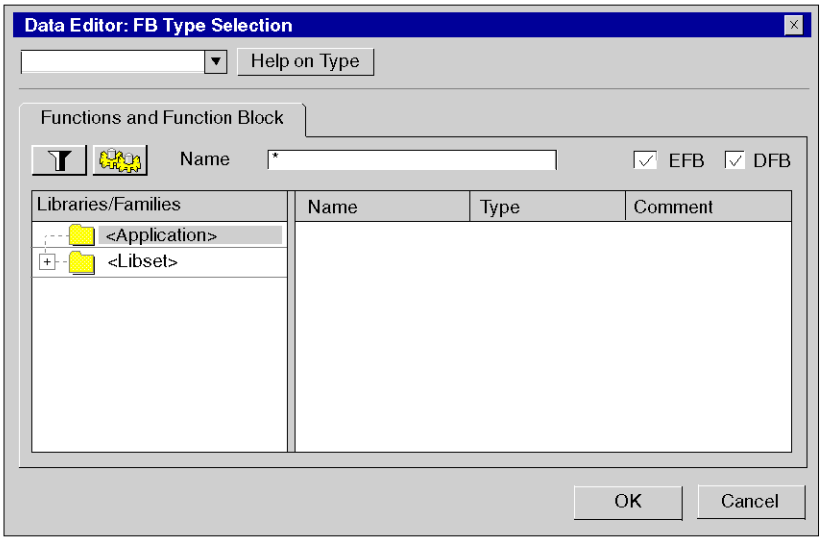
- Elementary function blocks (EFB),
- Derived function blocks (DFB).

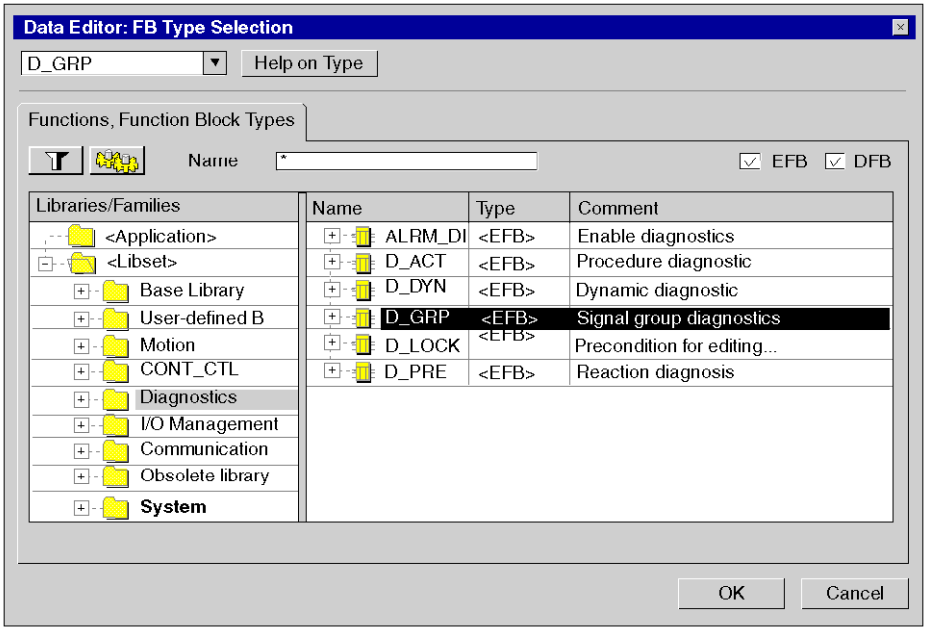
Creating / Deleting a variable instance of function blocks (EFB/DFB)

Creating a function block instance

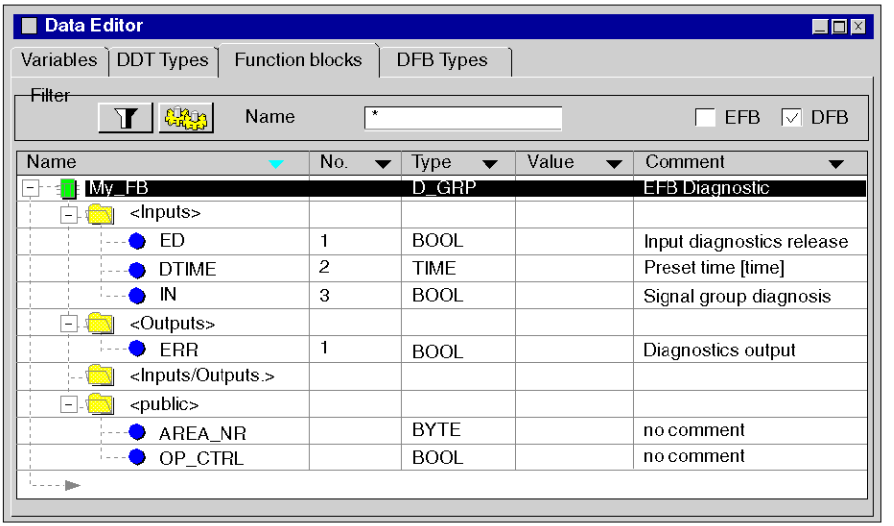
Carry out the following steps:

Steps	Actions
1	In the Data Editor select the **function blocks** tab.
2	Double-click the empty cell for **Name** (marked with an arrow), enter the name of the instance and confirm by pressing the **Enter**button.
3	Double-click on the corresponding **Type** field and click on  .

Steps	Actions
	The following window appears: 
4	In the **libraries/families** area select the following: • the **<Applications>** directory for a type already in the application, • the **<Library>** directory for an already archived type.

Steps	Actions
	Example: Select an EFB type archived in the **Diagnostics** library: 
5	In the **Name** column select the EFB or DFB type and confirm with **OK**.

Example:



Deleting a function block instance

Instances that are accessed at least once cannot be deleted in the data editor.

Example from an ST section:

```
FBI_1(IN:=TRUE, OUT=>VarA);
```

The application explained above will block the deletion because `FBI_1` is accessed.

```
VarA:=FBI_1.OUT
```

The application explained above will **not** block the deletion because `FBI_1` is not accessed.

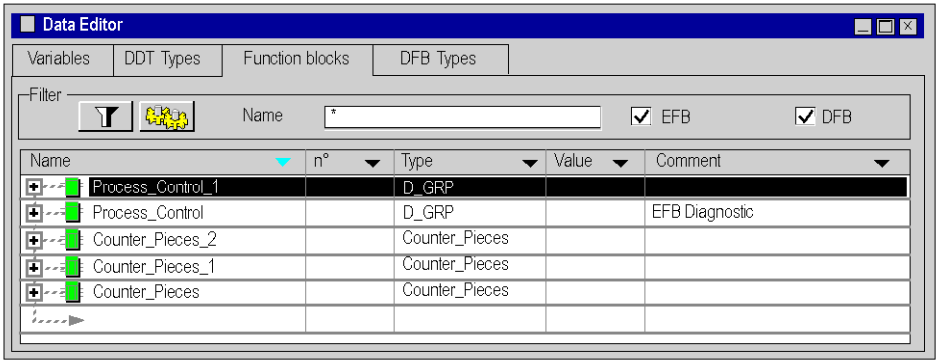
Displaying function block variable instances in the editor

Display sequence in a column

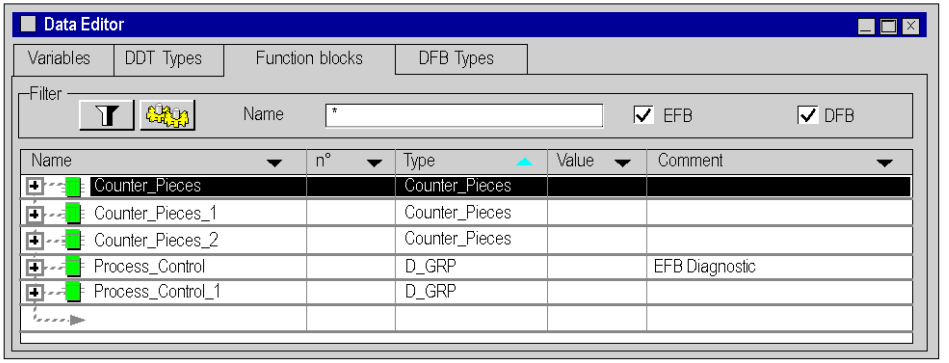
The variable instances can be displayed in **ascending** or **descending alphabetical order**. This sequence will be used in a column chosen by you.

To do this, click on the area that contains the name of the column in question and the arrow will turn blue.

Sorting in descending alphabetical order in the column **Name**



Sorting in ascending alphabetical order in the column **Type**

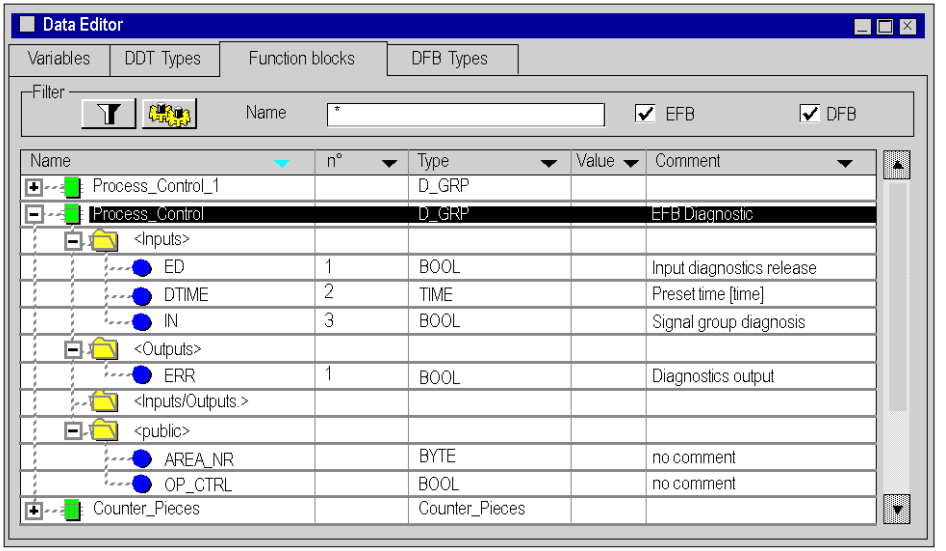


Displaying elements from one function block instance

To show the elements that belong to an instance of the function block type, carry out the following actions:

Step	Action
1	In the Data Editor select the **function blocks** tab.
2	Click on the **+** beside the instance of the desired function block type and the elements will then be shown on screen.
3	Carry out the procedure again if you want to display the interface data or the internal data for the function block on the screen.

Example: Display the instance **Control_Process** of the type EFB

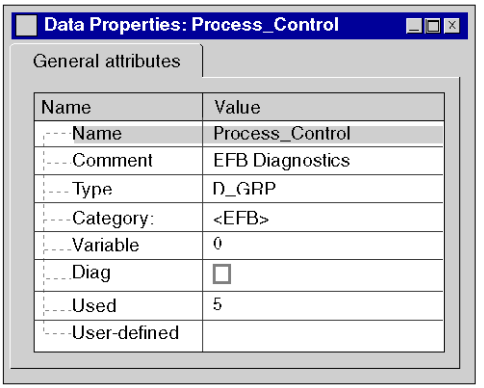


Displaying the attributes

For an **unopened** window **Data Properties**:

Step	Action
1	In the Data Editor select the **function blocks** tab.
2	Select the variable instance.
3	From the context menu select the command **Data Properties** and the attributes will be displayed in a window.

Example:



For a window that is already open **Data Properties**:

After the instance has been chosen, the window **Data properties** is updated automatically with the display for the attribute in question.

Sorting the variable instances

See Filtering of Data, page 433.

Changing the attribute for variable instances in a function block

Detailed information

The internal organization of a variable instance of function blocks (input / output parameters, public / private variables, code) cannot be changed from the instances, but must be performed from the type of the instance **DFB Types**) tab.

The changes listed below do not effect the internal organization of the function block, but rather its instantiation.

Changing the name of an instance

Carry out the following steps:

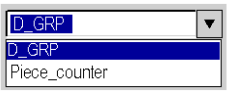
Step	Action
1	In the Data Editor select the **function blocks** tab.
2	Double-click on the instance name, enter the new name, and confirm with **Enter**.

NOTE: the instance name can only be used once.

Changing the type of an instance

There are three options.

In the Data Editor **Function blocks** tab:

Double-click on the variable type and ...	then...
enter the desired type,  depending on the input, an automatic search is carried out.	-
use the type field 	select the desired type.
click the button 	select the type in the "FB type selection", page 429 window.

Reactions to type changes of an instance.

If the type of an FB instance is changed in the Data Editor, the access will be adjusted to the instance.

The access is adjusted as follows:

- The connections are mapped with pin positions.

  Note: Compatibility of data types will not be checked, which means that errors may occur during the analysis following the adjustment.
- Excess pins of the new type remain unused.

- The connections of pins that no longer exist in the new type are lost.

IL and ST sections:

If necessary, formal parameters in IL and ST sections will be renamed according to the new block during formal access. The number of actual parameters will be adjusted during informal access.

> **NOTE:** Instance access in IL or ST sections will only be adjusted if required by the type change. However, if an adjustment is made, the old access will be removed and replaced with a generated access. This means that all comments and formatting contained in the old access will be lost.

FBD and LD sections

The new type will be displayed in the FBD and LD sections following the adjustment.

Managing variable instances belonging to the EDT/DDT/IODDT/Device DDT families

What's in This Chapter

Creation of EDT instances 390
Creation of DDT instances 394
Creating an IODDT instance 400
Managing a Device DDT instance 403
Managing an IODDT, DDT or an Array type instance 407
Creation of multiple variable instances 413
Displaying the variable instances in the editor 415
Changing attributes of variable instances 418
Creation of a Reference Type Instance 422

Object of this chapter

This chapter guides the user through the creation and management of variable instances belonging to the following families:

- Elementary data types (EDT)
- Derived Data Types (DDT)
- Derived Data Types that reference IODDT inputs/outputs (input / output data types)
- Device Derived Data Types (Device DDT)

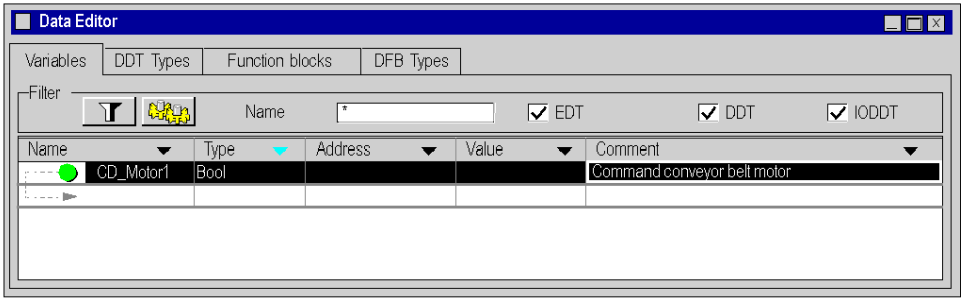
Creation of EDT instances

Instance of an Unlocated Variable

Creating the instance of an **unlocated** variable:

Step	Action
1	Select the **Variables** tab in the data editor.
2	Double-click the empty cell **Name** (marked with an arrow) and enter the name of the instance.
3	Double-click the **Type** cell and enter the type of the instance.
4	Enter a comment if necessary.

Example:

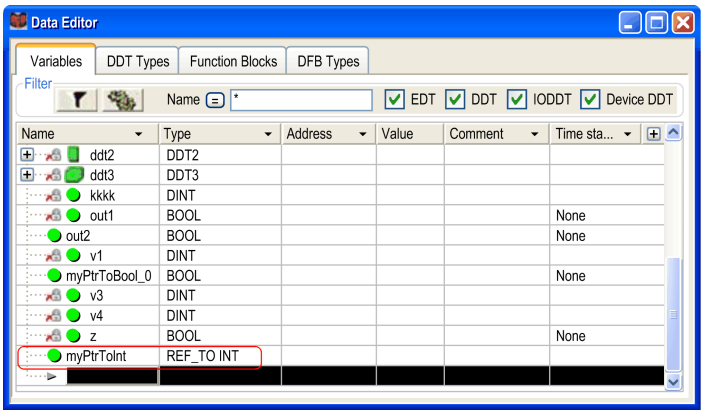


Instance of a Referenced Variable

Creating an instance of a **Referenced** variable:

Step	Action
1	Select the **Variables** tab in the data editor.
2	Double-click the empty cell **Name** (marked with an arrow) and enter the name of the instance.
3	Double-click the **Type** cell and enter **REF_TO** and the type referenced for this instance.
4	Enter the initial value, page 392 of the variable instance.

Example:

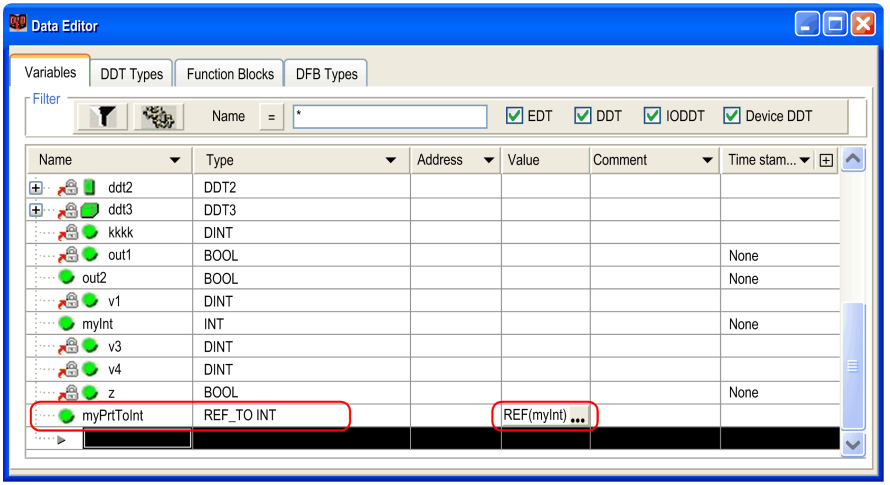


Initial Value of a Reference Type Variable

To set an initial value to a variable of reference type

Step	Action
1	In the **Variables** tab of the **Data Editor**, select the reference type variable to be initialized.
2	Double-click **Value**.
3	Enter `REF(VarName)`, where `VarName` is the variable referenced by the Instance of the Referenced Variable.

Example:



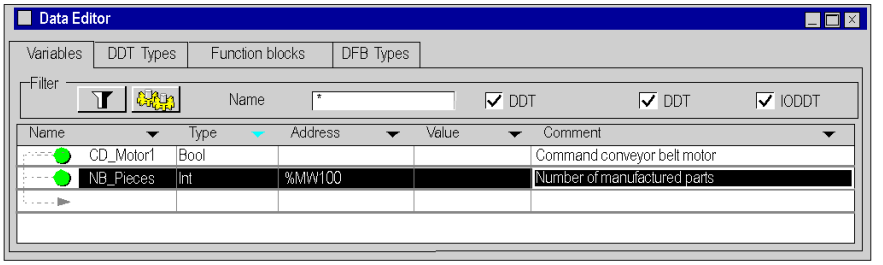
Instance of a Located Variable

Creating an instance of a **located** variable:

Step	Action
1	Select the **Variables** tab in the data editor.
2	Double-click on the empty cell **Name** (marked with an arrow) and enter the name of the instance.
3	Double-click on the **Type** cell and enter the type of the instance.
4	Double-click on the **Address** cell and enter the storage address for the controller.
5	Enter a comment if necessary.

NOTE: Several different instances can be assigned to one address (multiple assignment).

Example:

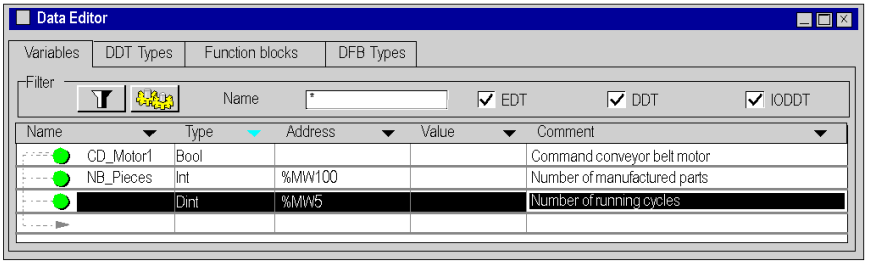


Instance with Direct Addressing

Creating an instance with direct addressing

Step	Action
1	Double-click on the empty **Address** cell and enter the storage address for the controller.
2	Enter a comment if necessary.

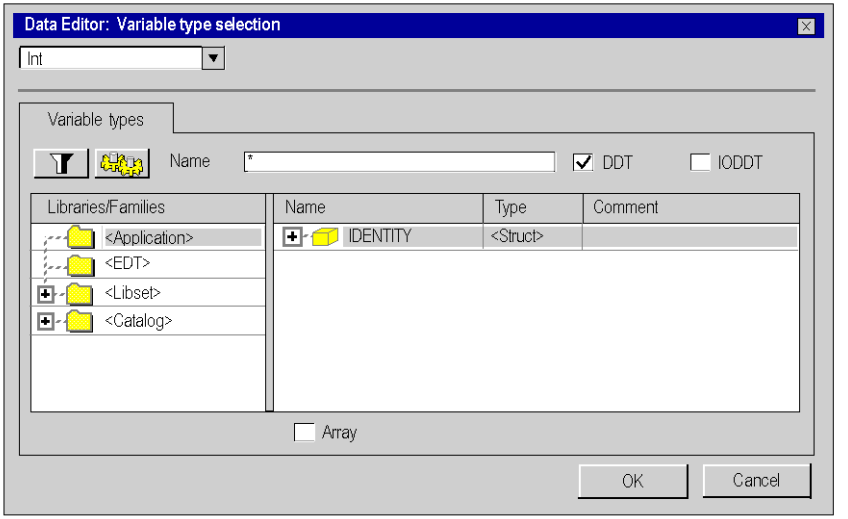
Example:



Creation of DDT instances

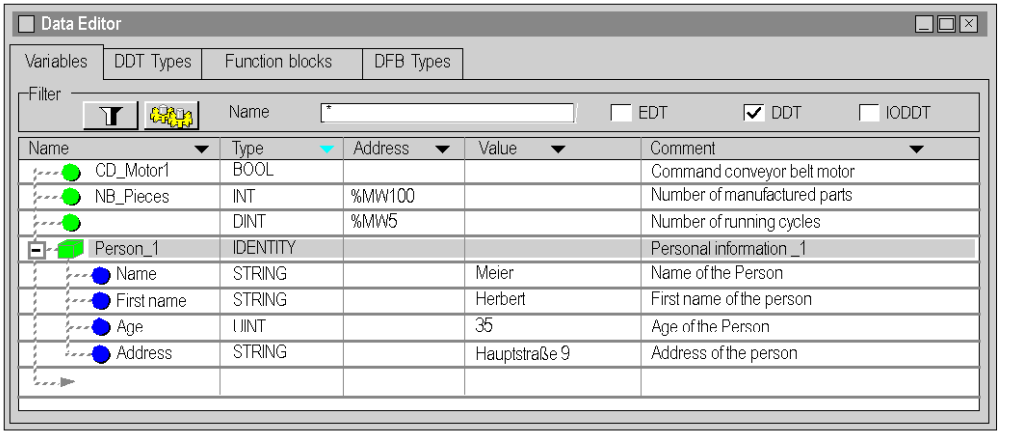
Structure type instance

Creating a Structure type instance:

Step	Action
1	Select the **Variables** tab in the data editor.
2	Double-click on the empty cell **Name** (marked with an arrow) and enter the name of the instance. Enter a comment (optional).
3	Double-click on the corresponding **Type** field and click on  . The following window appears. 
4	In the **libraries/families** area select the following: • the **<Application>** directory for an existing type in the application. • the **<Libset>** directory for an already archived type.
5	Enable the **DDT** checkbox, which will display the derived data types.

Step	Action
6	Select the desired structure type in the **Name** column and confirm with **OK**.
7	Fill out the **Value** column to initialize the elements of the structure. To carry out this task, click on **+** to show the structure.

Example:



NOTE: The comments will be transferred to the selected structure type, but they may be changed for the applicable instance.

Instance of a non-typical array.

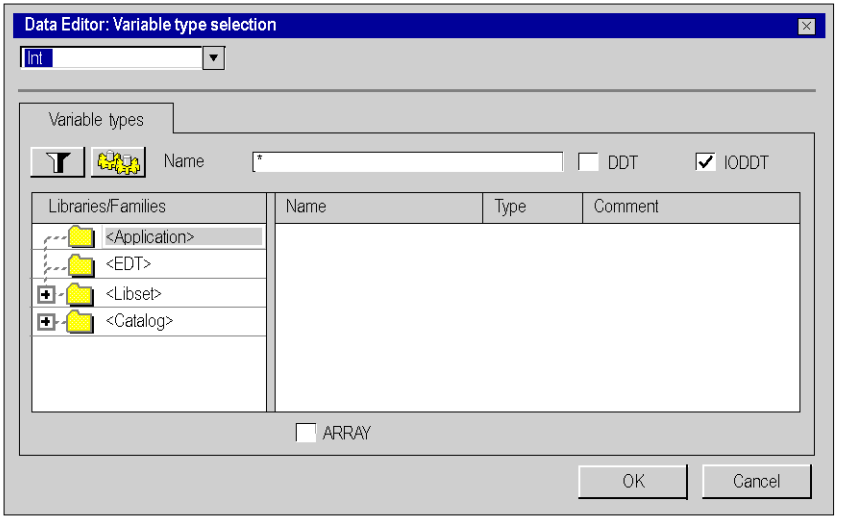
What is a **non-typical** array?

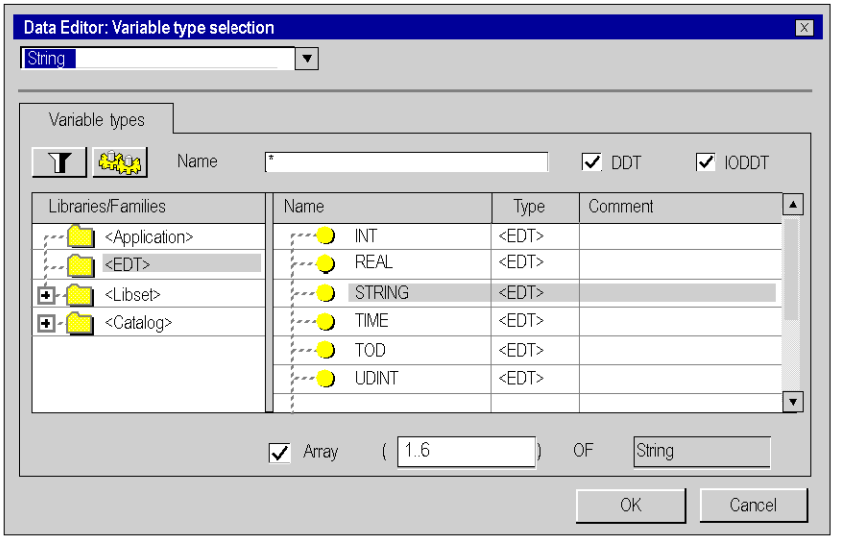
```
Defining an array type
Tab: ARRAY[1..10] of BOOL        ; Array of Type Tab

Array instantiation
My_Tab1 : Tab                     ;typical Array
My_Tab2 : ARRAY[1..10]OF BOOL     ;non-typical Array
```

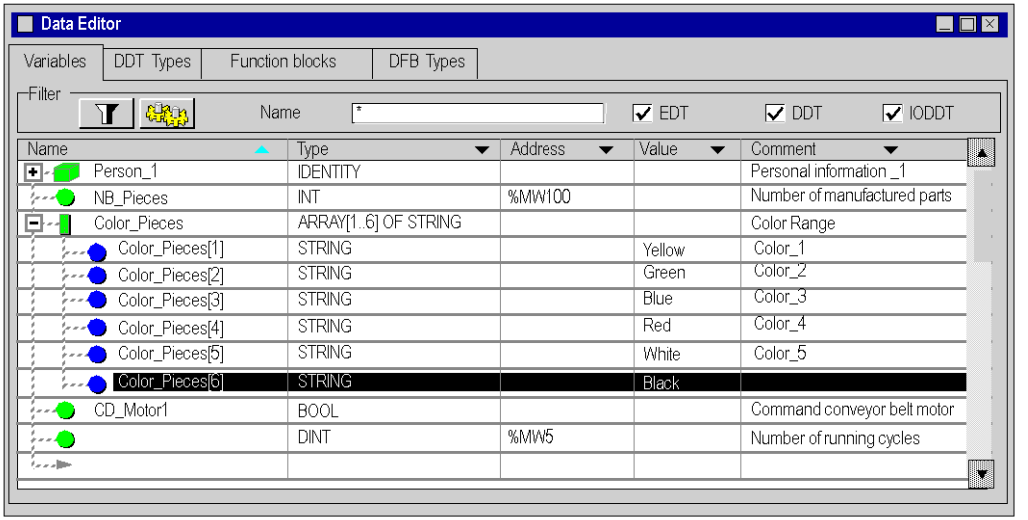
NOTE: My_Tab1 and My_Tab2 are two identical instances. The difference is that the type of the My_Tab1 instance was previously defined (the name of the Type is Tab) while the type of the My_Tab2 instance was defined during instancing.

Creating an instance for a **non-typical** array:

Step	Action
1	Select the **Variables** tab in the data editor.
2	Double-click on the empty cell **Name** (marked with an arrow) and enter the name of the instance. Enter a comment (optional).
3	Double-click on the corresponding **Type** field and click on  . The following window appears. 

Step	Action
4	Enable the **Array** check box and • fill in the field with the size of the array (example 1..6), • In the **libraries/families** area select the types that should be displayed: ◦ the **<EDT>** directory for an elementary type, ◦ the **<Application>** directory for an existing derived type in the application, ◦ the **<Libset>** directory for a derived type archived in a library. • Select the type for the elements of the array in the **Name** column (Example: `STRING` for an EDT). Example:  Confirm with **OK**.
5	Fill out the **Value** column to initialize the elements of the array as needed. To carry out this task, click on **+** to show the elements.

Example:

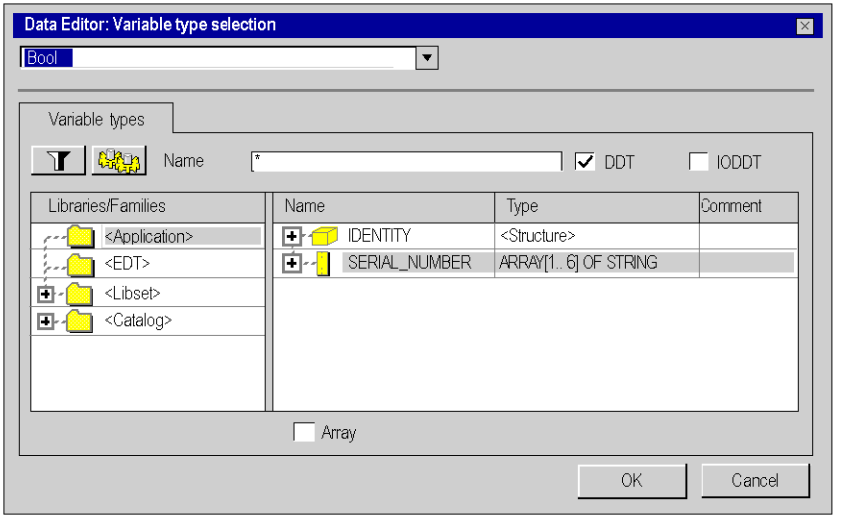


NOTE: Enter comments for the sub-elements of this instance of a **non-typical** array.

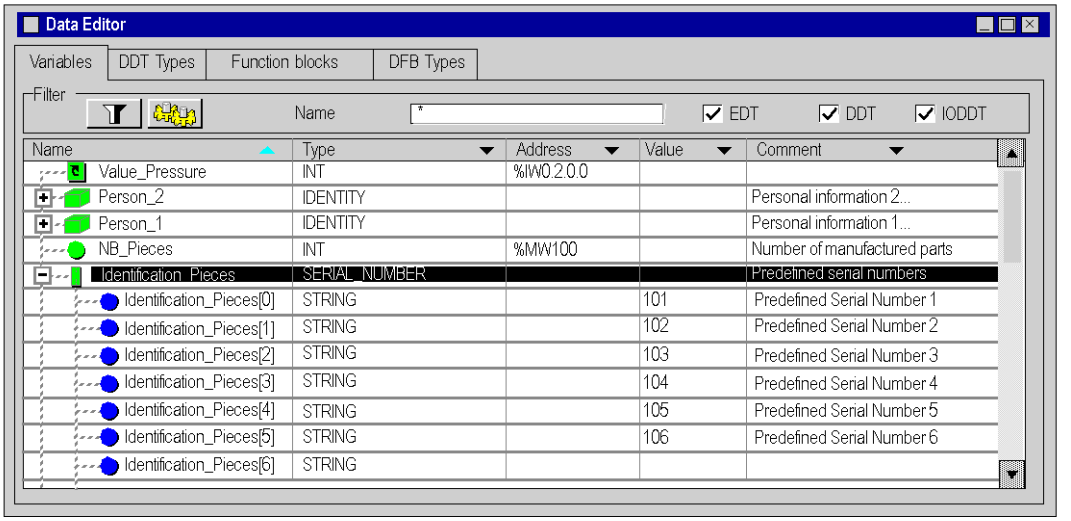
Type Array instance

Creating an Array type instance:

Step	Action
1	Select the **Variables** tab in the data editor.
2	Double-click on the empty cell **Name** (marked with an arrow) and enter the name of the instance. Enter a comment (optional).
3	Double-click on the corresponding **Type** field and click on  .

Step	Action
	The following window appears. 
4	In the **libraries/families** area select the following: • the **<Applications>** directory for a type already in the application, • the **<Libset>** directory for a derived type archived in a library.
5	Enable the **DDT** checkbox, which will display the derived data types.
6	Select the desired array type in the **Name** column and confirm with **OK**.
7	Fill out the **Value** column to initialize the elements of the array as needed. To carry out this task, click on **+** to show the elements.

Example:



NOTE: Comments cannot be entered for the sub elements of this instance of an array.

NOTE: If you resize an array variable type, you will lose the comments and the initial values of the elements of the array.

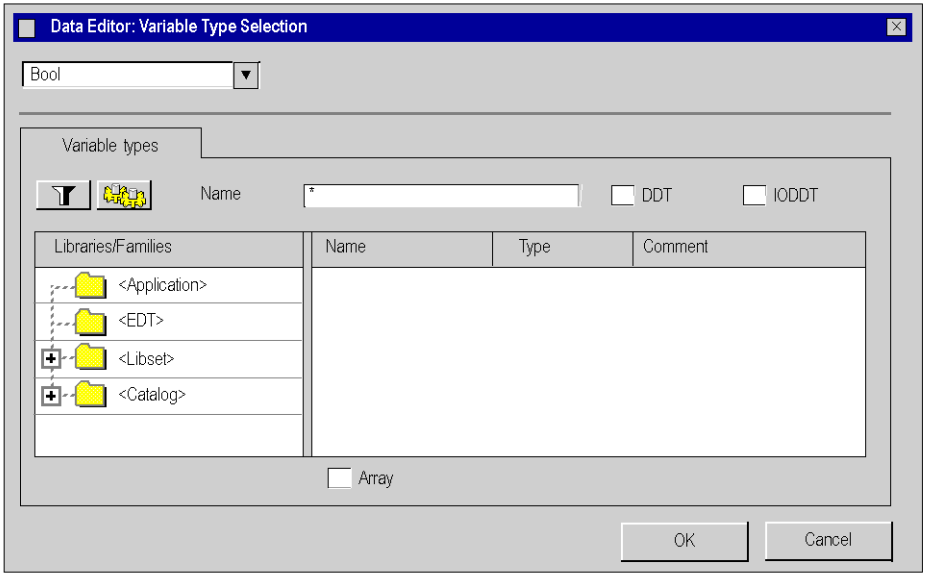
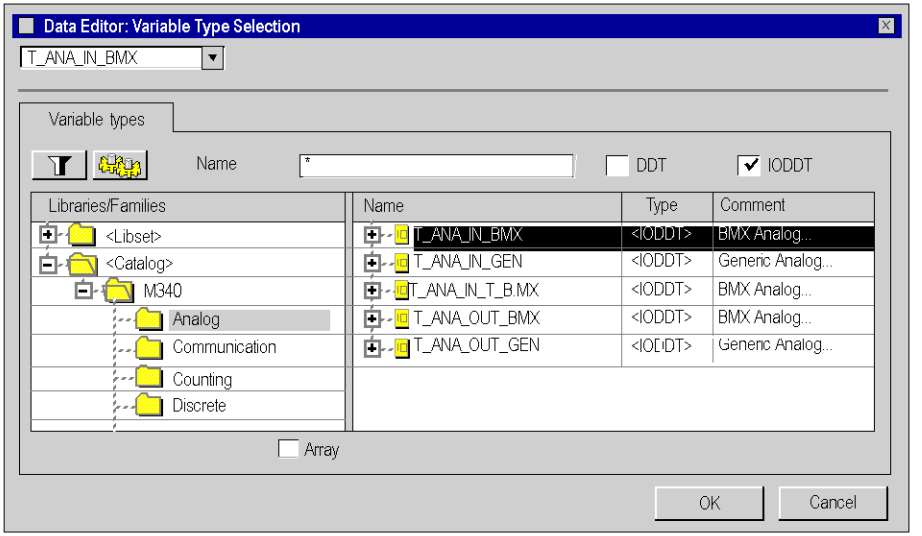
Creating an IODDT instance

Type IODDT instance

The data types (structures) have been **predefined by the manufacturer** and contain I/O language objects belonging to the channel of an application specific module (or the module itself).

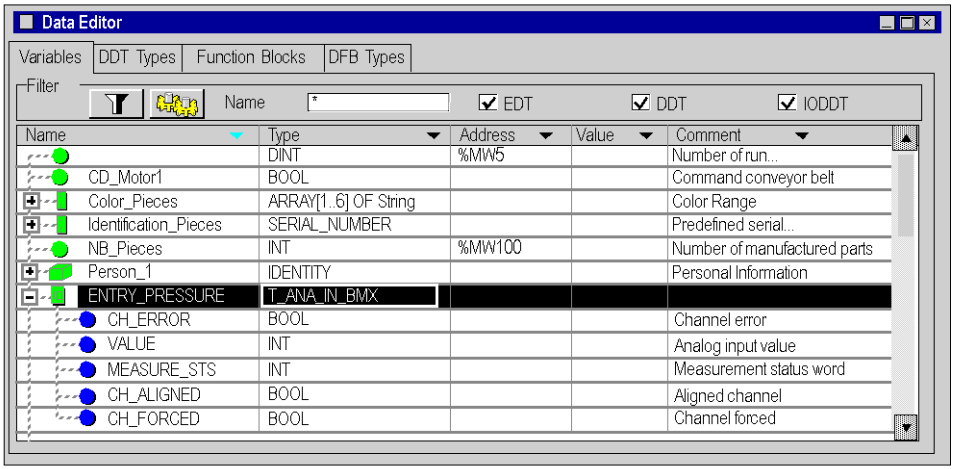
Perform the following steps to create an IODDT instance:

Step	Action
1	Select the **Variables** tab in the data editor.
2	Double-click on the empty cell **Name** (marked with an arrow) and enter the name of the instance.
3	Double-click on the corresponding **Type** field and click on it 

Step	Action
	. The following window appears. 
4	In the **Library/Family** area, display the **<Catalog>** index. The following window appears. 

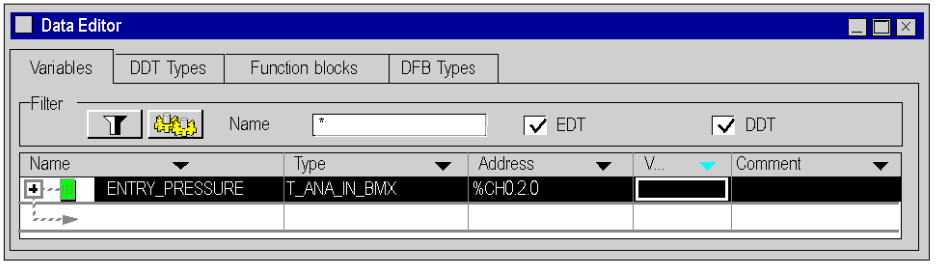
Step	Action
5	Select the Type of Controller, Function and the corresponding IODDT Type. Confirm with **OK**.

Example: Instance **ENTRY_PRESSURE** of Type **T_ANA_IN_BMX**:



Linking to the application specific module

Step	Action
1	Select the **Variables** tab in the data editor.
2	Enable the **IODDT** checkbox, which will display instances of Type IODDT.

Step	Action
3	Select the IODDT instance based on which the link will be created.
4	Enter the address of the module or its future storage location in the **Address** column (it is not necessary to configure the module during this phase). Example: 

Rules

Follow the rules below:

- An IODDT data type cannot be nestled in a DDT data type.
- An IODDT data type cannot be nestled in another IODDT data type.
- A public or private variable of a DFB cannot be an IODDT type.
- An input / output parameter of a DFB may be of Type IODDT. However, you can only use those IODDT types that are listed in the **Variable Type Selection** menu.
- An IODDT instance cannot be accessed from a DFB section.

Managing a Device DDT instance

Device DDT Instance

When adding a module to a rack, EcoStruxure Control Expert instantiates the associated device DDT structure. The main family of modules concerned by device DDT is the I/O modules family, some specific modules are also concerned. The device DDT name syntax is explained in the section on *Device DDT Naming Rule* (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).

Each module is associated with one implicit device DDT instance.

Implicit Device DDT instances are created by default on device insertion and refreshed automatically by the controller. They contain the modules status, modules and channels health bits, values of the modules inputs, values of the modules outputs, etc.

NOTE: Optional explicit DDT types are proposed in the data editor to support status or command data used by explicit exchanges on a channel of I/O modules in a remote I/O drop. Explicit DDT instances are created manually by the user in the data editor and used as input or output parameter with the function block managing the explicit exchange (see EcoStruxure™ Control Expert, I/O Management, Block Library).

Renaming a Device DDT

The following table describes a way to rename a module device DDT:

Step	Action
1	In the **Project Browser**, double-click the module that contains the device DDT to rename.
2	In the module window, select the **Device DDT** tab.
3	Select the module (not the channel).
4	Modify the device DDT name. **NOTE:** The total name length is limited to 32 characters. **NOTE:** A name needs to be unique in an application.
5	Validate the name changes in the advice window. This advice window appears when selecting another tab or view.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

- Verify that the EcoStruxure Control Expert configuration is consistent with the physical configuration.
- If you attempt to change the default names of your devices, manage the naming to prevent from addressing an unintended device.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Copying and Pasting a Module

During a module copy/paste action, the pasted module respects the default naming rule linked to its topological location.

Moving a Module

Moving a module does not modify its name.

If the same module type is inserted in the slot freed by the module shifting, the new inserted module should get the same topological name. In order to keep the naming uniqueness, *_rrrrrr* is added at the end of the string (*rrrrrr* being a random character series).

Replacing a Hot Standby Controller

Replacing a Hot Standby controller does not modify the device DDT name of the controller.

Checking the Device DDT Owner

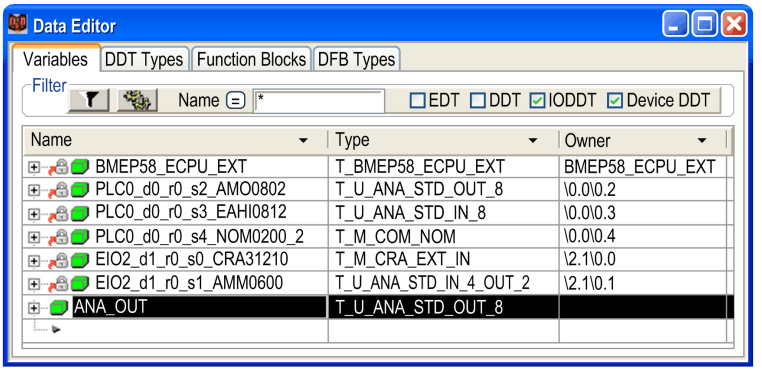
When a module is moved or a Device DDT is renamed, it may be difficult to locate the module in the architecture. This information is provided in the **Owner** column of the **Data Editor**. Displayed columns are configured, page 425 as needed.

Creating a Device DDT Instance

Perform the following steps to create a Device DDT instance:

Step	Action
1	Select the **Variables** tab in the data editor.
2	Double-click the empty cell **Name** (marked with an arrow) and enter the name of the instance.
3	Double-click the corresponding **Type** field and click the following button:  **Results:** The **Variable Type Selection** dialog box appears.
4	In the **Libraries/Families** area, display the **<Libset vx.y>** index. Where **x.y** indicates the EcoStruxure Control Expert software version.
5	Select the family (**I/O Management** for example) and the corresponding Device DDT type.
6	Confirm with **OK**.

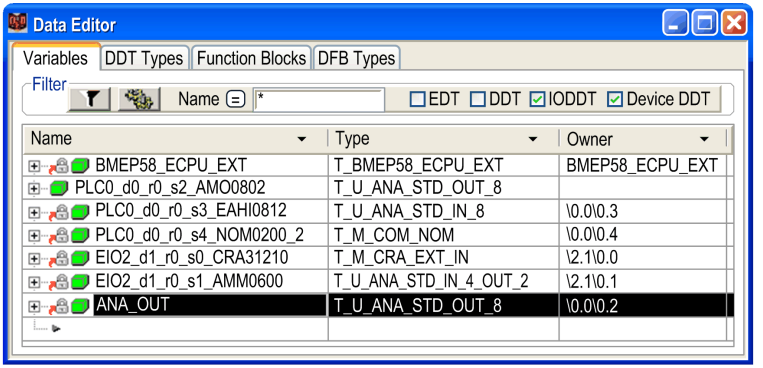
Example: Instance **ANA_OUT** of type **T_U_ANA_STD_OUT_8**:



Linking a Device DDT Instance to a Module

Use the following procedure:

Step	Action
1	Create a Device DDT as explained previously.
2	In the **Project Browser**, double-click the module to be associated to the created device DDT. **NOTE:** The created device DDT can only be linked to a module of the same type. For example, an 8 analog input module can only be linked to an 8 analog input device DDT.
3	In the module window, select the **Device DDT** tab.
4	Select the module (not the channel).
5	In the **Implicit device DDT > Name :** list, select the name of the device DDT created (instead of the current name).

Step	Action
6	Validate the name changes in the advice window. This advice window appears when selecting another tab or view. **NOTE:** Once a module is linked to a new device DDT, the variables that were linked to former device DDT name do not address the module anymore.
7	The previously created Device DDT instance is now linked to the module. It is displayed as follows in the data editor: 

NOTE: A variable linked to a module shows a lock in front of its name in the data editor.

Managing an IODDT, DDT or an Array type instance

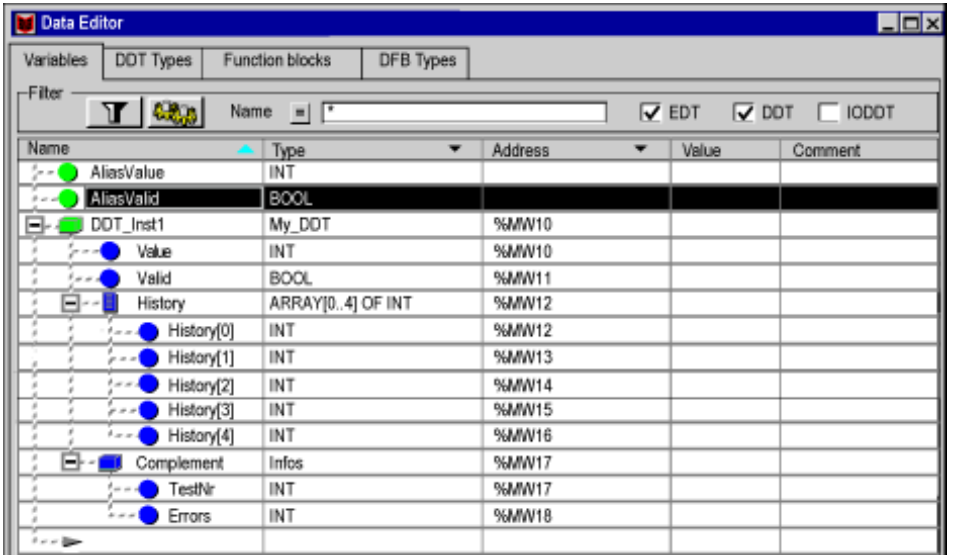
Adding an alias

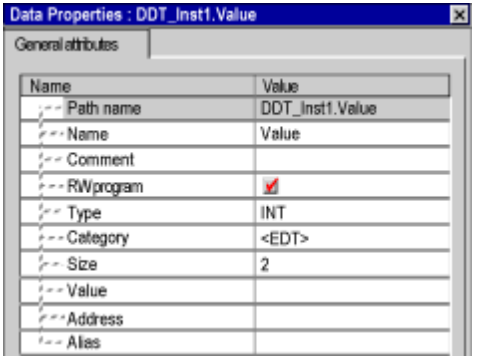
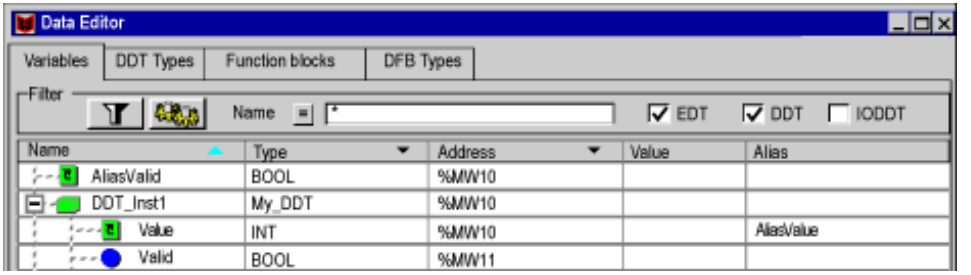
An alias name can be given to all elements of an IODDT, DDT or an Array type instance of application.

The aliases can be created with the following rules:

- The alias descriptions are only allowed for elements of type IODDT, DDT or an Array.
- The **type** and **address** for an alias cannot be changed as these attributes are defined by the IODDT, DDT or an Array
- **Name** and **Comment** can be changed.
- The alias displays the initial value of a DDT or Array element.

This procedure below is used for customizing elements used to make up a DDT instance, however the procedure is the same for an IODDT or Array instance.

Step	Action
1	Create one or more variable instances you want to customize that are identical with the elements of the IODDT / DDT / Array. For example **AliasValue** and **AliasValid**: 
2	Select an element of the **DDT_Inst1** instance (for example **Value**) that you want to assign to the previously created **AliasValue** instance. Alternatively, the user can also assign a new symbol as alias. In this case the alias variable is automatically be created with the type of the current element.

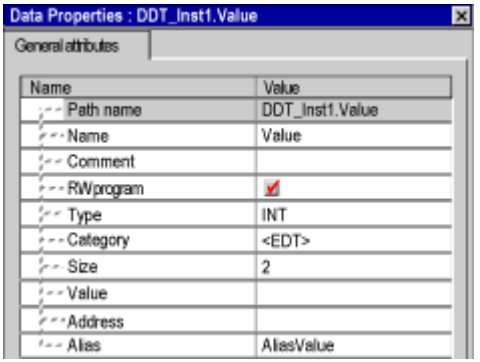
Step	Action
3	Using the shortcut menu select the **Properties** command. The **Data properties** field opens: 
4	In the **Alias** area enter the element **AliasValue** and then do the same for all elements that you want to assign. Along with the alias names and the elements of the DDT a symbol is shown. 

NOTE: The user can directly enter the AliasValue in the Alias column.

The "Data Properties" window

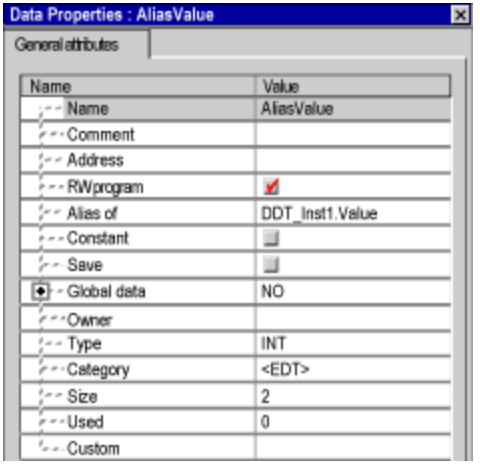
The **Data properties** window can be opened from the **IODDT structure**, **DDT structure** or an **Array** element. In such a case **Alias** is the structure attribute; it contains the names of the variables.

Example:

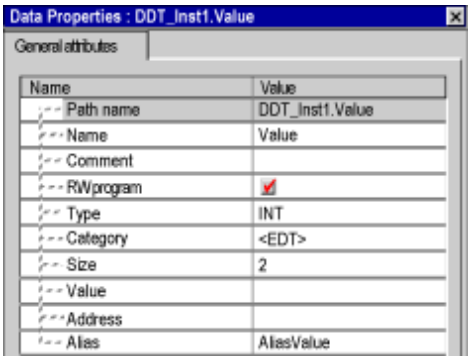
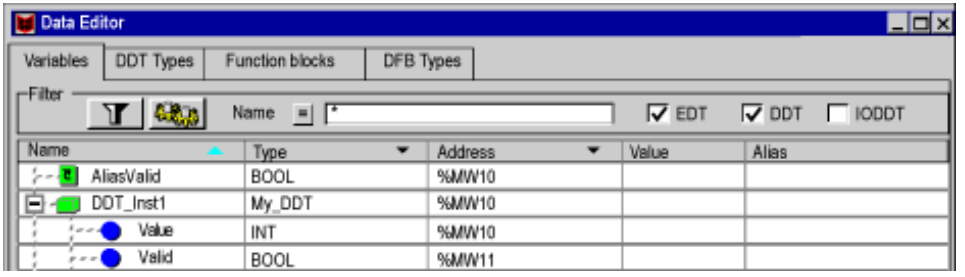


The **Data properties** window can also be opened from the **Alias instance**. In such a case **Alias of** is an instance attribute; it contains the names of the elements of the IODDT structure, DDT structure or an Array element.

Example:



Disconnecting an alias

Step	Action
1	Select the element from the appropriate IODDT structure, DDT structure or Array element.
2	From the shortcut menu select the command **Properties**, and the following dialog box will appear. Dialog box: 
3	Double-click on the target alias, delete the element in edit mode (AliasValue) and confirm. The symbols shown on the element disappear: 

NOTE: Alternatively, the user can directly delete the name in the column Alias or AliasOf.

NOTE: The alias instance returns to being a normal EDT instance and the attributes for the element to which the instance is assigned are retained. The element for the IODDT structure, DDT structure or an Array element contains the original manufacturer-defined name again.

Alias on sub fields of existing Alias

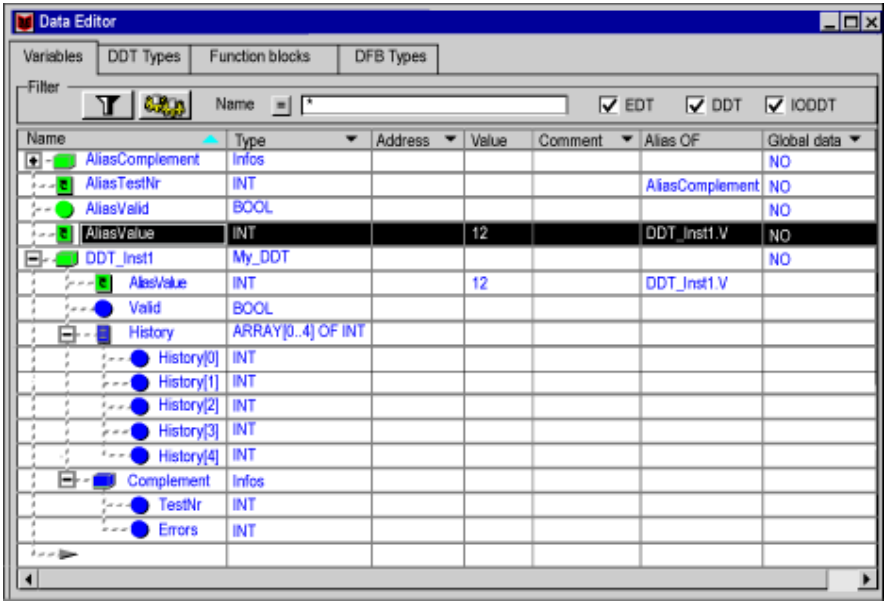
The user can set the alias on a sub field of an IODDT structure, DDT structure or an Array element that is already an alias.

Example: **AliasComplement** is alias of **DDT_Inst1.Complement** and **AliasTestNr** is alias of **AliasComplement.TestNr**.

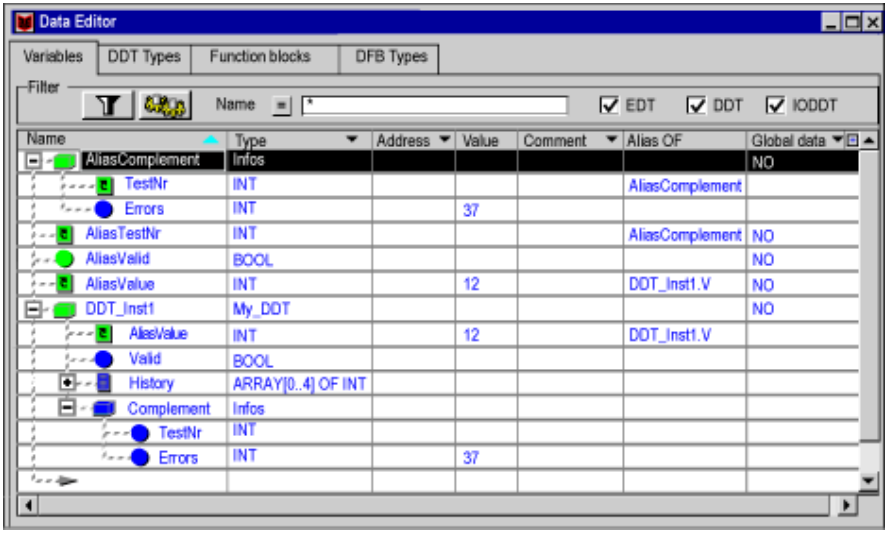
Management of initial value for an alias

The initial value of an alias is the initial value on the sub field of DDT or Array. This sub field path is given by attribute **Alias OF**. So when you get or set the initial value on an Alias in fact you get it or set it on the sub field path given by attribute **Alias OF**.

Example: Initial value of alias "AliasValue" is the initial value given on path "DDT_Inst1. Value". (12 in the following example screen)



Additionaly if we get or set initial value of “ DDT_Inst1.Complement.Errors” you will get or set initial value on “ AliasComplement.Errors”, because “ AliasComplement” is an alias on “ DDT_Inst1.Complement” (see following figure).



Creation of multiple variable instances

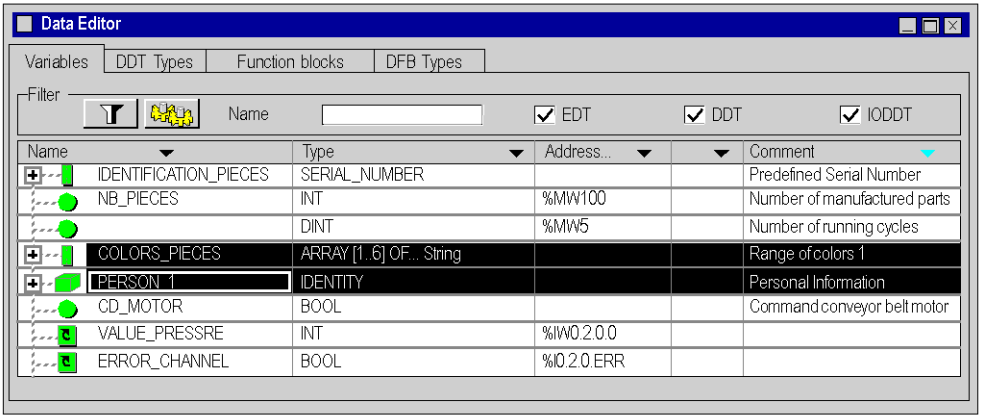
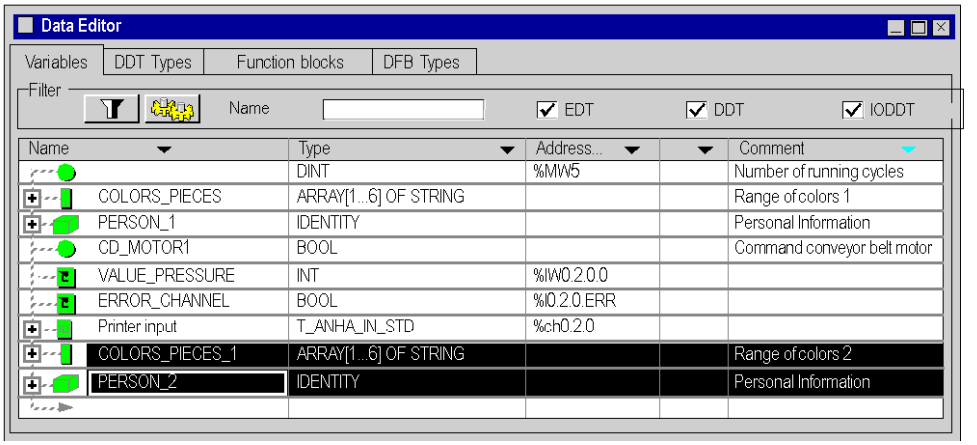
Procedure

There are two possibilities for this:

- using the **Copy/Paste** function,
- using the **Import** function.

Copying / pasting of instances.

The following procedure shows the method to create and rename an instance, following defined rules, using the **Copy/Paste** function:

Step	Action
1	Select the **Variables**tab in the data editor.
2	Select the desired instances (rows). 
3	Using the context menu select the **Copy** command.
4	Select an open position (row) from which the instances will be copied.
5	From the context menu select the command **Insert** and the new instances will be created and renamed. Example: 

NOTE: Renaming Rules:

- From a name of the instance **<Name>_<Number>** or **<Name>** results after renaming **<Name>_<Number+1>** or **<Name_0>**.

Importing instances

See "Importing Variables", page 1710.

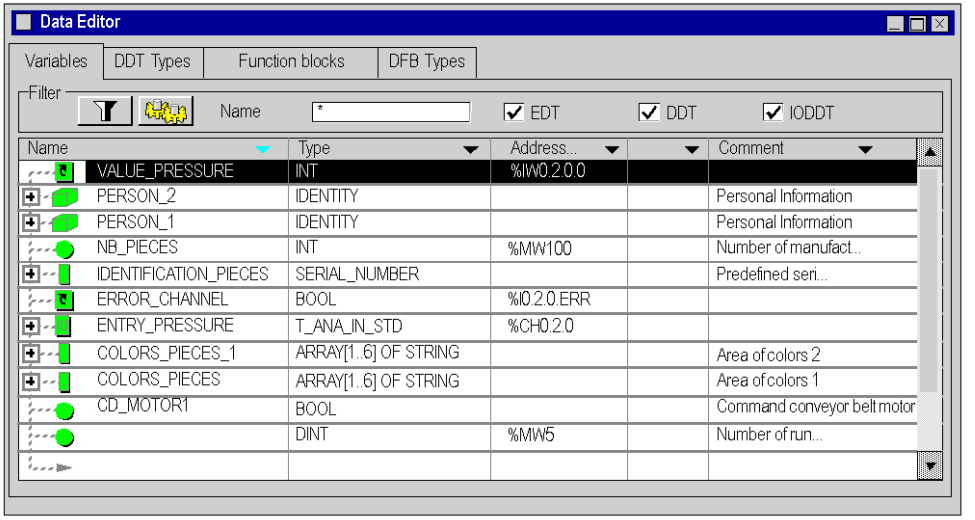
Displaying the variable instances in the editor

Display sequence in a column

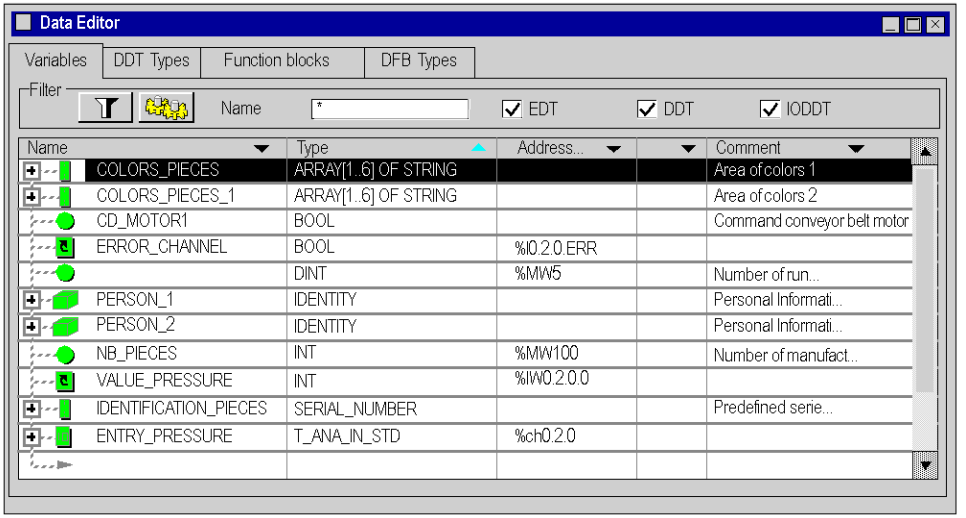
The variable instances can be displayed in **ascending** or **descending alphabetical order**. This sequence will be used in a column chosen by you.

For this, click on the area that contains the name of the column in question which determines the sorting order. After that the corresponding arrow will turn blue.

Sorting in descending alphabetical order in the column **Name**



Sorting in ascending alphabetical order in the column **Type**

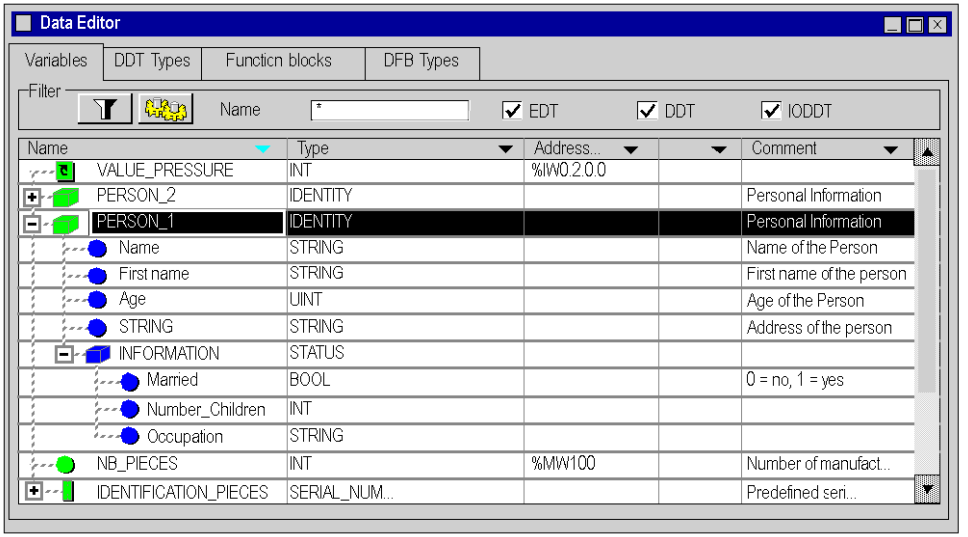


Display the elements of an instance of the "derived" type

To show the elements that belong to an instance of the "derived" type (structure or array), carry out the following steps:

Step	Action
1	Select the **Variables** tab in the data editor.
2	Enable the **DDT** check box.
2	Click on the **+** beside the instance of the desired structure or array type and this will then be shown on screen.
3	Run the operation again if you want to display an interlaced element.

Example: Display the instance `Person_1` of the structure type `IDENTITY`.

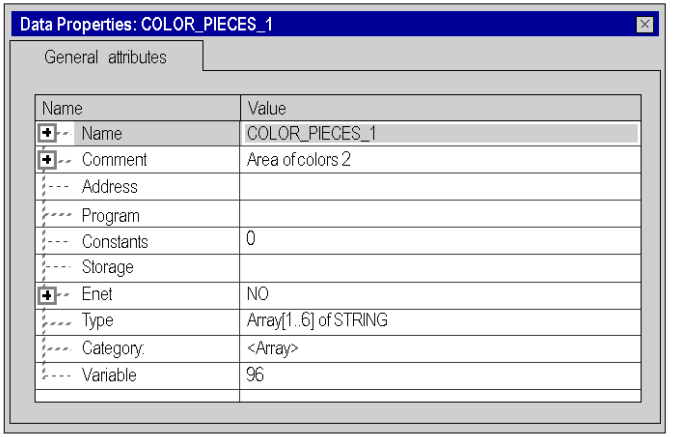


Displaying the instance attributes

For an**unopened** field **Data Properties**:

Step	Action
1	Select the **Variables** tab in the data editor.
2	Select the desired instance.
3	Select the **Properties** command from the shortcut menu and the attributes will be displayed in a window.

Example:



For a window that is already open **Data Properties**:

After the instance has been chosen, the window **Data properties** is updated automatically with the display for the attribute in question.

Sorting the variable instances

See Filtering of Data, page 433

Changing attributes of variable instances

Changing the name of an instance

Carry out the following steps:

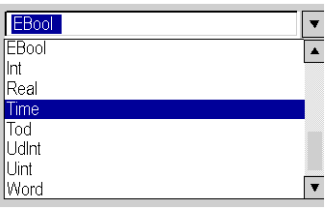
Step	Action
1	Select the **Variables** tab in the data editor.
2	Double-click on the instance name and enter the new name.

NOTE: The instance name can only be used once.

Changing the type of an instance

There are three options.

In the Data Editor select the **Variables** tab:

Double-click on the variable type and ...	**then...**
enter the desired type 	
use a Type from the list 	select the desired type
click the button 	select the appropriate type in the "Variables type selection", page 429 window

Changing the address of an instance

Carry out the following steps:

Step	**Action**
1	Select the **Variables** tab in the data editor.
2	Double-click on the instance address and enter the new address.

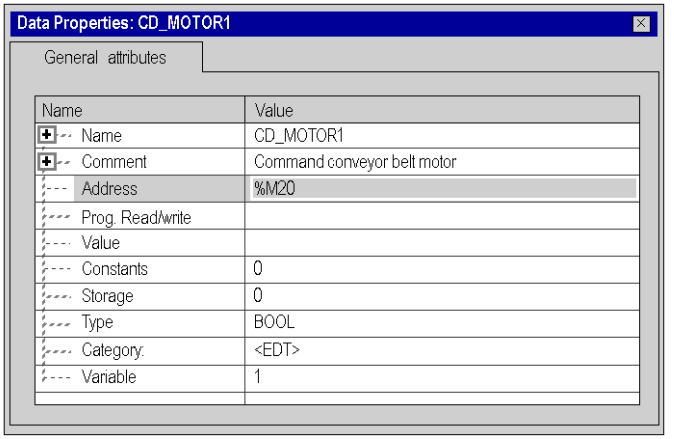
NOTE: Several different instances can be assigned to one address (multiple assignment).

Changing an attribute in the "Data Properties" field

Carry out the following steps:

Step	Action
1	Select the **Variables** tab in the data editor.
2	Select the desired instance.
3	Use the shortcut menu to select the **Properties** command.
4	In the **Data properties** field select the desired attribute.
5	Change the attribute.

Assigning an address for the CD_MOTOR1 instance (the variable is located):



Changing the attribute in several instances

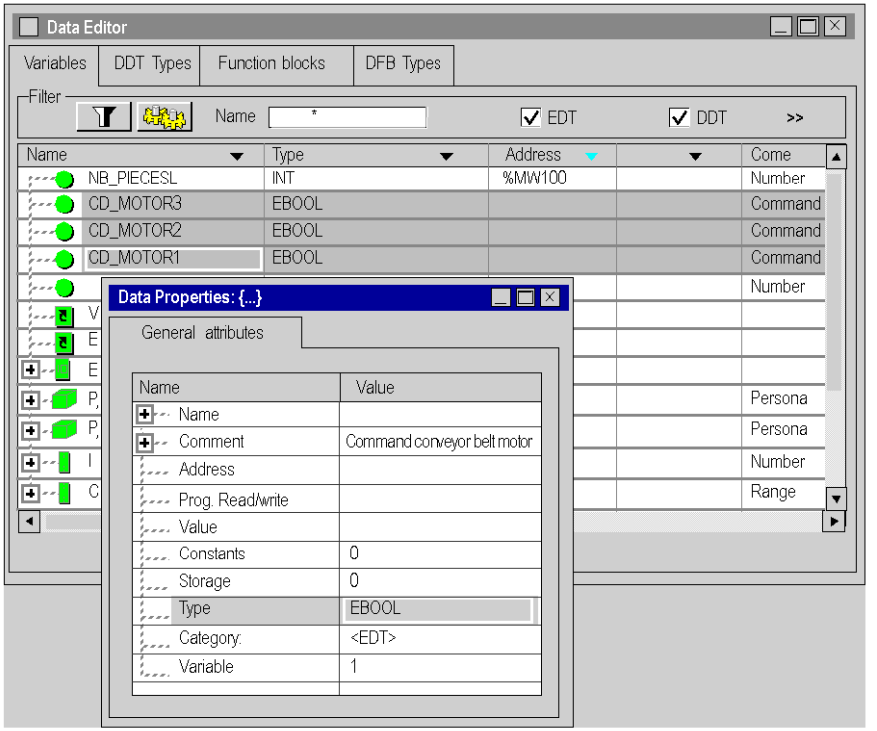
Carry out the following steps:

Step	Action
1	Select the **Variables** tab in the data editor.
2	Select the desired instances.
3	Use the shortcut menu to select the **Properties** command.
4	Select the desired attribute in the **Data Properties** window.
5	Change the attribute.

NOTE: For this functionality the following restrictions apply:

- names and addresses cannot be changed,
- the contents of the attribute to be changed must be the same for all selected instances,
- the change must be tolerated by all the instances selected,
- for sub-element instances with different superior elements, multiple selections are not possible.

Example: As can be seen in the following graphic, the three instances have the same type and can be changed from Bool to Ebool.



Rule for instances of the type Structure

The **internal** organization of an instance of type structure **(type, that belongs to the DDT family)** cannot be changed via the instance, but rather via the type of the instance **(DDT type tab)**.

In contrast, the changes listed above **(instance name, type, address)** are applied directly to the instances and do not affect the internal organization, only the instantiation of the type.

Creation of a Reference Type Instance

Reference Type Instance

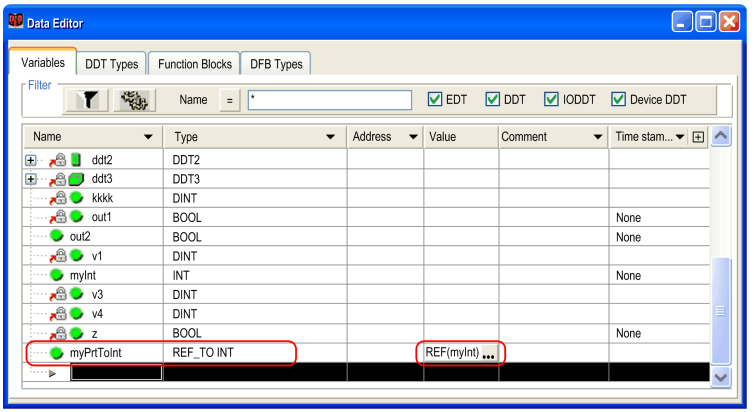
The reference type is used to create references to EDT, DDT and Device DDT variables.

Creating a Reference to a Variable

Creating a reference type instance:

Step	Action
1	Select the **Variables** tab in the **Data Editor**.
2	Double-click on an empty cell **Name** (marked with an arrow) and enter the name of the instance. Enter a comment (optional).
3	Double-click on the corresponding **Type** field and click  . The **Data Editor: Variable Type Selection** dialog opens.
4	Click the **REF_TO** checkbox.
5	Enable the **EDT**, **DDT** or **Device DDT** checkbox.
6	Select the desired type in the **Name** column and confirm with **OK**.
7	Fill out the **Value** column to initialize the variable by typing the name of the variable to be referenced.

Example of the **Data Editor** with a reference to a variable:



General Functions

What's in This Chapter

Configure the Data Editor Columns 425
Description of the Data Editor: Variable type selection..... 429
Filtering of Data 433
Exporting a subset of variables 437
Analyze DDT and DFB Data Types..... 438
Purge of Unused Instances..... 439
Local printing of data..... 441
Save the context of the Data Editor 442

Object of this chapter

This chapter guides the user through the use of the general features available in the data editor.

Configure the Data Editor Columns

Introduction

In every **Data Editor** tab you can configure the column display (a default configuration is suggested). This lets you:

- Define the columns you want to display.
- Define the sequence for the columns shown.

Available Columns

The following table shows the different columns available in the **Data Editor**:

Column Name		Description
Name		Name of the instance or data type.
Type		Data type. By default this attribute is selected.
Value		Initial value for the variables. By default this attribute is selected.
Comment		Comment for the variables. By default this attribute is selected.
Alias		User-defined name, which makes it possible to change an IODDT, DDT, or an Array element to a name defined by the manufacturer or the user. By default this attribute is selected. **NOTE:** An Alias value cannot be set for a variable if the Bit rank attribute has been set, page 360 for that variable.
Alias of		Complete path of the variable subfield on which you set the Alias attribute. By default this attribute is selected.
Constant		Kind of variable, the variable is a constant or not.
Custom		Free text.
Used		Number of times the variables are used in the program and in other parts of the project (animation tables, user windows).
Owner		Only for **Device DDT**, it displays the address of the modules that own this variable.
Address		Address in the controller for referencing the variables. By default this attribute is selected.
Global data		Shows if the variable is a variable that was swapped between different stations via the network.
+	**Group**	Number of the group (the variables) in which the variable is transported.
	ID	Number for the unique display of a variable in the group (the variables).
Save		Define if the variable accepts the mechanism for updating initial values with current values. **NOTE:** When different variables are mapped on the same *%MW* address, it is mandatory to set **Save** to one of the overlapping variables only.
HMI variable		When **General Project Settings > Data Dictionary > Only HMI variables** is selected, this reduces memory usage. By default this attribute is selected.
Time stamping		Select if a variable is time stamped or not and the kind of time stamping, see the System time stamping variables (see *System Time Stamping, User Guide*) or Applicative time stamping variables (see *EcoStruxure™ Control Expert, Applicative Time Stamping, User Guide*).
+	**Source**	Gives the source of the time stamping.
	TS ID	Gives the ID of the time stamped variable.
Scanned Variables		Attribute that you can use to tag a variable to be a scanned variable by the controller, through Ethernet for instance.

Column Name	Description
R/W Rights of Referenced Variable	Access rights of a reference type instance, used when the reference dereferences a read only variable, see the *Reference Access rights* (see *EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual*). By default this attribute is selected.
Data Protect	Use this attribute to help protect your data from modification request in monitoring mode. When **Data Protect** is selected, it is not possible to modify the variable. By default this attribute is de-selected. **NOTE:** The project setting **Data memory protect** must be enabled: **Tools > Project Settings > PLC embedded data**.
Exchange On STBY	Only for M580 Hotstandby systems (see *Modicon M580 Hot Standby, System Planning Guide for, Frequently Used Architectures*), if selected, the variable or the function block instance is automatically exchanged between Primary controller and Standby controller. By default this attribute is selected.
+ **Task**	Give the task in which a Variable, EFB instance, or DFB instance is used.
Retain	BME•586040 controllers present this variable attribute, which determines whether the variable value is kept after a warm start of the controller. If the attribute is: • Selected: Variable data is kept and is applied to the variable after a warm start. • De-selected: Variable data is discarded after a warm start; the variable value is reset. For other Modicon M580 controllers, this attribute is read-only. It is selected by default and cannot be de-selected.
no.	Order number of a function block.
984 Address	LL984 address of the variable (%Mx = 00000x, %Ix = 10000x, %IWx = 30000x, %MWx = 40000x) This column is only available for Quantum controllers (but not for Quantum safety-related controllers).
RW program	Access rights of the variable, read only or read and write by the program.
Descriptor x	Descriptors (1 to 9) of variables. **Descriptor 1** is displayed after selecting this column in the **Column Configuration** dialog box. **Descriptors 2** to **9** can be displayed using the **+** sign in the **Descriptor** column header.
Version	Type version.
Bit rank	Gives the bit rank of the variable in its structure type.
Nature	Only for the Program Unit data editor. Define if the Program Unit variable is an input, output or Input/output parameter, a public or private variable. By default this attribute is selected.
Effective Parameter	Only for the Program Unit data editor. Variable assignment to the program unit parameters (input, output and input/output parameters).

Defining Columns

Step	Action
1	Select the desired tab in the **Data Editor**.
2	Via the context menu, open the **Column Configuration** dialog box using **Customize Columns.**
3	Check the check boxes that correspond to the columns you wish to display.
4	Confirm the changes using the command **OK.**

NOTE: The **Name** and **Type** columns are required.

Description of the buttons

Button	Description
OK	Confirms the configuration and exits the dialog box.
Cancel	Discards the changes and exits the dialog box.
Default	Restores the default configuration.
Help	Opens the **Help** of the **Column Configuration** dialog box.

Change the Column Sequence

Step	Action
1	Select the desired tab in the **Data Editor**.
2	Via the context menu, open the **Column Configuration** dialog box using **Customize Columns**.
3	Place the cursor on the chosen attribute and move it to the desired location by clicking the two buttons to the right of the window.
4	Confirm by clicking **OK**.

NOTE: The first column must correspond with the attribute **Name**.

Description of the Data Editor: Variable type selection

Introduction

The **Data Editor: Variable Type Selection** provides the following options:

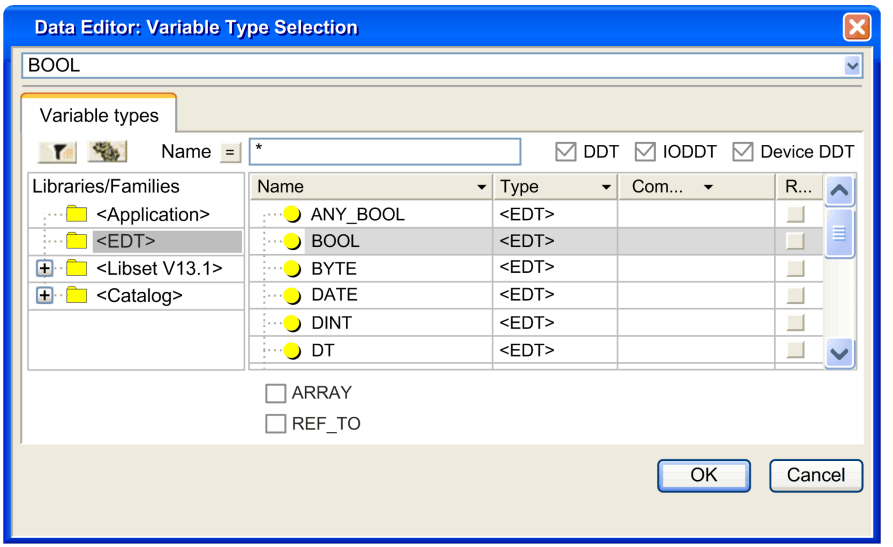
- Selection:
  - of derived data types (typical structures/arrays),
  - the function block data types (EFB\DFB),
- Creating non-typical arrays.

At a Glance

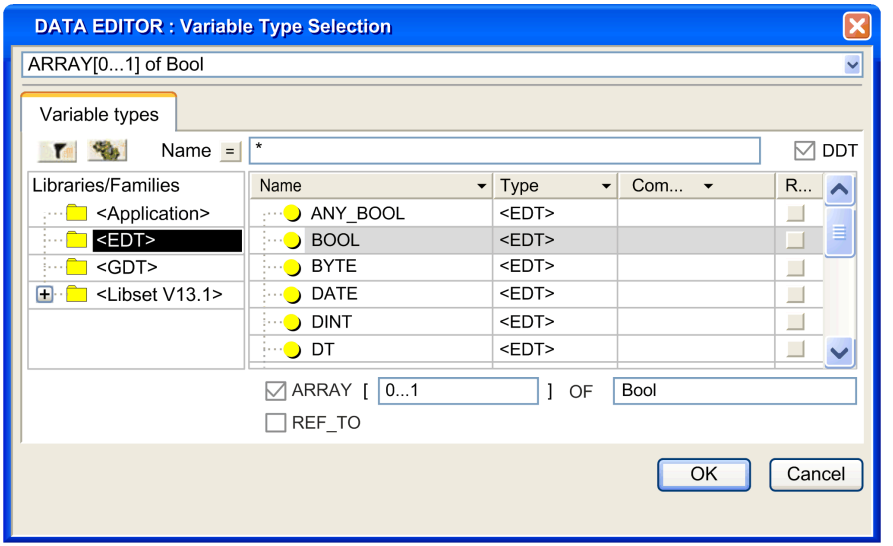
Data Editor: You can access the variables type selection by double-clicking on the **Type** column and clicking on  in the Data Editor.

Depending on the tab used to start the editor for the selection of the data type, one of the following windows is displayed:

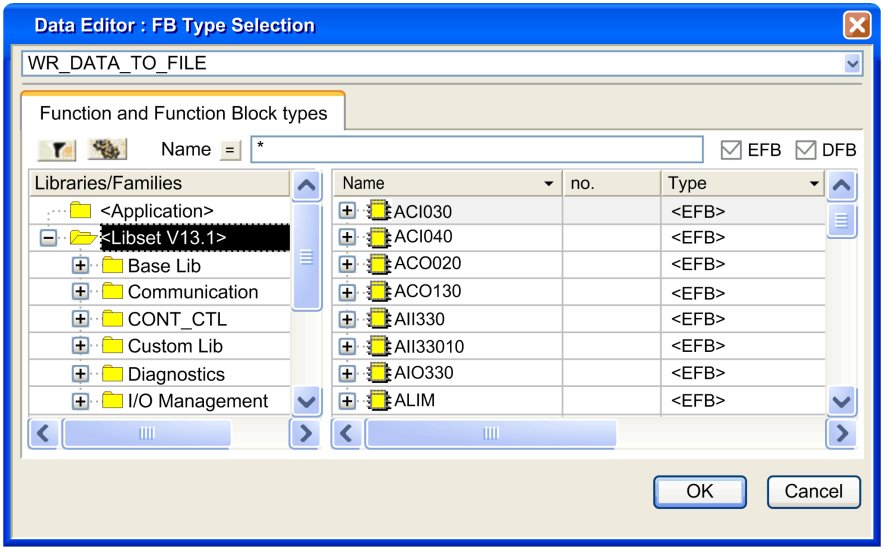
Variables:



DDT Types:



Function blocks:



NOTE: Access to the editor for the selection of the data types is not possible via the **DFB types** tab.

Description of the area Libraries/Families

This area contains directories and subdirectories where various selectable data types are classified.

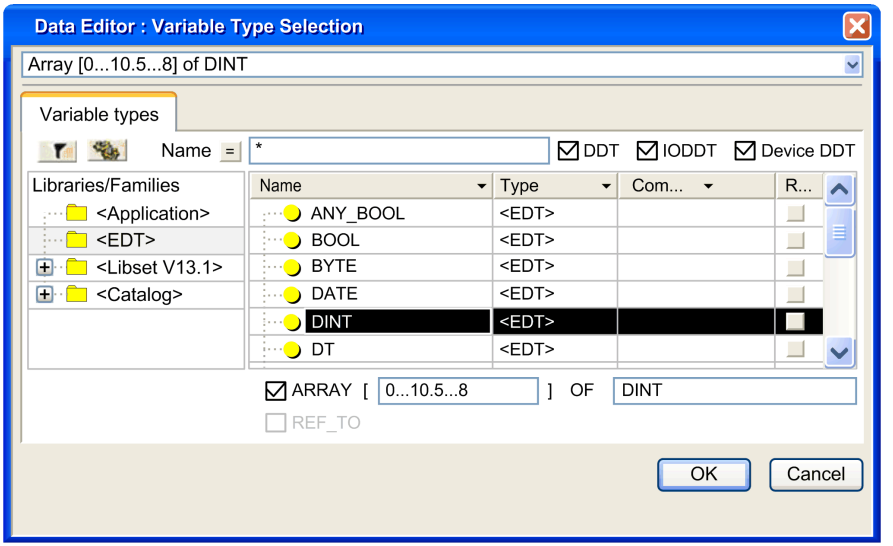
If you choose one of the directories or subdirectories, the types for the right-hand side of the columns Name/Type/Comment are shown.

- The directory **<Application>** shows the data types **used in the project**:
  - the Derived Data Types (DDT); the checkbox **DDT** must be enabled.
  - the Derived Function Block (DFB); the **DFB** checkbox must be enabled.
  - the Derived Data Types that refer to the inputs/outputs (IODDT); the **IODDT** checkbox must be enabled.
  - the data types for the elementary function blocks (EFB); the **EFB** checkbox must be enabled,
  - the Device Derived Data Types (DDDT); the **Device DDT** checkbox must be enabled.
- the **<EDT>** directory shows the types for the elementary data,
- the **<GDT>** directory shows the types for the generic data,
- the **<Libset>** directory that is organized into special libraries contains the **families made available by the manufacturer or the functions archived by the** user:
  - the Derived Data Types (DDT); the **DDT** checkbox must be enabled.
  - the data types for the sequence controls (SFC); the **DDT** checkbox must be enabled,
  - the data types for the derived function blocks (DFB); the **DFB** checkbox must be enabled,
  - the elementary function blocks (EFB); the **EFB** checkbox must be enabled,
  - the Device Derived Data Types (DDDT); the **Device DDT** checkbox must be enabled.
- the **<Catalog>** directory that shows the derived data types that reference inputs/outputs (IODDT) **from the manufacturer**; the **IODDT** checkbox must be enabled.

ARRAY checkbox

If this checkbox is enabled, you can use the editor to select the array types to have a single size or several sizes (maximum six dimensions).

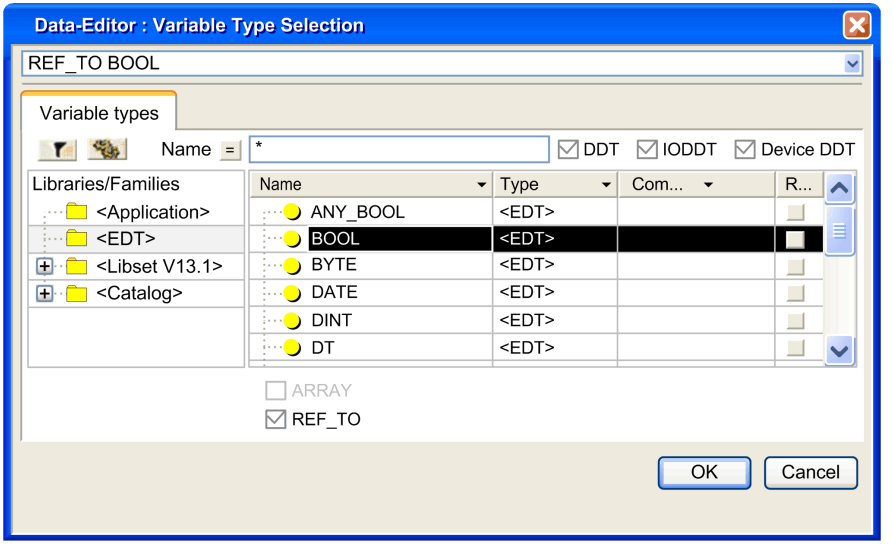
Example: Array with two dimensions 10x4 DINT



REF_TO checkbox

If this checkbox is enabled, you can use the editor to create a reference type instance.

Example: REF_TO BOOL



Filtering

You can create filters in the data editor to display the data types according to your requirements.

The **Name** field is used to enter the name of the data type that you want to display in the **Name/Type/Comment** area. You can use place holder symbols (* or ?) to carry out multiple search operations.

Clicking on the button  updates the display corresponding to the filter term defined in the field **Name**.

Clicking on this button  opens the **Filtering data**, page 434 dialog box where you can define various filter attributes.

Clicking on this button  inverts the filter. The button changes from **=** to **<>** and vice versa.

Filtering of Data

Introduction

You can create filters in the data editor to display the data according to your requirements.

A filter is the sum of the conditions applied to various attributes.

In every tab, it is standard for all variable instances or data types to be shown.

The various filters you have defined are saved when the data editor is closed and restored when it is opened again.

At a Glance

The filter parameterization in the data editor is determined depending on the tab chosen. Description of the Data editor different fields depending on the tab:

Area	In which table is it available ?	Purpose
Button 	All tabs	Clicking on this button updates the display corresponding to the filter term defined in the **name** field.
Button 	All tabs	Clicking on this button opens the **Filtering data** dialog box where you can define the conditions for the individual attributes.
Button 	All tabs	Clicking on this button inverts the filter. The button changes from **=** to **<>** and vice versa.
Name	All tabs	Field where the name (symbol) of the instance or the data type you want to display can be entered. You can use the place holder (* or ?) for this.
EDT	Variables tab	The enabled check box authorizes the display of the Elementary Data Types (EDT).
EFB	Function Blocks tab	The enabled check box authorizes the display of the Elementary Function Blocks (EFB).
DDT	Variables tab	The enabled check box authorizes the display of the Derived Data Types (DDT).
DFB	Function Blocks tab	The enabled check box authorizes the display of the Derived Function Blocks (DFB).
IODDT	Variables tab	The enabled check box authorizes the display of the Derived Data Types (DDT) that reference inputs/outputs (IODDT).
Device DDT	Variables tab	The enabled check box authorizes the display of the Derived Data Types (DDT).

The dialog box Filtering data

The dialog box **Filtering data** makes it possible to:

- view the current filter conditions,
- change the current filter conditions,
- define new filter conditions.

NOTE: The contents of the **Filtering data** dialog box depends on the tab from which it was called.

Description of the dialog box columns Filtering data

Description:

Column	Description
Name	This column contains the attributes for the (**Variables**, **DDT Types**, **Function Blocks** or **DFB Types**) tab used to open the dialog box. Check the box for the attributes that you want to filter.
Condition	The column allows you to define the filter condition for every attribute.
Inverted	Using this checkbox you can invert the filter condition for the attribute in question.

Options for defining a filter for **Variables**:

Name	Condition	inverted disabled	inverted enabled
Comment	Free text e.g. `*motor*`	All variables whose comment attribute contains the word `Motor` are filtered.	All variables whose comment attribute does not contain the word `Motor` are filtered.
Type	Free text e.g. `Int`	All variables of data type `Int` are filtered.	All variables not of data type `Int` are filtered.
Alias of	**Yes**	All variables defined as an alias are filtered.	All variables not defined as an alias are filtered.
Value	**User defined**	All variables defined with an initial value are filtered.	All variables not defined with an initial value are filtered.
Constant	**Yes**	All variables defined as constants are filtered.	All variables not defined as constants are filtered.
Custom	Free text e.g. `*HMI*`	All variables are filtered if their user-defined text (free text for use in an external tool or for special user requirements) contains the word `HMI`.	All variables whose user-defined text does not contain the word `HMI` are filtered.
Used	**>0**	All variables used in the project are filtered.	All variables not used in the project are filtered.
Owner	Free text e.g. `\2.1\*`	All variables which owner begins with `\2.1\` are filtered.	All variables which owner doesn't begin with `\2.1\` are filtered.
Address	Free text e.g. `%MW1`	All variables with the address `%MW1` are filtered.	All variables that do not have the address `%MW1` are filtered.
Global data	**Yes**	All variables defined as global data are filtered.	All variables not defined as global data are filtered.

Name	Condition	inverted disabled	inverted enabled
Save	**Accept**	All variables that accept saving and restore operations are filtered.	All variables that do not accept saving and restore operations are filtered.
HMI variable	**Yes**	All variables defined as HMI variable are filtered.	All variables not defined as HMI variable are filtered.
Time stamping	Free text e.g. `Both edges`	All variables which **Time stamping** attribute is `Both edges` are filtered.	All variables which **Time stamping** attribute is not `Both edges` are filtered.
Scanned Variables	**Yes**	All variables defined as Scanned Variables are filtered.	All variables not defined as Scanned Variables are filtered.
Exchange On STBY	**Yes**	All variables defined as **Exchange On STBY** are filtered.	All variables not defined as **Exchange On STBY** are filtered.
Retain	**Yes**	All variables defined as **Retain** Variables are filtered.	All variables not defined as **Retain** are filtered.

Options for defining a filter for **DDT types**:

Name	Condition	inverted disabled	inverted enabled
Comment	Free text e.g. `*Motor*`	All DDT types whose comment attribute contains the word `Motor` are filtered.	All DDT types whose comment attribute does not contain the word `Motor` are filtered.

Options for defining a filter for **Function Blocks**:

Name	Condition	inverted disabled	inverted enabled
Comment	Free text e.g. `*Motor*`	All function blocks whose comment attribute contains the word `Motor` are filtered.	All function blocks whose comment attribute does not contains the word `Motor` are filtered.
Diag	**Accept**	All function blocks that accept the attribute **Diag** are filtered.	All function blocks that do not accept the attribute **Diag** are filtered.
Type	Free text e.g. `TON`	All function blocks of type `TON` are filtered.	All function blocks not of type `TON` are filtered.
Custom	Free text e.g. `*HMI*`	All function blocks are filtered if their user-defined text (free text for use in an external tool or for special user requirements) contains the word `HMI`.	All function blocks whose user-defined text does not contain the word `HMI` are filtered.
Used	**>0**	All function blocks used in the project are filtered.	All function blocks not used in the project are filtered.

Name	Condition	inverted disabled	inverted enabled
Owner	Free text e.g. `\2.1\*`	All function blocks which owner begins with `\2.1\` are filtered.	All function blocks which owner doesn't begin with `\2.1\` are filtered.
HMI variable	**Yes**	All function blocks defined as HMI variable are filtered.	All function blocks not defined as HMI variable are filtered.
Exchange On STBY	**Yes**	All function blocks defined as **Exchange On STBY** Variables are filtered.	All function blocks not defined as **Exchange On STBY** are filtered.
Retain	**Yes**	All function blocks defined as **Retain** Variables are filtered.	All function blocks not defined as **Retain** are filtered.

Options for defining a filter for **DFB types**:

Name	Condition	inverted disabled	inverted enabled
Comment	Free text e.g. `*Motor*`	All DFB types whose comment attribute contains the word `Motor` are filtered.	All DFB types whose comment attribute does not contain the word `Motor` are filtered.
Diag	**Accept**	All DFB types that accept the attribute **Diag** are filtered.	All DFB types that do not accept the attribute **Diag** are filtered.
Protection	**Protected**	All protected DFB types are filtered.	All non-protected DFB types are filtered.

Exporting a subset of variables

Introduction

The chapter Import / Export, page 1686 explains how to export variables.

However, all variables and FB instances, and subsets, were always exported (EDT, DDT, IODDT, EFB, DFB).

In addition, a subset of variables may be filtered out and/or selected via the Data Editor and only this subset will then be exported.

Filter

Filter the variables as described in the chapter Filtering of Data, page 433.

Selecting

Click on a variable to select it in the Data Editor. You can select multiple variables by holding down the **Alt** or **Ctrl** keys.

Export

Export the filtered or selected variables via the context menu (right-click).

Analyze DDT and DFB Data Types

At a Glance

DDT (Derived Data Types) and DFB (Derived Function Block) will have one of the following status **after setup**:

- Status "in Process",
- Status "after the analysis".

This means that the data are instantiated with a type that is in process. The created instance has characteristics of the prior type that was analyzed prior to processing.

After the analysis, to the extent that the type has no errors, the applicable instances within the application will be updated..

Local analysis of data

Running the local analysis checks the selected data type for compatibility with the corresponding instances throughout the entire project.

For DFB types, each section of the type is analyzed separately..

Perform the following steps to start a local analysis:

Step	Action
1	Select the desired data type in the **DDT Types** or **DFB Types** tab.
2	Select the **Analyze Type**command from the shortcut menu.
3	Check the result of the analysis in the **Display Window** on the lower part of the window.

NOTE: Double-click on an error row in the display window to go to the source of the error.

Global analysis of data

Running the global analysis checks all selected data types for compatibility with the corresponding instances throughout the entire project.

Perform the following steps to start a global analysis:

Step	Action
1	Select the **Generate> Analyze Project** command in the menu.
2	Check the result of the analysis in the display window on the lower part of the window.

NOTE: Double-click on an error row in the display window to go to the source of the error.

Purge of Unused Instances

Overview

To purge unused instances of EDTs, EFBs and DFBs the Data Editor tabs provide the respective context menus.

Purge Unused Variable

The purge depends on the check boxes (EDT, DDT, IODDT and Device DDT), i.e. only the visible data (of checked types) will be purged.

Step	Action
1	Select the **Variables** tab of the Data Editor.
2	Right-click any column of the tab.
3	Select the menu command **Purge Unused Variables** from the context menu.
4	Confirm with **Yes**. **Result**: The unused data will be purged.

Purge DDT Types

NOTE: The purge feature is not available for DDT types.

Purge Unused FB Instances

The purge depends on the check boxes (EFB, DFB), i.e. only the visible data (of checked types) will be purged.

Step	Action
1	Select the **Function Blocks** tab of the Data Editor.
2	Right-click any column of the tab.
3	Select the menu command **Purge Unused FB Instances** from the context menu.
4	Confirm with **Yes**. **Result**: The unused data will be purged.

NOTE: When a FFB is not available for a kind of PLC (after a section import for instance) but still in the application (removed from the section but still in the list of the Application FFBs), the purge feature should be used before building the application.

Purge Unused Private Variables/DFB Types

From the context menu of the **DFB Types** tab two purge operations are accessible.

Step	Action
1	Select the **DFB Types** tab of the Data Editor.
2	Right-click any column of the tab.
3	Select one of the two menu commands **Purge Unused Private Data Instances** or **Purge unused Types** from the context menu.
4	Confirm with **Yes**. **Result**: The unused data will be purged.

Local printing of data

Introduction

Printing is a user-defined process and is executed using the following items:

- selected tab,
- Filtering variables,
- Sorting variables,
- Using certain variables

How to print

Carry out the following steps:

Step	Action
1	Select the user-defined adjustment of the printing process in the Data Editor: • Select the tab, • Select the filter, page 433: • Select the sorting order (alphabetically ascending or descending), • Using structured variables
2	Select the **File** > **Print** menu command or use the key combination **Ctrl+P** or use the toolbar icon  .

NOTE: Local printing takes the access security into consideration. Only information appearing in the window is printed.

NOTE: If an IODDT, a DDT or an Array element of an instance has an alias, the instance is displayed with the alias of the element as in the **Data Editor**.

Save the context of the Data Editor

Principles

When you leave the Data Editor, the configuration of the editor (visual aspect) will be saved. This includes:

- the size and position of the Editor window,
- the selected tab,
- the filter of the variables display,
- The configuration of the columns in each tab.

The following information will not be saved:

- An instance or variable type will be hidden when the Editor is reopened.
- The selection of single or multiple variables.

Communication

What's in This Part

Presentation of the communication editors 445
Network Configuration 448
Configuration of X-Way Routing Premium Stations 456

Subject of this Part

This part describes the communication editors.

Presentation of the communication editors

What's in This Chapter

Presentation of the communication editors of the project browser 445

Subject of this Chapter

This chapter describes the communication editors.

Presentation of the communication editors of the project browser

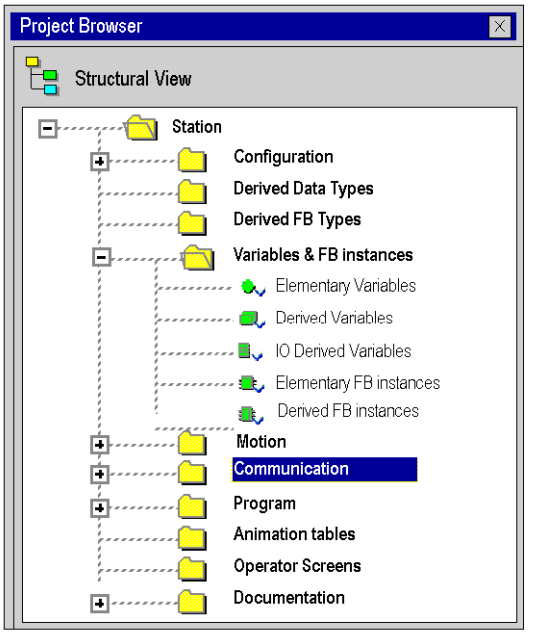
At a Glance

The communication editors allow you to configure and manage the different communication entities at project level.

They are accessible through the project browser by clicking the **Communication** tab.

Illustration

The figure below shows an example of the project browser



Network Configuration

What's in This Chapter

Network Configuration Principle Using Control Expert 448
Creating a Logic Network 449
Configuring a Logic Network 451
Associating a Logic Network with Network Hardware 452

Subject of this Chapter

This chapter presents the tools for configuring a network at the global level and at the station level.

Network Configuration Principle Using Control Expert

At a Glance

With Control Expert, the installation of a network takes place from the application browser and from the hardware configuration editor.

The method involves the following four steps:

- creation of a logic network,
- configuration of the logic network,
- declaration of the module or of the PCMCIA card (for Premium),
- association of the card or of the module with the logic network.

These four methods are presented further on in this documentation.

NOTE: The advantage of this method is that from the second step onwards, you can design your communication application (you do not need the hardware to start working) and use the simulator for functional testing of it.

NOTE: The first two steps are carried out in the project browser and the following two in the hardware configuration editor.

This manual introduces the method. For details of the various network configurations, please refer to the following documentation:

- Ethernet configuration for Premium (see Premium and Atrium Using EcoStruxure™ Control Expert, Ethernet Network Modules, User Manual), Ethernet configuration for M340 (see Modicon M340 for Ethernet, Communications Modules and Processors, User Manual), and Ethernet configuration for Modicon M580 (see Modicon M580, Hardware, Reference Manual),
- Modbus Plus configuration (see Premium and Atrium Using EcoStruxure™ Control Expert, Modbus Plus Network, User Manual),
- Fipway configuration (see Premium and Atrium using EcoStruxure™ Control Expert, Fipway Network, User Manual).

Creating a Logic Network

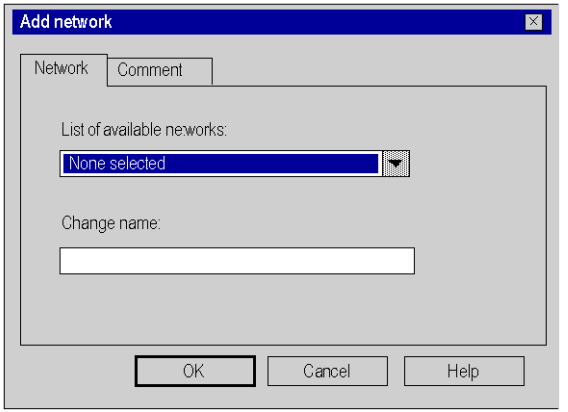
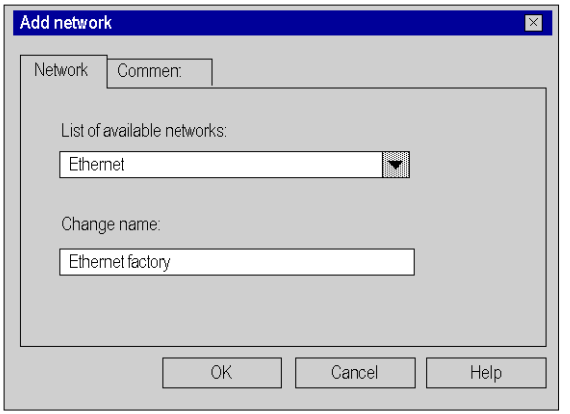
At a Glance

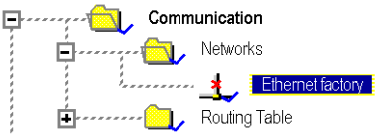
The first step in implementing a communication network is to create a logic network.

Creating a Logic Network

The following table describes how to create a network using the project browser.

Step	Action
1	Expand the *Communication* directory in the project browser. **Result:**: 
2	Right-click in the *Networks* sub-directory and select the **New network** option. **Result:**:

Step	Action
	
3	Select the network that you want to create from the list of available networks and give it a meaningful name. **Result:** Example of an Ethernet network:  **Note**: You can also add a comment, if you so desire, by clicking on the **Comment** tab.

Step	Action
4	Click OK and a new logic network is created. **Result:** We have just created the Ethernet network that appears in the project browser  **Note**: As you can see, a small icon indicates that the logic network is not associated with any PLC hardware. Furthermore, the small blue "v" sign indicates that the project needs to be rebuilt before it can be used in the PLC.

Configuring a Logic Network

At a Glance

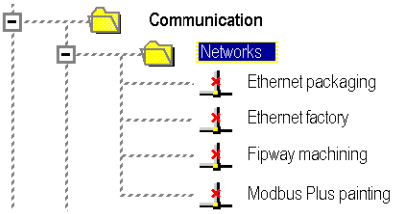
The second step in implementing a communication network is to configure a logic network.

This manual introduces the access to network configuration. For information on how to configure the various networks, please refer to the following documentation:

- Ethernet configuration for Premium (see Premium and Atrium Using EcoStruxure™ Control Expert, Ethernet Network Modules, User Manual), Ethernet configuration for M340 (see Modicon M340 for Ethernet, Communications Modules and Processors, User Manual), and Ethernet configuration for Modicon M580,
- Modbus Plus configuration (see Premium and Atrium Using EcoStruxure™ Control Expert, Modbus Plus Network, User Manual),
- Fipway configuration (see Premium and Atrium using EcoStruxure™ Control Expert, Fipway Network, User Manual).

Configuring a Logic Network

The table below describes how to access the configuration of a network from the project browser.

Step	Action
1	In the project browser, expand the directory tree under the **Networks** sub-tab located in the **Communication** tab of the tree directory to display all the project networks. **Example:**: 
2	Double-click the network you want to configure to obtain the network configuration window. **Note**: The windows differ according to the network family selected. However, for all networks, from this window it is possible to configure the Global Data, IPO scanning, Peer Cop utilities, common words, etc. **Note**: For Ethernet networks, an intermediate step is necessary, which involves selecting the family of the module that will be used in the hardware configuration.

Associating a Logic Network with Network Hardware

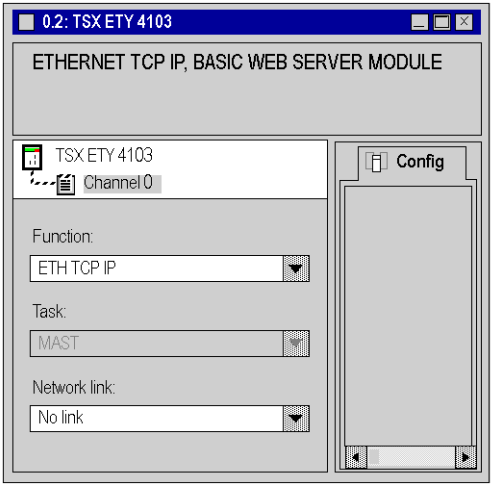
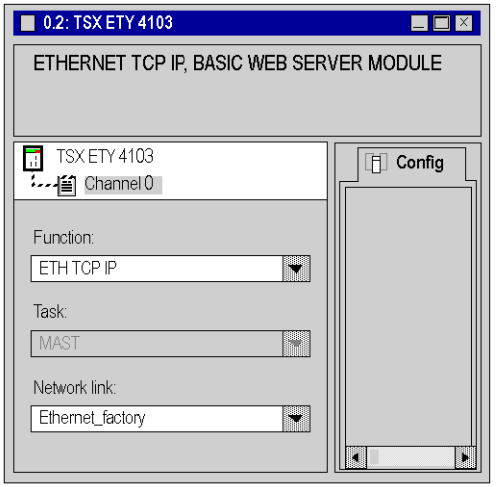
At a Glance

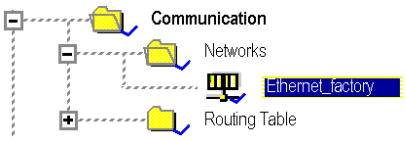
The final step in implementing a communication network is to associate a logic network with a network module, Modbus Plus card or Fipway card. Although the screens differ, the procedure is the same for each network device.

How to Associate a Logic Network

The following table describes how to associate a logic network to a network device declared in the hardware configuration editor.

Step	Action
1	Open the hardware configuration editor.
2	Right-click the device (Ethernet module, Fipway PCMCIA card or Modbus Plus PCMCIA card) that you wish to associate with a logical network.

Step	Action
3	Select the channel and function. **Result:** For a TSX ETY 4103 module: 
4	In the **Network link** field, select the network to be associated with the card. **Result:** 
5	Confirm your choice and close the window. **Result:** The logic network is associated with the device. The icon associated with this logic network changes and indicates the existence of a link with a PLC. Furthermore, the rack, module and channel numbers are updated in the logic network configuration screen. In our example we obtain the following project browser:

Step	Action
	 Communication Networks Ethernet_factory Routing Table

Configuration of X-Way Routing Premium Stations

What's in This Chapter

Configuration 457
Configuration of Multi-Network Services 458
Configuring an X-Way Router Module 459
Examples of X-Way Routing Stations 463
Examples of Partial Routing 466

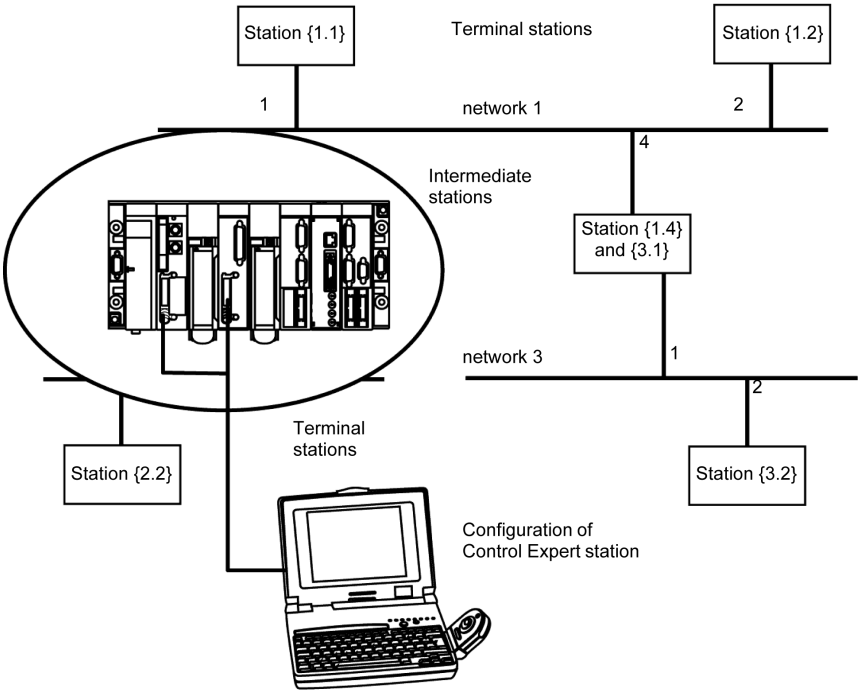
Subject of this Chapter

This chapter presents the operating modes required for configuring X-Way routing Premium stations.

Configuration

At a Glance

In an intermediate station, the management of several network couplers requires a configuration phase in order to distribute the functional characteristics to the various network entities.



NOTE: Multi-network routing information is constructed at the station level at the time of configuration of each bridge. No consistency check is done on routing data for the same network architecture.

Configuration of Multi-Network Services

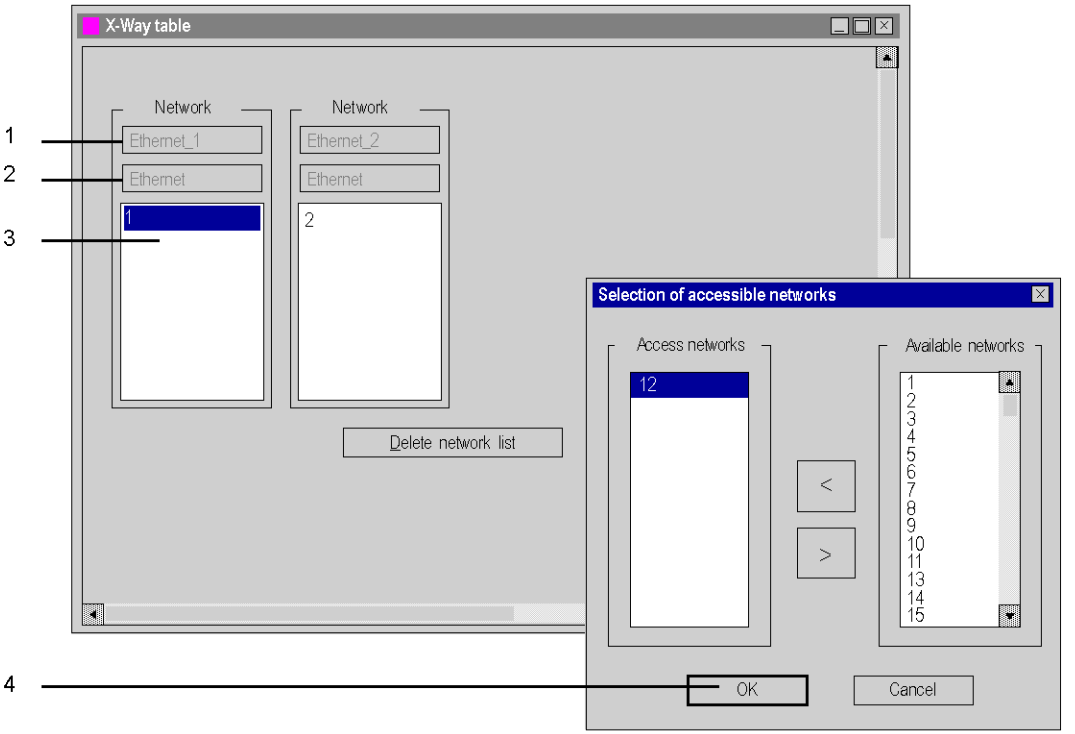
At a Glance

In a station that supports various network modules, each network connection point is considered as an address for the station. When configuring each module, it is necessary to define the list of network numbers that are accessible for each connection point.

Depending on the processor selected during hardware configuration, a bridge station can only manage 3 or 4 network modules. The table will therefore have a maximum of 4 elements.

Illustration

A specific screen allows entry of routing data for all the network modules of a station.



Elements and Functions

The following table describes the various zones in the configuration screen:

Label	Field	Function
1	Logical network	Used to display the logical network name.
2	Network type	Used to display the network type.
3	Accessible networks	Used: • for the unshaded **Logical Network** zone, to enter the list of networks accessible by this module, • for the shaded **Logical Network** zones, to display the list of networks accessible by these modules.
4	Available networks	Used to select the networks accessible by a module configured as a bridge. A list of numbers from 1 to 127 shows the networks available for a connection point. Each network number selected as being accessible is removed from the list of available networks in order to avoid configuration errors.

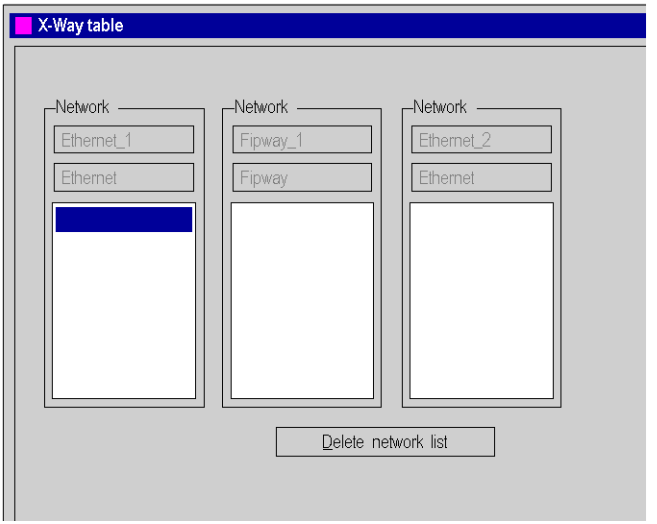
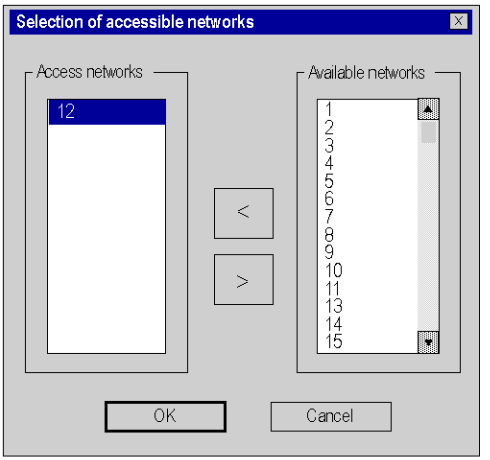
Configuring an X-Way Router Module

At a Glance

Before configuring the module as an X-Way router, the station's logical networks must be created.

Procedure

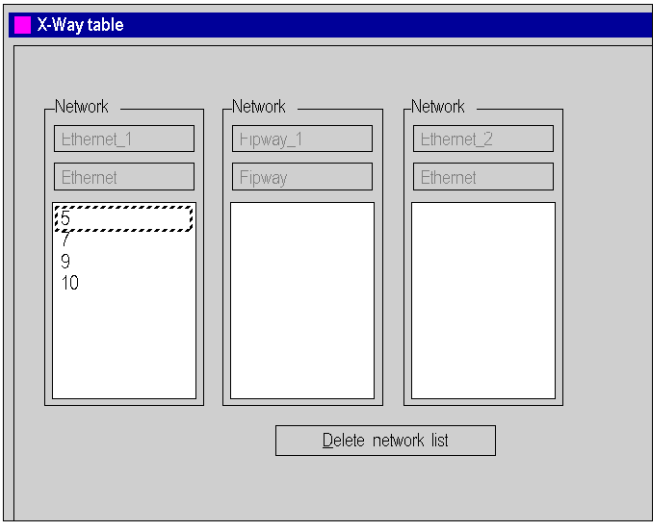
The following procedure is used to access, and then configure the station's module as an X-Way router.

Step	Action
1	Open the **Communication** tab in your project browser and in the **Routing table** tab, click the **X-Way table** tab. **Result**: The following window appears.  If the list of accessible module networks is empty, the window appears automatically (without double-clicking).
2	Double-click the highlighted field in order to configure the first network. **Result:** The **Selection of accessible networks** window appears. 
3	Double-click the number of the required network from the **Available networks** scroll list.

Step	Action
	Result: The network number is assigned in the **Access networks** scroll list.
4	Perform operation 3 as many times as necessary to define all the networks accessible by the module. Once finished, proceed to step 5.
5	Confirm the selection by clicking **OK**.
6	Confirm the configuration of the X-Way router by closing the window or clicking the **Enable** button in the toolbar.

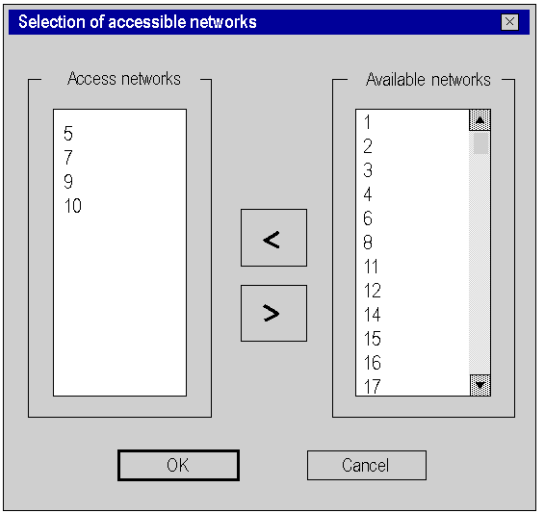
Removing the Bridge Function

It is possible to remove the bridge function from the module.

Step	Action
1	Access the following X-Way window. 
2	Click **Delete network list**.
3	Confirm the configuration

Removing Access to a Network

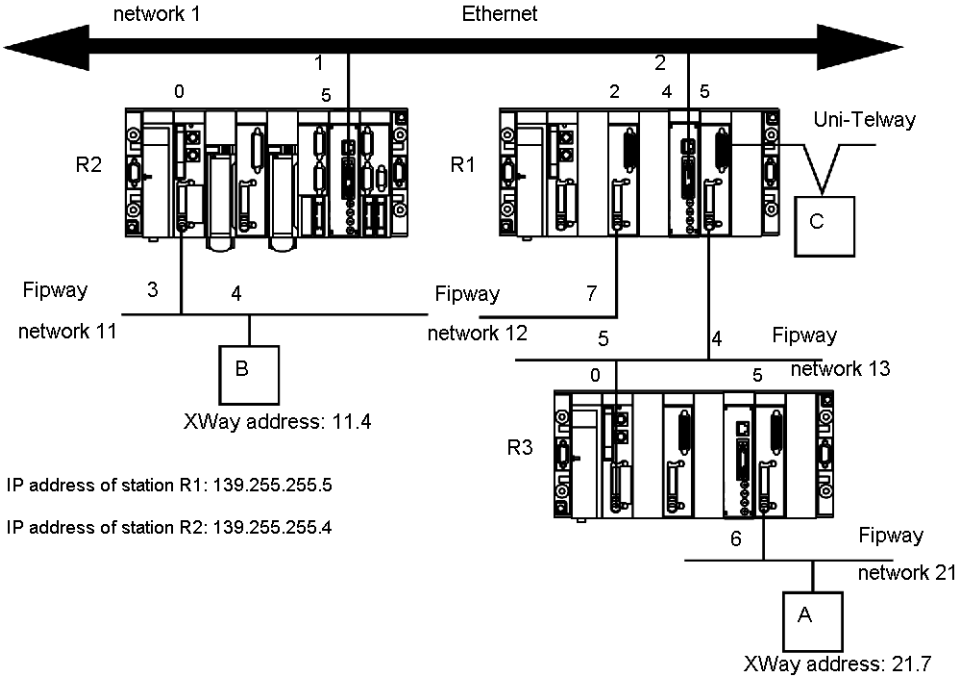
It is possible to remove access to a single network.

Step	Action
1	Access the following X-Way window of the accessible networks. 
2	Double-click the numbers of the accessible networks to be removed (left column). **Result:** The network number is reassigned in the **Available networks** scroll list.
3	Confirm the selection by clicking **OK**.
4	Confirm the router configuration.

Examples of X-Way Routing Stations

At a Glance

Each station must be configured in order to define the list of accessible networks.



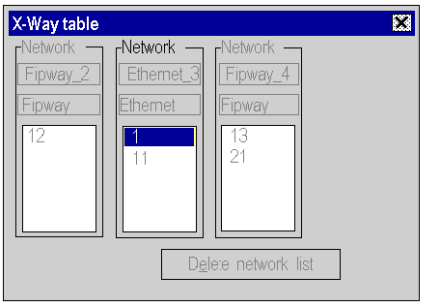
Configuration of Station R1

The module at slot 2 can only access network 12.

The module at slot 4 can only access network 1 and 11.

The module at slot 5 can access networks 13 and 21.

The bridge configuration of the station is therefore as follows:

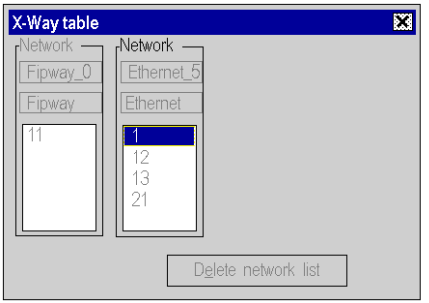


Configuration of Station R2

The module at slot 0 can only access network 11.

The module at slot 5 can access networks 1, 12, 13 and 21.

The bridge configuration of the station is therefore as follows:

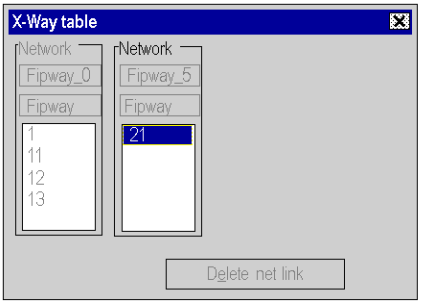


Configuration of Station R3

The module at slot 0 can access networks 13, 12, 1 and 11.

The module at slot 5 can only access network 21.

The bridge configuration of the station is therefore as follows:



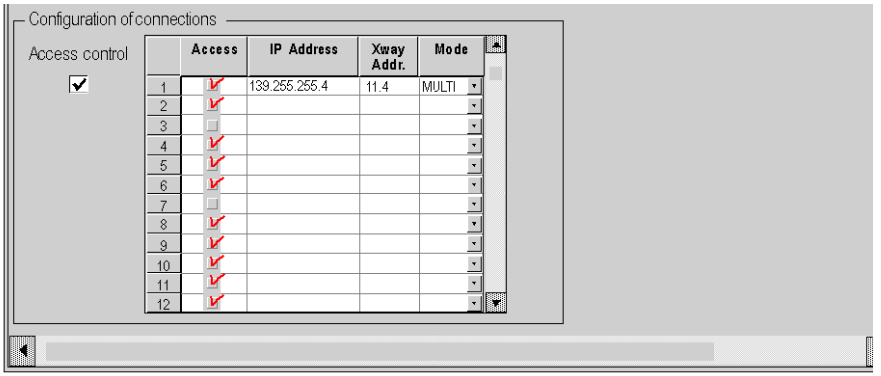
Messaging

To use the communication function (see EcoStruxure™ Control Expert, Communication, Block Library) such as Read_VAR for ethernet exchange between stations, configure the TCP/IP Messaging in the Ethernet network configuration (see Premium and Atrium Using EcoStruxure™ Control Expert, Ethernet Network Modules, User Manual) screen. In the **Messaging** tab, for each exchange set the IP address and target XWay address.

For example if station R3 needs to communicate with station B, in R1**Messaging** tab:

- set the R2 IP address (139.255.255.4) in the field **IP address**, and
- set the B XWay address (11.4) in the field **XWay address**.

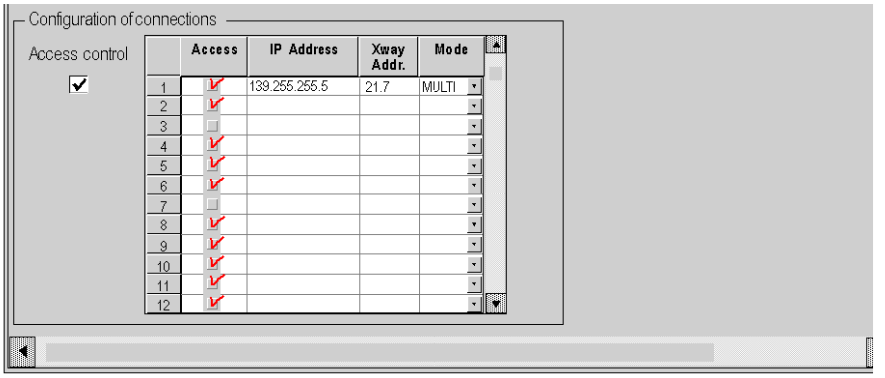
The following illustration shows the R1 **Messaging** tab:



For another example if station B needs to communicate with station A, in R2, **Messaging** tab:

- set the R1 IP address (139.255.255.5) in the field **IP address**, and
- set the A XWay address (21.7) in the field **XWay address**.

The following illustration shows the R2 **Messaging** tab:



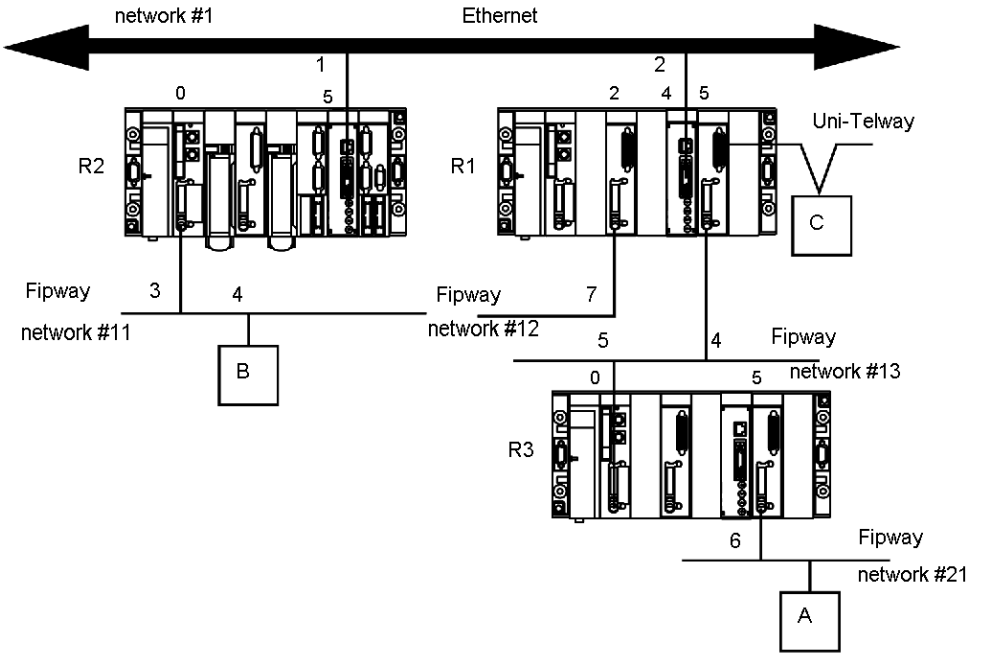
Examples of Partial Routing

At a Glance

When configuring a module as a bridge, it is possible to assign to it only a part of the available networks, instead of all of them. This selection is used to define a partial routing.

Illustration

Each station must be configured in order to define the list of accessible networks.



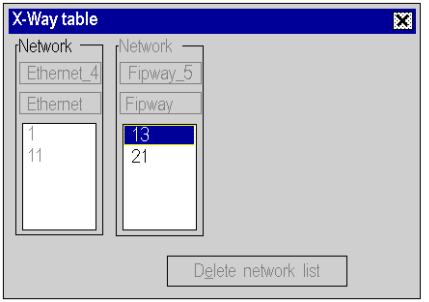
Configuration of Station R1

The module at slot 2 is not involved in the routing of data.

The module at slot 4 can access networks #1 and #11.

The module at slot 5 can access networks #13 and #21.

The bridge configuration of the station is therefore as follows:

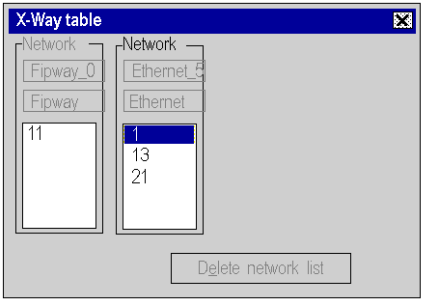


Configuration of Station R2

The module at slot 0 can only access network #11.

The module at slot 5 can access networks #1, #13 and #21. Network #12 is inaccessible.

The bridge configuration of the station is therefore as follows:

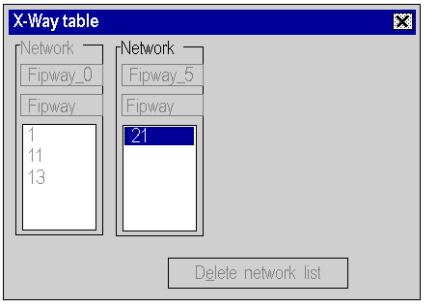


Configuration of Station R3

The module at slot 0 can access networks #13, #1 and #11. Network #12 is no longer accessible.

The module at slot 5 can only access network #21.

The bridge configuration of the station is therefore as follows:



Programming

What's in This Part

Accessing Tasks, Program Units and Sections 472
Control Expert Software Options 549
Common Function for graphical language 615
FBD Editor 651
LD Editor 745
SFC Editor 890
Common Function for Textual Language 1016
IL Editor 1051
ST Editor 1082
LL984 Editor 1120
Programming Languages Data Selection Dialog Box 1181
I/O Objects tab 1207
Memory Tab 1225
Type Library Browser 1231
Search/Replace Tool 1239
User's Function Blocks 1254

Subject of this part

This part describes the programming editors.

Accessing Tasks, Program Units and Sections

What's in This Chapter

Access to Tasks 472
Accessing Program Units 475
Accessing Sections 502

Overview

This section describes how to access tasks, Program Units and sections.

NOTE: Program Units are not available for Momentum, Quantum, Premium and Atrium.

Access to Tasks

Subject of this Section

This section covers the creation and modification of program tasks.

Create and Configuration of a Task

General

The first operation you need to perform to create an application program is to define the tasks.

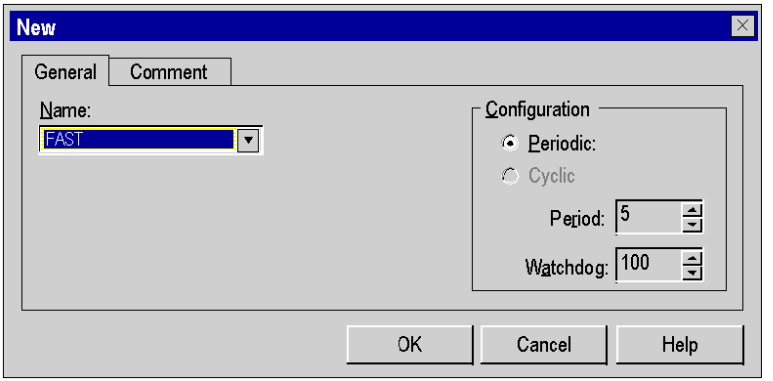
By default, only the master task is proposed. It is possible to create the following tasks:

the FAST task and the auxiliary tasks AUX0 to 3.

NOTE: For a fipio (see Premium and Atrium Using EcoStruxure™ Control Expert, Fipio Bus, Setup Manual) configuration, the window displayed is different.

Creating a Task

Carry out the following actions:

Step	Action
1	In the project browser, double-click on the program directory. The MAST directory appears in the **Tasks** directory.
2	Right-click on the Tasks directory, and then execute the **New task...** command from the contextual menu.
3	Click on **New task...**, and the following dialog box appears: 
4	Select the task: • FAST: Fast task • AUX 0, AUX1, AUX2 or AUX3: Auxiliary tasks (for processors with auxiliary tasks)
5	Select the type of scanning: • Periodic • or cyclic (for master task only)
6	Set the task period.
7	Set the **Watchdog** value which must be greater than the period value. We recommend to set it at twice the logic execution time.
8	The **Comment** tab can be used to add a comment to describe the role of the task for example.

Displaying and Modifying Task Properties

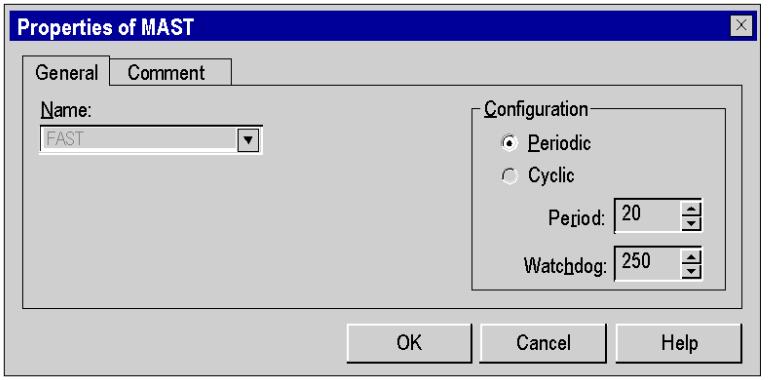
General

This procedure should be followed to display and/or modify the properties of an existing task.

NOTE: For a fipio (see Premium and Atrium Using EcoStruxure™ Control Expert, Fipio Bus, Setup Manual) configuration, the window displayed is different.

Task Properties

Carry out the following actions:

Step	Action
1	In the project browser, double-click on the program directory. The MAST directory (as well as the FAST and AUX directories if they have been created) appears in the **Tasks** directory.
2	Right-click on the MAST, FAST, or AUX directory, and then execute the **Properties** command from the contextual menu.
3	Click on **Properties**, and the following dialog box appears: 
4	Select the type of task: • Periodic • or cyclic
5	If you have selected a periodic task, set the task period.

Step	Action
6	Set the **Watchdog** value which must be greater than the period value. We recommend to set it at twice the logic execution time.
7	The **Comment** tab can be used to add a comment to describe the role of the task for example.

Accessing Program Units

Overview

This section describes accessing Program Units for Modicon M580 and M340.

Creating Program Units

Introduction

Create Program Units for Modicon M340:

- in the MAST and FAST tasks
- in offline and online mode

Create Program Units for Modicon M580:

- in the MAST, FAST, AUX0, and AUX1 tasks
- in offline and online mode

Creating a Program Unit

Creating a Program Unit:

Step	Action
1	Left-click the **Logic** directory of the desired task, or place the cursor using the arrow keys on this directory.
2	Click the right mouse button or use **Shift+F10** to select **New Program Unit...** in the context menu. **Result:** The **New** dialog box for the Program Unit opens.
3	Enter the name of the Program Unit. It must be unique in the entire project and conform to the general naming conventions.

Step	Action
4	Select the type of access protection if necessary from the **Protection** list box: • **None** No protection • **Read only** Write protection • **No read & write** Read and write protection See also section Project Protection, page 192.
5	Select the functional module to which the section is allocated from the **Localization** tab if necessary. See also section Functional module, page 237.
6	Define a variable of BOOL or EBOOL type in the **Condition** tab with which the Program Unit can be activated/deactivated (section active when variable=1) if required. You have the following options to select variables: • You can enter the name of the variable/address directly or copy it from the clipboard. • Use the command button **...** to open an instance selection dialog box, page 1181. Only for EBOOL variable, activate the **Can be forced** check box when you want to force the variable.
7	Enter a comment in the **Comment** tab if desired.
8	Confirm the entries with **OK** and the Program Unit is automatically created in the **Logic** directory with the following structure: • **Interface & Variables** • **Sections** folder • **Animation Tables** folder You can refer to the chapter *Animation Tables*, page 1473, which refers to the global project, for the implementation of animation tables in the Program Unit. **NOTE:** For the animation tables the accessible variable scope is limited to the Program Unit scope.
9	Declare the parameters and variables for the Program Unit.
10	Create sections in FBD, LD, IL, ST, page 502 or SFC, page 510.

Displaying or modifying Program Unit properties

Displaying or modifying Program Unit properties:

Step	Action
1	Left-click on the desired Program Unit, or place the cursor on this Program Unit with the arrow keys.
2	Select (right-click or **Shift**+**F10** key combination) the **Properties** menu command from the shortcut menu. **Result:** The **Properties of** dialog box for the Program Unit opens.
3	Implement the desired changes.
4	Confirm all entries with **OK**.

Importing/exporting Program Units

See:

- Importing programs, page 1701
- Exporting programs, page 1699

Properties Dialog Box for Program Units

Structure of the properties dialog box

The properties dialog box consists of four tabs:

- **General**

  Enter the name and the type of access protection of the Program Unit in this tab.
- **Localization**

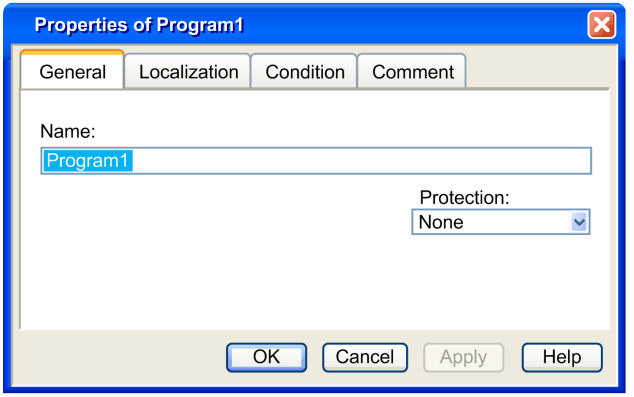
  You can enter the functional module of the Program Unit in this tab.
- **Condition**

  You can enter a BOOL or EBOOL variable which can enable/disable the Program Unit in this tab.
- **Comment**

  You can enter a comment in this tab.

General tab

Representation of the **General** tab:

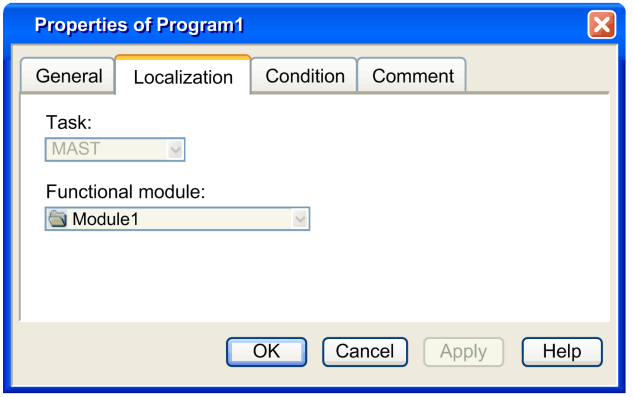


Elements of the **General** tab:

Element	Description
Name	Enter the name of the Program Unit in this text box. The name must be unique in the entire project and conform to the general naming conventions.
Protection	Select the type of access protection from this list box. • **None** No protection • **Read-Only** Write protection • **No read & write** Read and write protection See also *Program Unit, Section and Subroutine Protection*, page 192.
OK	Use this command button to accept the entries and close the properties dialog box.
Apply	Use this command button to accept the entries without closing the properties dialog box.

Localization tab

Representation of the **Localization** tab:

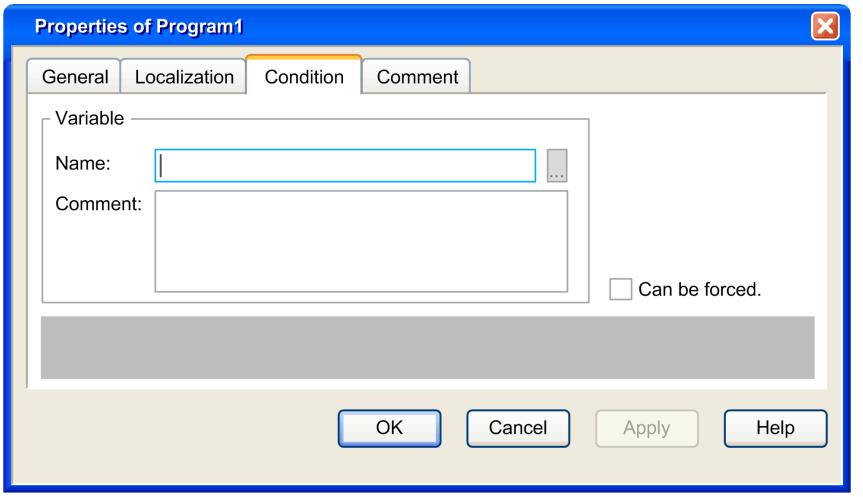


Elements of the **Localization** tab:

Element	Description
Task	The task of the Program Unit is displayed in this field.
Functional module	Select the functional module to which the Program Unit is to be allocated to in this list box. For detail, refer to chapter *Functional Modules*, page 237.
OK	Use this command button to accept the entries and close the properties dialog box.
Apply	Use this command button to accept the entries without closing the properties dialog box.

Condition tab

Display of the **Condition** tab:



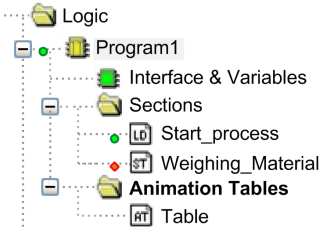
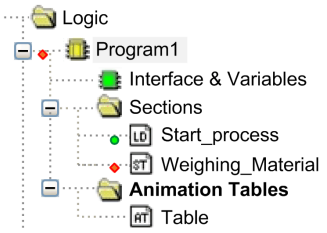
Elements of the **Condition** tab:

Element	Description
Variable	You can enter variable of BOOL or EBOOL type which can conditionally enable/disable the Program Unit in this tab. The scope to select the variable is a either a global variable, or a public variable or an output parameter of another Program Unit. In this area, you can enter the following: • **Name** ◦ You can directly enter the name of the variable/address or paste it from the clipboard. ◦ Use the command button **...** to open an instance selection dialog box, page 1181. • **Comment** The comments about the selected variables are displayed in this text box.
Can be forced	This check box indicates whether the selected variable can be forced or not.
OK	Use this command button to accept the entries and close the properties dialog box.
Apply	Use this command button to accept the entries without closing the properties dialog box.

The animation of the activation condition behavior is provided in the project browser. The Program Unit and its sections have activation conditions, page 508.

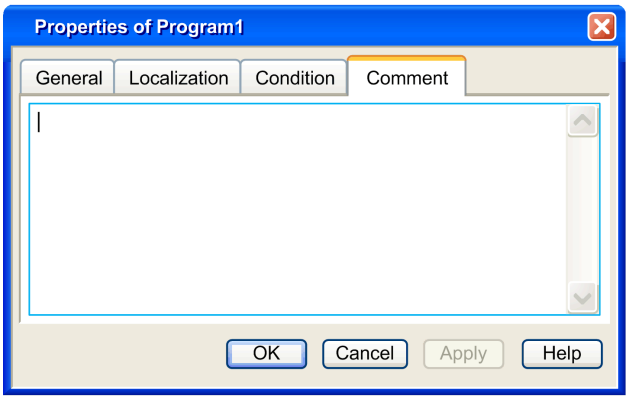
In the project browser, the flag (green for activate, red for inactivate) of the Program Unit and the sections reflect the variable value of the variable used for their activation.

A section under a Program Unit is executed only if the section is activated and the Program Unit is activated.

Case:	Description
	The section `Start_process` is executed and the section `Weighing_Material` is not executed.
	The `Program1` is not activated, `Start_process` and `Weighing_Material` sections are not executed.

Comment tab

Representation of the **Comment** tab:



Elements of the **Comment** tab:

Element	Description
Text box	A comment about the Program Unit can be entered in this text box.
OK	Use this command button to accept the entries and close the properties dialog box.
Apply	Use this command button to accept the entries without closing the properties dialog box.

Program Unit Data Scope

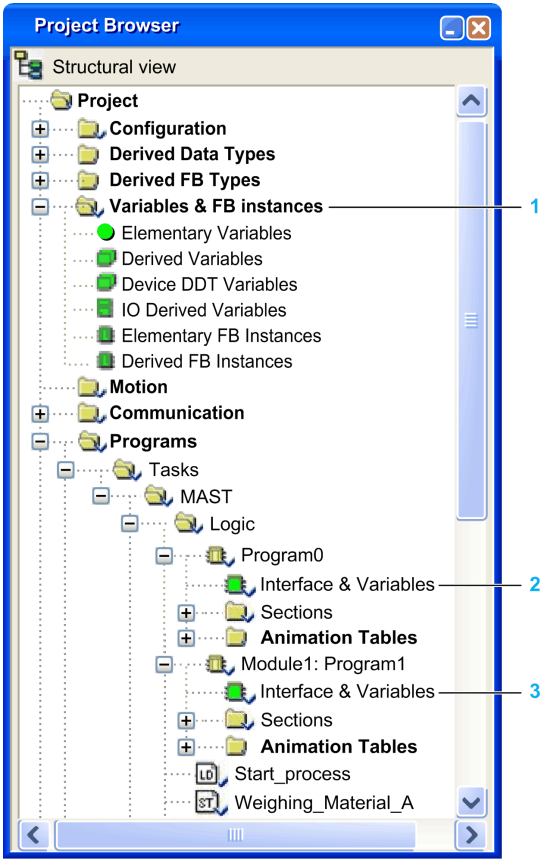
Introduction

When Program Units are allowed in programming your application (Modicon M580 and M340), variables are determined by the scope to which they belong to:

Global variables:	They refer to the application wide - or global - scope of data in a project which are: • Unlocated variable • Located variable (mapped at %M flat address)
Program Unit variables:	They refer to the Program Unit data scope and have one of the following attributes: • **Private**: can only be R/W in the scope of this Program Unit. • **Public**: can be R/W out of the scope of this Program Unit. • Parameter (**Inputs**, **Outputs**, and **Inputs/Outputs**): linked to Public variables (from other Program Units) or Global variables. • **External**: Global variables used in this Program Unit.

NOTE: The management of topological variable is only supported in the data editor, refer to parameter assignment, page 492 to map a variable of a Program Unit to a topological variable.

Data Scope in Project Browser



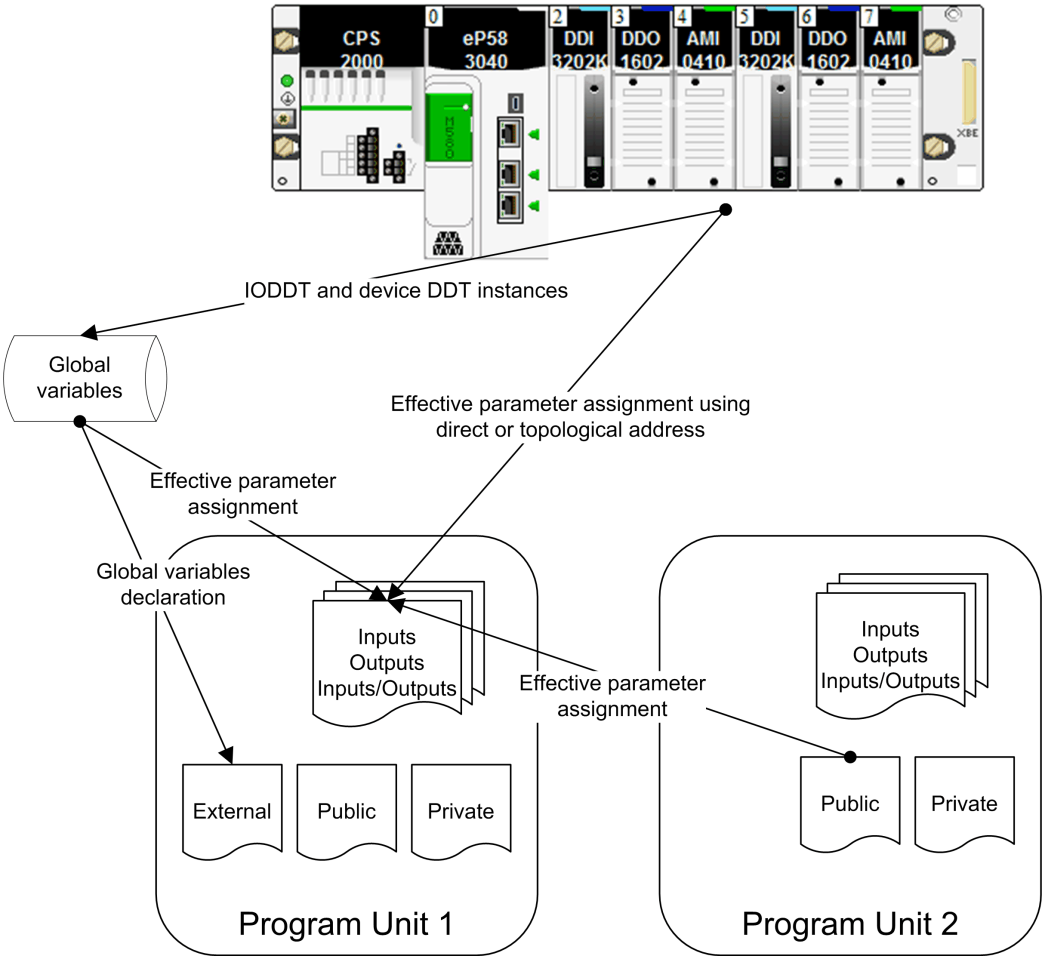
1 Global variables, managed via the Data Editor

2 Variables under `Program0` scope, managed via the Program Unit Data Editor of the Program0.

3 Variables under `Program1` scope, managed via the Program Unit Data Editor of the Program1.

Variables Accessibility

The variable accessibility is dependent to the scope (Global, Program Unit):



NOTE: System behavior of the PLC is accessible using the system bits and system words into the Program Unit.

Program Unit Data Editor Access

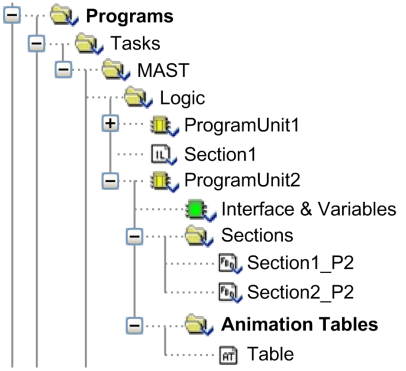
Introduction

The variables under the scope of a Program Unit can be accessed from the structural view in the Program Unit data editor. It has features which support the following:

- Creating and declaring variables of different natures,
  - Parameters (Inputs, Outputs, and Inputs/Outputs)
  - Public, which can be accessible also from an other Program Unit
  - External, for global variables used in the Program Unit
  - Private, only accessible in scope of the Program Unit
- Assigning variables to Program Unit parameters (via effective parameter attribute),
- Searching/sorting/filtering variables.
- Managing of instances of data belonging to the family of function blocks (EFB/DFB).

Accessing the Program Unit Data Editor

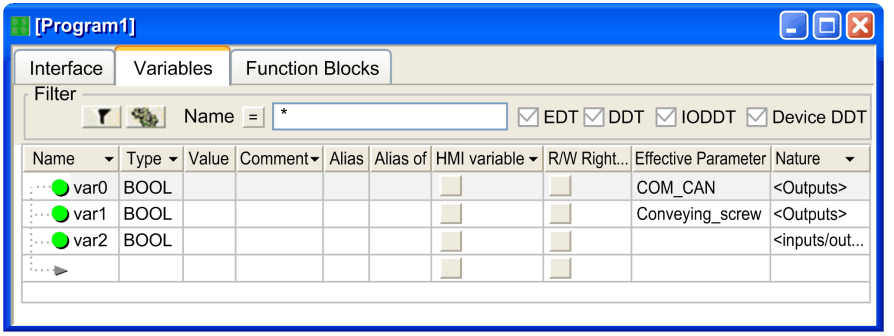
Structural view of a project:



Carry out the following steps:

Step	Action
1	Click the left mouse button on the selected **Interface & Variables** directory, or place the cursor on this directory with the arrow keys.
2	Select (right-click or **Shift+F10** combination) the **Open** command from the shortcut menu. or Double click on the **Interface & Variables**. or Select **Edit > Open**. **Result**: The Program Unit data editor opens. The **Variables** tab is shown by default.

Program Unit Data Editor:



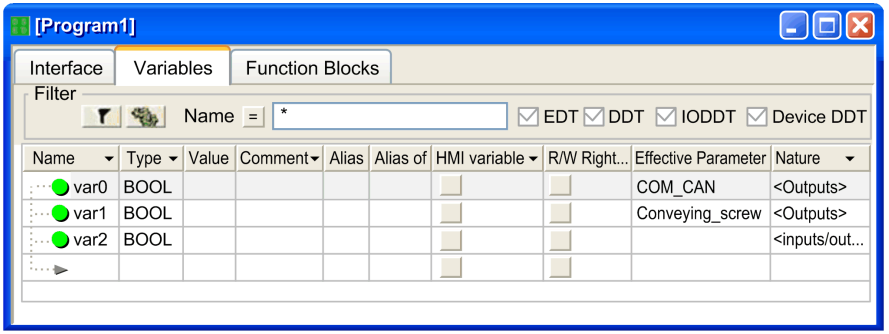
Description of Program Unit Data Editor Tabs

Introduction

NOTE: The different fields can be assigned parameters, page 425.

Tab: Variables

The **Variables** tab enables the management of all the variables that belong to the Program Unit:



Description:

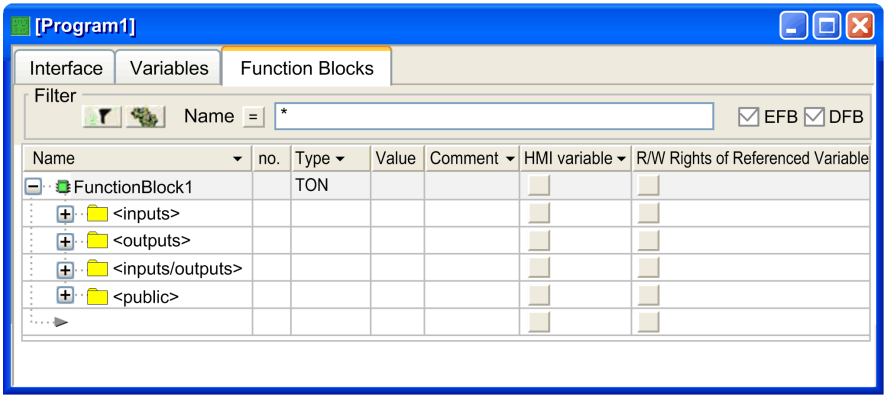
Element	Description
Filter	
Button 	Clicking on this button updates the display corresponding to the filter term defined in the **name** field.
Button 	Clicking on this button opens a dialog box, page 433 for defining the filters.
Button 	Clicking on this button inverts the filter. The button changes from **=** to **<>** and vice versa.
Name	Column where you enter the name (symbol) of the variable you want to display. You can use the place holders (***** or **? for this.**).
EDT	Displaying variables with elementary data types
DDT	Displaying variables with derived data types
IODDT	Display the variables with derived data types (DDT) that reference inputs/outputs
Device DDT	Display the variables with device derived data types
Properties	
Name	Column where you enter the name (symbol) of the variable.

Element	Description
Type	Column where you select the variable type.
Value	Column where you initialize the variable.
Comment	Column where you enter the comment for the variable.
Alias	Column where you can change the name of a variable. Only for **<private>** variables.
Alias of	Column where you enter the complete path of the variable subfield on which you set Alias attribute.
HMI variable	Column where you check if the variable is embedded in the data dictionary.
R/W Rights of Referenced variable	Column where you set the access rights of a reference type variable.
Effective Parameter	Column where you can assign a variable to the Program Unit parameter.
Nature	Column where you set the nature of the Program Unit variable.
The above properties are displayed by default, for customized properties, refer to **Configuring the Data Editor columns**, page 425.	

NOTE: For more detailed information, refer to chapter *Managing of variables belonging to the Program Unit*, page 491.

Tab: Function Blocks

The **Function Blocks** tab enables the management of the instance of elementary and derived function blocks used in the sections of the Program Unit:



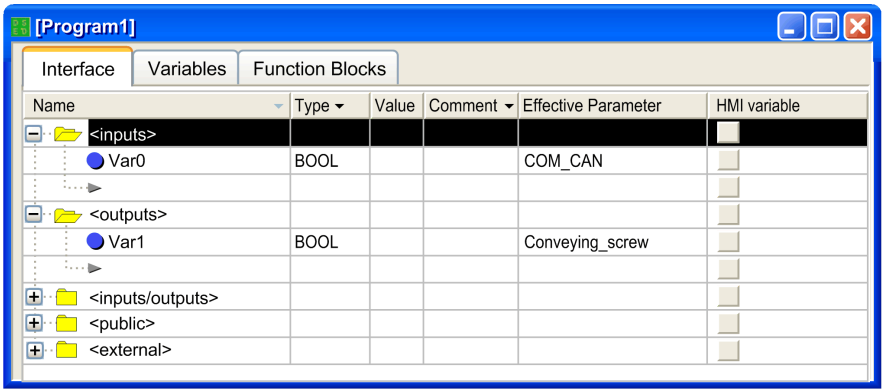
Description:

Element	Description
Filter	
Button 	Clicking on this button updates the display corresponding to the filter term defined in the **name** field.
Button 	Clicking on this button opens a dialog box, page 433 for defining the filters.
Button 	Clicking on this button inverts the filter. The button changes from **=** to **<>** and vice versa.
Name	Column where you enter the name (symbol) of the instance or the data type you want to display. You can use the place holders (***** or **? for this.**).
EFB	Displaying elementary data types
DFB	Displaying derived data types
Properties	
Name	Column where you enter the name (symbol) of the function block instance.
no.	Order number of the inputs and outputs of the function block.
Type	Column where you select the function block type.
Value	Column where you enter the initial value for the variable.
Comment	Column where you enter the comment for the variable.
HMI variable	Column where you check if the variable is embedded in the data dictionary.
R/W Rights of Referenced variable	Column where you set the access rights of a reference type variable.
The above properties are displayed by default, for customized properties, refer to **Configuring the Data Editor columns**, page 425.	

NOTE: For more detailed information, refer to chapter *Managing of instances of date belonging to the family of function blocks (EF)*, page 380.

Tab: Interface

The **Interface** tab enables the management of the Program Unit variables sorted by **Nature**:



NOTE: The **<inputs>**, **<outputs>**, **<inputs/outputs>**, **<public>** and **<external>** variables that belong to the Program Unit are displayed in this tab. These variables can be declared in this tab and they automatically added in the **Variables** tab and vice versa.

Description:

Element	Description
Properties	
Name	This column contains the Column where you set the nature of the Program Unit variable. Column where you enter the name (symbol) of the variable.
Type	Column where you select the variable type.
Value	Column where you initialize the variable.
Comment	Column where you enter the comment for the variable.
Effective Parameter	Column where you can assign a variable to the Program Unit parameter.
HMI variable	Column where you check if the variable is embedded in the data dictionary.
The above properties are displayed by default, for customized properties, refer to **Configuring the Data Editor columns**, page 425.	

Program Unit Variables

Naming Rule

You can duplicate a variable name when the variables belong to different scopes:

- Two Program Unit scopes, or
- The global scope and a Program Unit scope.

NOTE: In the same Program Unit scope, you cannot duplicate a variable name.

Variables Definition and Attributes

Program Unit Variables	Definition and access	Nature	Type allowed
Input parameters	These parameters are read-only access in the Program Unit. They are used to transfer values to the Program Unit.	**<inputs>**	EDT, DDT, ARRAY, References (REF_TO), Device DDT
Output parameters	These parameters are read/write access in the Program Unit. They are used to transfer values from the Program Unit.	**<outputs>**	EDT, DDT, ARRAY, References (REF_TO), Device DDT
Input/Output parameters	These parameters are read/write access in the Program Unit. They are used to transfer values to and from the Program Unit.	**<inputs/ outputs>**	EDT, DDT, ARRAY, Device DDT, IODDT
Public variables	These local variables are accessible from another Program Unit using the effective parameter assignment.	**<public>**	EDT, DDT, ARRAY
Private variables	These local variables are only accessible inside the scope of the Program Unit to which they belong.	**<private>**	EDT, DDT, ARRAY, References (REF_TO), DFB, EFB **NOTE:** A REF_TO can be initialized with a variable of the Program Unit which **Nature** is **<private>** or **<public>**.
External variables	The global variables are accessible in the sections of the Program Unit. Declare these variables with the same name and data type as the ones defined in the global project.	**<external>**	EDT, DDT, ARRAY, References (REF_TO)

Parameter Assignment

The following table gives the rule for assigning to parameters of a Program Unit:

Program Unit Parameters	Assignment
Input parameters	Optional except for ARRAY, DDT, Device DDT, STRING, and REF_TO
Output parameters	
Input/Output parameters	Compulsory

Assign a variable for Program Unit parameter (formal parameter) with the **Effective Parameter** attribute. The assigned variable is a global variable or a public variable of another Program Unit.

Use Program Unit parameters and assigned variables of the same data type. For example, if the data type INT is defined for the input parameter, then you cannot assign to this parameter a variable of the data type DINT or REAL.

The only exceptions are for BOOL and EBOOL data types which can be mixed. For example, a %Mi internal bit of data type EBOOL can be assigned to an input parameter defined as BOOL.

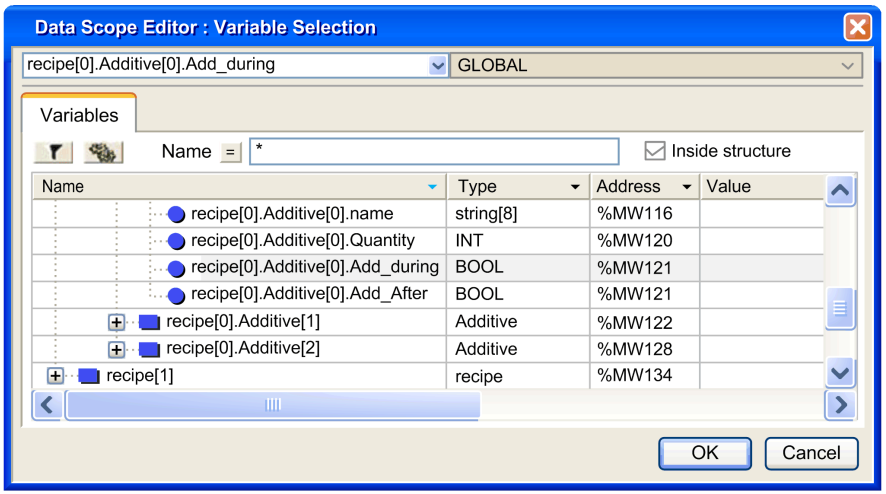
You have the following options to enter the variable in the **Effective parameter** field:

- You can enter the variable name.

  or

- Using the button **...** you can open a **Data selection** dialog box and confirm the selected variables there with **OK**.

Example of a Data selection dialog box according to the scope:



NOTE: As the management of topological variables is supported only in the data editor of the application (**Address** attribute of the variable), map an interface variable of a Program Unit to a topological variable with these steps:

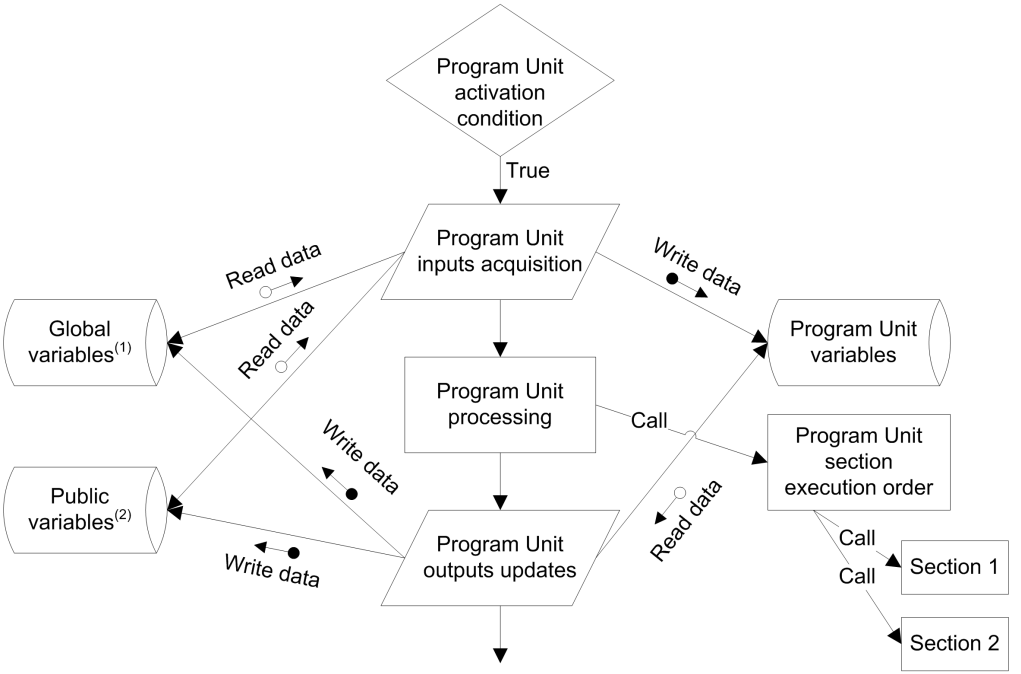
1. Define a global located variable in the Data editor.
2. Assign the global variable as an effective parameter of the Program Unit.

Program Unit Execution

The copy of the data from effective parameter to input formal parameter is performed before the Program Unit execution.

The copy of the data from output formal parameter to the effective parameter is performed after the Program Unit execution.

The following figure shows how a Program Unit is executed:

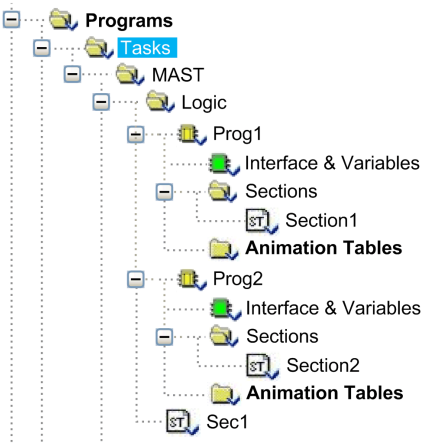


(1) Under the global scope.

(2) Under the scope of another Program Unit.

Example of Program Unit Variable Usage

The following example defines two Program Units and a section under a task:



This is a variable belonging to the global scope (declared in the **Data Editor**):

Name	Type
`Glob1`	BOOL

This is the definition of the variables belonging to the `Prog1` scope (declared in the **Program Unit Data Editor**):

Name	Type	Nature
`Glob1`	BOOL	<external>
`Input1`	BOOL	<inputs>
`Output1`	BOOL	<outputs>
`InOut1`	BOOL	<inputs/outputs>
`Pub1`	BOOL	<public>
`Priv1`	BOOL	<private>

The following table shows the syntax, the usage, and the access rights of the variables according to their nature and scope:

Variable	Usage in Prog1	Usage in the code of Prog2	Usage in Prog2 as Effective Parameter	Usage in Sect1
Glob1	Glob1 (R/W)	No	Glob1 (R/W)	Glob1 (R/W)
PROG1.Input1	Input1 (R)	No	No	PROG1.Input1 (R)
PROG1.Output1	Output1 (R/W)	No	No	PROG1.Output1 (R/W)
PROG1.InOut1	InOut1 (R/W)	No	No	PROG1.InOut1 (R/W)
PROG1.Glob1	No	No	No	PROG1.Glob1 (R/W)
PROG1.Pub1	Pub1 (R/W(1))	No	PROG1.Pub1 (R/W(2))	PROG1.Pub1 (R/W(1))
PROG1.Priv1	Priv1 (R/W)	No	No	No
%S6	%S6 (R/W(3))	%S6 (R/W(3))	No	%S6 (R/W(3))
%M23	%M23	%M23	No	%M23
%MW17	%MW17	%MW17	No	%MW17

No: The variable could not be used

R: Read only

R/W: Read and write

(1) The write access can be restricted if Pub1 has the attribute **RW program** set to read only.

(2) The write access can be restricted in external usage if Pub1 has the attribute **RW program external use** set to read only.

(3) The write access depends on the write access of the system bit or system word.

Purge of Unused Variables

To purge unused variables, the Program Unit data editor tabs provide the context menu **Purge Unused Variables**.

The purge depends on the active (selected) boxes (EDT, DDT, IODDT, and Device DDT). That is, only the visible data (of selected types) is purged:

Step	Action
1	Select the **Variables** tab of the Program Unit data editor.
2	Right-click a column of the tab.

Step	Action
3	Select the menu command **Purge Unused Variables** from the context menu.
4	Confirm with **Yes** to purge the unused variables.

NOTE: This action can also be done from the **Interface** tab and **Function Blocks** tab.

Export a Subset of Variables and FB Instances

A subset of variables or FB instances may be filtered out or selected with the Program Unit Data Editor.

Program Unit variables and FB instances are saved in source format *.XPU*.

For any of the variables or FB instances you export, you can choose two options:

- export with or without the content of all the DDTs used,
- export with or without the content of all the DFB types used.

By default, the two "export with" options are selected.

To export filtered variables:

Step	Action
1	Select the **Variables** tab in the Program Unit data editor.
2	Filter the variables as described in the chapter *Filtering of Data*, page 433.
3	Select the menu command **Export Filtered** from the context menu. **Result:** A dialog box appears on the screen.
4	Select the target folder for the export (directory tree) then enter the file name.
5	Select the export mode (with or without DDT, DFB type). **Result:** When a "with" option is selected, the corresponding box is selected.
6	Activate the **Export** command. **Result:** A progress indicator shows how the export is progressing.
7	A message is displayed in the **display window** to indicate that the export is finished.

To export selected variables:

Step	Action
1	Select the **Variables** tab in the Program Unit data editor.
2	Click on a variable to select it in the Program Unit data editor. You can select multiple variables by holding down the **Alt** or **Ctrl** keys.
3	Select the menu command **Export Selected** from the context menu. **Result:** A dialog box appears on the screen.
4	Select the target folder for the export (directory tree) then enter the file name.
5	Select the export mode (with or without DDT, DFB type). **Result:** When a "with" option is selected, the corresponding box is selected.
6	Activate the **Export** command. **Result:** A progress indicator shows how the export is progressing.
7	A message is displayed in the **display window** to indicate that the export is finished.

NOTE: The above export procedures can be also performed for the FB instances defined under the Program Unit by selecting the **Function Blocks** tab of the Program Unit data editor instead of the **Variables** tab.

Import a Subset of Variables and FB Instances

A subset of variables or FB instances may be imported with the Program Unit Data Editor.

NOTE: If the DDTs or associated DFB types have been exported (options selected on export), they are imported with the list of variables.

If the list of variables already exists in the Program Unit, the software prompts you to choose an option:

- **keep:** Retain the component of the current Program Unit.
- **replace:** Replace the Program Unit component with the one from the import file.
- **rename:** Rename the imported component to keep both components.

Import variables:

Step	Action
1	Select the **Variables** tab in the Program Unit data editor.
2	Rigth-click **Import** in the contextual menu to activate it and open a dialog box.
3	Choose the source folder for the import (directory tree) then select the file to be imported (XPU or XSY). **Result**: the name of the file then appears in the **File name** field.

Step	Action
4	Activate the **Import** command. **Result**: a progress indicator shows how the import is progressing.
5	A message tells you when the import is complete. **NOTE:** An on-screen message reports the detection of errors during the import.

NOTE: The above import procedure can be also performed for the FB instances defined under the Program Unit by selecting the **Function Blocks** tab of the Program Unit data editor instead of the **Variables** tab.

Access Right for Animation Tables in a Program Unit

The access right for object of an Animation table in a Program Unit depends on the nature of the variable:

Nature	Type	Access Right	
		With Effective Parameter	Without Effective Parameter
<inputs>	BOOL, EBOOL, and INT	R	R/W
	ARRAY, DDT, STRING, and REF_TO	R	N/A
<outputs>	BOOL, EBOOL, and INT	R	R
	ARRAY, DDT, STRING, and REF_TO	R	N/A
<inputs/outputs>	BOOL,	R	R
	EBOOL, EDT, ARRAY, DDT, STRING, and REF_TO	R	N/A
R: Read only **R/W:** Read and write **N/A:** Not applicable			

Nature	Type	Access Right
<public>	EDT, DDT, and ARRAY	R/W
	Constant	R

Nature	Type	Access Right
<private>	EDT, DDT, ARRAY, DFB, and EFB	R/W
	REF_TO, and constant	R
R: Read only **R/W:** Read and write		

NOTE: In the animation table dedicated to a Program Unit (sub folder **Animation Tables** under Program Unit directory), only variables that belong to this Program Unit can be added.

The implementation of animation tables dedicated to a Program Unit and its management is similar to the ones in the **Animation Tables** folder in the **Project** directory. For detailed information refer to chapter *Animation Tables*, page 1473.

HMI Access

The following Program Unit variables are accessible from HMI/SCADA:

Parameters:	They have read only access by default. **NOTE:** <inputs> parameter without assigned **Effective parameter** have read and write access by default.
Public:	They are read and write access by default.

Using the attribute **RW program external used**, you can restrict the default access as:

- Read only for public variables.
- No access for public variables and parameters.

In the Program Unit data editor, select the attribute **HMI variable** to embed the variable in the data dictionary. These variables are visible for HMI/SCADA.

To access the Program Unit variable from HMI/SCADA, use the variable syntax: `<Program Unit name>.<variable name>`. For example: `Program1.MyVar`

Program Unit Management

Introduction

There are two ways to duplicate a Program Unit:

- Copy of the Program Unit without copy of the effective parameters.

- Copy of the Program Unit with copy of the effective parameters.

You can use the menu command, shortcut menu or drag-and-drop to duplicate a Program Unit.

Moving a Program Unit

The execution order of programs elements can be changed exclusively in the structural view.

Process the following steps to move a Program Unit:

Step	Action
1	Click on the desired Program Unit with the left mouse button and hold the mouse button down.
2	Move the Program Unit to the required position within the current task or into another task.

NOTE: A Program Unit containing an SFC section can not be moved form the **MAST** task to another task.

Duplication Without Effective Parameters

Process the following steps to duplicate a Program Unit without copying the effective parameters:

Step	Action
1	Select the desired Program Unit.
2	Copy the Program Unit by: • Click **Copy** from the shortcut menu or • The **Ctrl+C** key combination
3	Select the location for the new Program Unit. It can be a Program Unit.
4	Paste the new Program Unit by: • Click **Paste After** from the shortcut menu or • The **Ctrl+V** key combination **Result:** The new Program Unit is created just after the selected lProgram Unit.

If using the drag-and-drop function to duplicate a Program Unit without copy of the effective parameters:

- Press the key combination **Ctrl + left-click** on the Program Unit to be duplicated (holding down the mouse button),
- drop the Program Unit at the desired location.

NOTE: A Program Unit containing an SFC section can be only duplicated in the MAST task.

Duplication With Effective Parameters

Process the following steps to duplicate a Program Unit with copying the effective parameter:

Step	Action
1	Select the desired Program Unit.
2	Copy the Program Unit by: • Click **Copy** from the shortcut menu or • The **Ctrl+C** key combination
3	Select the location for the new Program Unit. It can be a Program Unit.
4	Paste the new Program Unit by clicking **Paste Special After...** from the shortcut menu. **Result:** A message box opens asking if you want to copy the effective parameters.
5	Click **Yes** to confirm the duplication with effective parameters. **NOTE:** If you click **No**, the duplication is done without effective parameters, page 500.
6	**Result:** The new Program Unit is created just after the selected Program Unit.

If using the drag-and-drop function to duplicate a Program Unit with copy of the effective parameters:

- Press the key combination **Ctrl + Shift + left-click** on the Program Unit to be duplicated (holding down the mouse button),
- drop the Program Unit at the desired location.

NOTE: A Program Unit containing an SFC section can be only duplicated in the MAST task.

Analyze

Process the following steps to analyze a Program Unit:

Step	Action
1	Select the desired Program Unit.
2	Analyze the Program Unit by clicking **Analyze** from the shortcut menu.
3	Check the result of the analysis in the **Display Window** on the lower part of the window.

NOTE: Double-click on an error row in the display window to go to the source of the error.

Accessing Sections

Overview

This section describes accessing sections.

Creating an FBD, LD, IL, or ST Section

Introduction

Create FBD, LD, IL, or ST sections for Modicon M340:

- in the MAST and FAST tasks (**Logic** directory of the task)
- in a Program Unit that belongs to a task (**Sections** directory of the Program Unit)
- in offline and online mode

Create FBD, LD, IL, or ST sections for Modicon M580:

- in the MAST, FAST, AUX0, and AUX1 tasks (**Logic** directory of the task)
- in a Program Unit that belongs to a task (**Sections** directory of the Program Unit)
- in offline and online mode

Create FBD, LD, IL, or ST sections for Quantum, Premium and Atrium:

- in MAST, FAST, and AUX tasks (**Sections** directory of the task)
- in offline and online mode

Information for creating LD sections

An LD section contains 11-63 columns and 17-3998 rows.

The number of columns is set in the **Tools > Project Settings** dialog box, in the **Program > Languages > LD** tab, in the **Number of Columns** text box.

This setting only applies to sections to be created, not existing sections. Verify that you define the number of columns before you create a new LD section.

(The number of rows and columns can be changed during the creation of section contents.)

Information for creating FBD sections

An FBD section contains 86,400 grids (number of rows x number of columns).

The number of rows and columns is set in the **Tools > Project Settings** dialog box, in the **Program > FBD** tab.

This setting only applies to sections to be created, not existing sections. Always define the number of rows and columns before you create a new FBD section.

(The number of rows and columns can be changed during the creation of section contents.)

Creating a section

Creating a section:

Step	Action
1	To create a section in a task, select the desired directory (**Sections** or **Logic**) or place the cursor using the arrow keys on this directory.
2	Click the right mouse button or use **Shift+F10** to select **New Section** in the context menu. Or Select **Edit > New Section** **Result:** The **New** dialog box for the section opens.
3	Enter the section name. The section name must be unique in the entire project and conform to the general naming conventions. **NOTE:** Two sections of different program units can have the same name.
4	Select the programming language of the section.
5	Select the type of access protection if necessary from the **Protection** list box: • **None** No protection

Step	Action
	• **Read-Only** Write protection • **No read & write** Read and write protection See also section Project Protection, page 192.
6	Select the functional module to which the section is allocated from the **Localization** tab if necessary. See also section Functional module, page 237.
7	Define a Boolean variable in the **Condition** tab with which the section can be activated/deactivated (section active when variable=1) if required. You have the following options to select variables: • You can enter the name of the variable/address directly or copy it from the clipboard. • Use the command button **...**to open an instance selection dialog box, page 1181. Activate the **Can be forced** check box when you want to force the variable.
8	**Only applies to Quantum projects:** Assign a read RIO drop and a write RIO drop to the section in the **Remote IO** tab if required. See also section *Executing Quantum sections with distributed inputs/outputs* (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).
9	Enter a comment in the **Comment** tab if desired.
10	If you wish to create additional sections, confirm the entries with **Apply**. In this case, the **New** dialog box will remain open and you can generate additional sections. If you do not wish to create additional sections, confirm the entries with **OK**. In this case, the **New** dialog box will close and the generated section will open automatically.

Displaying or modifying section properties

Displaying or modifying section properties:

Step	Action
1	Left-click on the desired section, or place the cursor on this section with the arrow keys.
2	Select (right-click or **Shift**+**F10** key combination) the **Properties...** menu command from the shortcut menu. **Result:** The **Properties of** dialog box for the section opens.
3	Implement the desired changes.
4	Confirm all entries with **OK**.

Importing/exporting sections

See:

- Importing programs, page 1701
- Exporting programs, page 1699

Properties Dialog Box for FBD, LD, IL, or ST Sections

Structure of the properties dialog box

The properties dialog box consists of four or five tabs depending on the platform of the project:

- **General**

  Enter the name, the programming language and the type of access protection of the section in this tab.

- **Localization**

  You can enter the functional module of the section in this tab.

- **Condition**

  You can enter a Boolean variable which can enable/disable the section in this tab.

- **Remote IO**

  This tab is only available for Quantum projects.

  Assign a read RIO drop and a write RIO drop to the section in this tab.

  Assigning drops will remove them from the TASK I/O update routine and will be managed from the section.

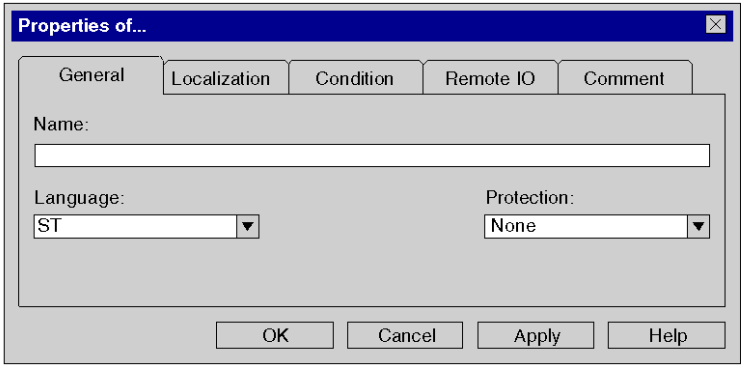
  When disabling a section by a condition, the read and write of the assigned drops are still active.

- **Comment**

  You can enter a comment in this tab.

General tab

Representation of the **General** tab:

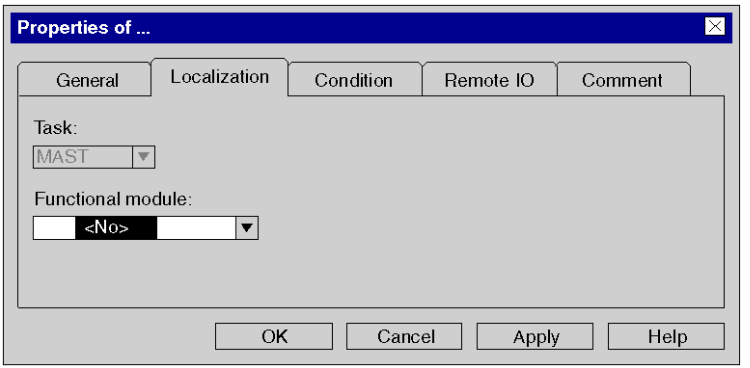


Elements of the **General** tab:

Element	Description
Name	Enter the name of the section in this text box. The section name must be unique in the entire project and conform to the general naming conventions.
Language	Select the programming language of the section from this list box. **Note:** The programming language can only be defined for new sections. The programming language cannot be changed for existing sections.
Protection	Select the type of access protection from this list box. • **None** No protection • **Read-Only** Write protection • **No read & write** Read and write protection See also *Program Unit, Section and Subroutine Protection*, page 192.
OK	Use this command button to accept the entries and close the properties dialog box.
Apply	Use this command button to accept the entries without closing the properties dialog box.

Localization tab

Representation of the **Localization** tab:

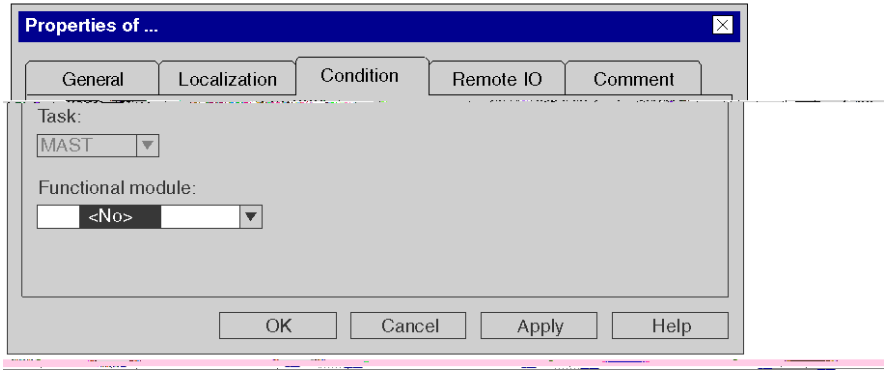


Elements of the **Localization** tab:

Element	Description
Task	The task of the section is displayed in this field.
Functional module	Select the functional module to which the section is to be allocated to in this list box. For detail, refer to chapter *Functional Modules*, page 237.
OK	Use this command button to accept the entries and close the properties dialog box.
Apply	Use this command button to accept the entries without closing the properties dialog box.

Condition tab

Display of the **Condition** tab:

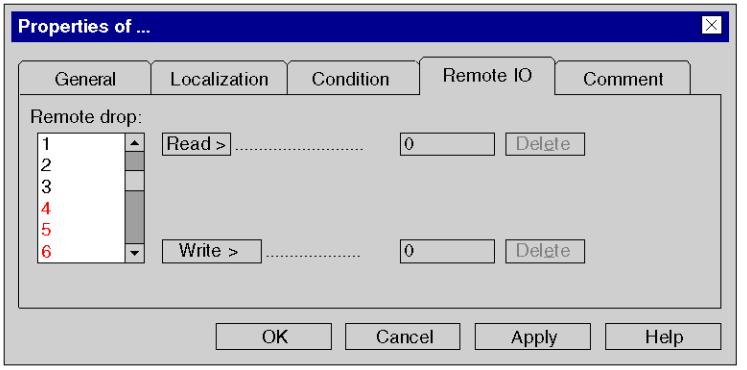


Elements of the **Condition** tab:

Element	Description
Variable	You can enter Boolean variables which can conditionally enable/disable the section in this tab. In this area, you can enter the following: • **Name** ◦ You can directly enter the name of the variable/address or paste it from the clipboard. ◦ Use the command button **...** to open an instance selection dialog box, page 1181. • **Comment** The comments about the selected variables are displayed in this text box.
Can be forced	This check box indicates whether the selected variable can be forced or not.
OK	Use this command button to accept the entries and close the properties dialog box.
Apply	Use this command button to accept the entries without closing the properties dialog box.

Remote IO tab

Representation of the **Remote IO** tab (only for Quantum projects):

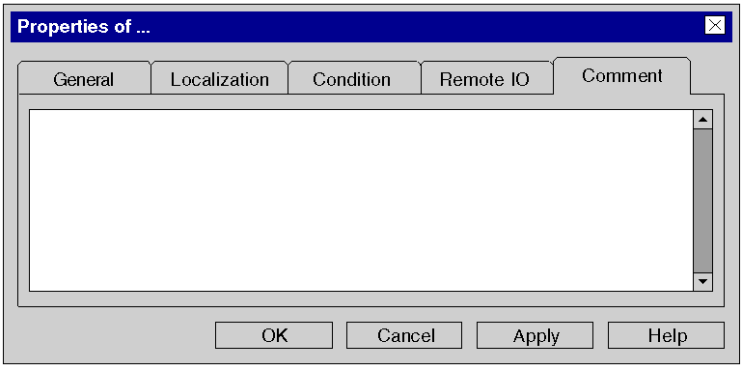


Elements of the **Remote IO** tab:

Element	Description
Remote drop	Select the remote drop to read/write from this list box. Already configured drops are displayed in black; non-configured in red. See also section Executing Quantum sections with distributed inputs/outputs (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.
Read >	Use this button to accept the selected drop as the drop to read. I.e. when processing starts on the current section, the inputs for this drop are read.
Write >	Use this button to accept the selected drop as the drop to write. I.e. when processing ends on the current section, the outputs for this drop are written.
Delete	Use this command button to reset the selected read/write drops.
OK	Use this command button to accept the entries and close the properties dialog box.
Apply	Use this command button to accept the entries without closing the properties dialog box.

Comment tab

Representation of the **Comment** tab:



Elements of the **Comment** tab:

Element	Description
Text box	A comment about the section can be entered in this text box.
OK	Use this command button to accept the entries and close the properties dialog box.
Apply	Use this command button to accept the entries without closing the properties dialog box.

Creating SFC sections

Introduction

Create SFC sections for Modicon M340, and Modicon M580:

- in the MAST task (**Logic** directory of the task)
- in a Program Unit that belongs to the MAST task (**Sections** directory of the Program Unit.)
- in offline and online mode

Create SFC sections for Quantum, Premium and Atrium:

- in the MAST task (**Sections** directory of the task)
- in offline and online mode

When creating an SFC section, the following three directories are created in addition to the actual SFC section:

- **Unused macros**

  In this directory, all macro sections are stored which were created but are no longer being used, page 900.

- **Actions**

  This directory contains all Action sections, page 963 for the current SFC section.

- **Transitions**

  This directory contains all transition sections, page 972 for the current SFC section.

Creating SFC sections

Creating SFC sections:

Step	Action
1	To create a section in the MAST task, select the desired directory (**Sections** or **Logic**) or place the cursor using the arrow keys on this directory.
2	Click the right mouse button or use **Shift**+**F10** to select **New Section** in the context menu. Or Select **Edit** > **New Section** **Result:** The **New** dialog box for the section opens.
3	Enter the section name. The section name must be unique in the entire project and conform to the general naming conventions.
4	Select the programming language SFC.
5	If necessary, assign an automation area from 0-15 to the section in the **Area number** list box. This automation area (area number) can be used as sorting criteria in the diagnostic viewer, page 1525. It makes sense to assign the area number corresponding to the functional unit. **Example:** Cutting: No. 1 Milling: No. 2 Thread cutting: No. 3 In the example, the **Area number** must have the value 1, 2 or 3.
6	If you want an explicit acknowledgment of all diagnostic results for the SFC section (e.g. exceeding the step monitoring times) to be required in the diagnostic viewer, page 1525, activate the **Operator control** check box.

Step	Action
7	Select the type of access protection if necessary from the **Protection** list box: • **None** No protection • **Read-Only** Write protection • **No read & write** Read and write protection See also Project Protection, page 192.
8	Select the functional module to which the section is allocated from the **Localization** tab if necessary. See Functional module, page 237.
9	Define a Boolean variable in the **Condition** tab with which the section can be activated/deactivated (section active when variable=1) if required. You have the following options to select variables: • You can enter the name of the variable/address directly or copy it from the clipboard. • Use the command button **...**to open an instance selection dialog box, page 1181. Activate the **Can be forced** check box when you want to force the variable.
10	**Only applies to Quantum projects:** Assign a read RIO drop and a write RIO drop to the section in the **Remote IO** tab if required. See also section *Executing Quantum sections with distributed inputs/outputs* (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).
11	Enter a comment in the **Comment** tab if desired.
12	If you wish to create additional sections, confirm the entries with **Apply**. In this case, the **New** dialog box will remain open and you can generate additional sections. If you do not wish to create additional sections, confirm the entries with **OK**. In this case, the **New** dialog box will close and the generated section will open automatically.

Displaying or modifying section properties

Displaying or modifying section properties:

Step	Action
1	Left-click on the desired section, or place the cursor on this section with the arrow keys.
2	Select (right-click or **Shift**+**F10** key combination) the **Properties...** menu command from the shortcut menu. **Result:** The **Properties of** dialog box for the section opens.

Step	Action
3	Implement the desired changes.
4	Confirm all entries with **OK**.

Importing/exporting sections

See sections:

- Importing programs, page 1701
- Exporting programs, page 1699

Properties Dialog Box for SFC Sections

Structure of the properties dialog box

The properties dialog box consists of five tabs for Quantum projects and four tabs for Modicon M580, M340, Premium, and Atrium projects:

- **General**

  You can enter the name, the programming language, the type of access protection and diagnostic settings for the section in this tab.
- **Localization**

  You can enter the functional module of the section in this tab.
- **Condition**

  You can enter a Boolean variable which can enable/disable the section in this tab.
- **Remote IO**

  This tab is only available for Quantum projects.

  Assign a read RIO drop and a write RIO drop to the section in this tab.

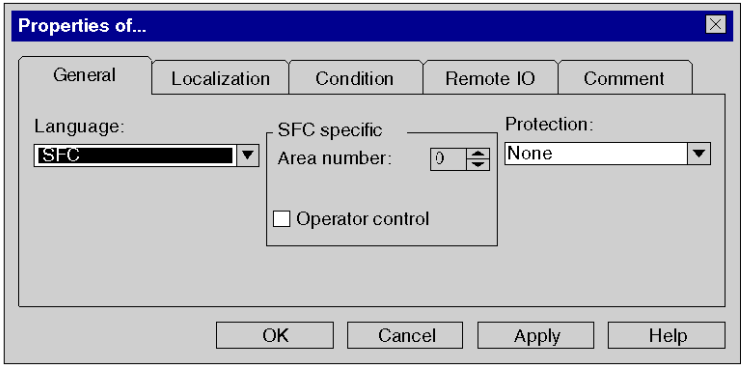
  Assigning drops will remove them from the TASK I/O update routine and will be managed from the section.

  When disabling a section by a condition, the read and write of the assigned drops are still active.
- **Comment**

  You can enter a comment in this tab.

General tab

Representation of the **General** tab:



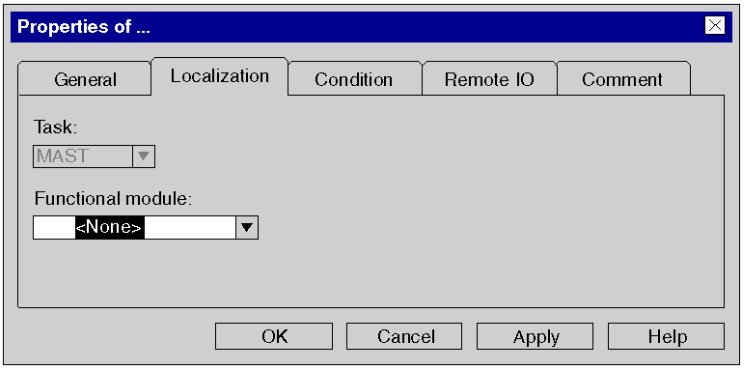
Elements of the **General** tab:

Element	Description
Name	Enter the name of the section in this text box. The section name must be unique in the entire project and conform to the general naming conventions.
Language	Select the programming language of the section from this list box. **Note:** The programming language can only be defined for new sections. The programming language cannot be changed for existing sections.
Area number	This list box is used to assign an automation range to the section; see also how to create an SFC section, page 510. This automation area (area number) can be used as sorting criteria in the diagnostic viewer, page 1525. The valid range for **Area number** 0...15.
Operator control	When you enable this check box, any diagnostics events occurring in the SFC section (e. g. violation of the step supervision times) must be acknowledged in the diagnostics viewer, page 1525.
Protection	Select the type of access protection from this list box. • **None** No protection • **Read-Only** Write protection • **No read & write** Read and write protection See also Project Protection, page 192 .

Element	Description
OK	Use this command button to accept the entries and close the properties dialog box.
Apply	Use this command button to accept the entries without closing the properties dialog box.

Localization tab

Representation of the **Localization** tab:

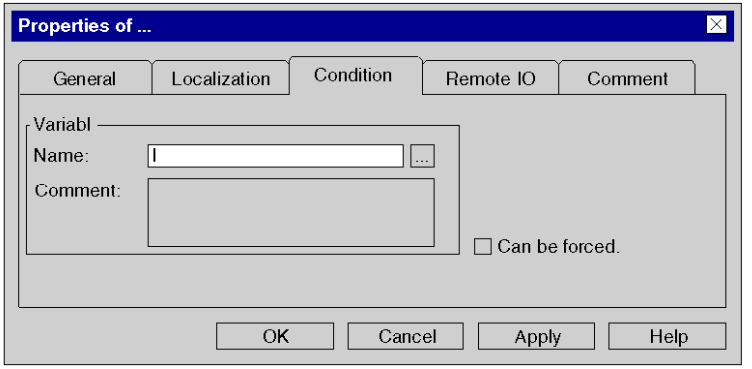


Elements of the **Localization** tab:

Element	Description
Task	The task of the section is displayed in this field.
Functional module	Select the functional module to which the section is to be allocated to in this list box. See also Functional module, page 237.
OK	Use this command button to accept the entries and close the properties dialog box.
Apply	Use this command button to accept the entries without closing the properties dialog box.

Condition tab

Display of the **Condition** tab:

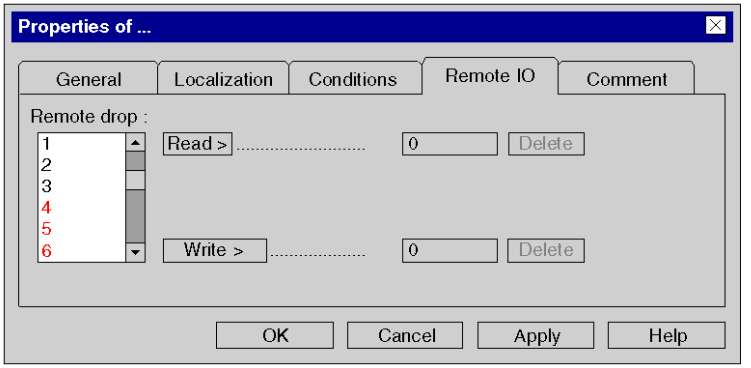


Elements of the **Condition** tab:

Element	Description
Variable	You can enter a Boolean variable which can conditionally enable/disable the section in this tab. In this area, you can enter the following: • **Name** ◦ You can directly enter the name of the variable/address or paste it from the clipboard. ◦ Use the command button **...** to open an instance selection dialog box, page 1181. • **Comment** The comments about the selected variables are displayed in this text box.
Can be forced	This check box indicates whether the selected variable can be forced or not.
OK	Use this command button to accept the entries and close the properties dialog box.
Apply	Use this command button to accept the entries without closing the properties dialog box.

Remote IO tab

Representation of the **Remote IO** tab (only for Quantum projects):

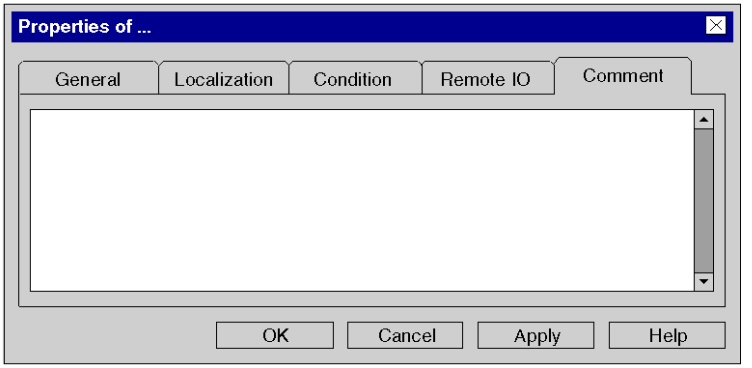


Elements of the **Remote IO** tab:

Element	Description
Remote drop	Select the remote drop to read/write from this list box. Already configured drops are displayed in black; non-configured in red. See also section Executing Quantum sections with distributed inputs/outputs (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).
Read >	Use this button to accept the selected drop as the drop to read. I.e. when processing starts on the current section, the inputs for this drop are read.
Write >	Use this button to accept the selected drop as the drop to write. I.e. when processing ends on the current section, the outputs for this drop are written.
Delete	Use this command button to reset the selected read/write drops.
OK	Use this command button to accept the entries and close the properties dialog box.
Apply	Use this command button to accept the entries without closing the properties dialog box.

Comment tab

Representation of the **Comment** tab:



Elements of the **Comment** tab:

Element	Description
Text box	A comment about the section can be entered in this text box.
OK	Use this command button to accept the entries and close the properties dialog box.
Apply	Use this command button to accept the entries without closing the properties dialog box.

Creating action, transition and macro sections

Creating action, transition and macro sections

The descriptions are found in the following sections:

- Creating action sections, page 963
- Creating transition sections, page 972
- Creating macro sections, page 937

Displaying or modifying section properties

Displaying or modifying section properties:

Step	Action
1	Left-click the target section, or place the cursor on this section with the arrow keys.
2	Select (right-click or **Shift+F10** key combination) the menu command **Properties** from the shortcut menu. **Result:** The properties dialog box, page 519 for the section is opened.
3	Modify the section name as required. Note that when you modify the name, the name of a link between the section and the element calling (action, transition or macro section) is deleted.
4	Select the type of access protection if necessary from the **Protection** list box: • **None** No protection • **Read-Only** Write protection • **No read & write** Read and write protection See also section Program Unit, Section and Subroutine Protection, page 192.
5	Enter a comment in the **Comment** tab if desired.
6	Confirm all entries with **OK**.

Action, transition and macro sections properties dialog box

Structure of the properties dialog box

The properties dialog box consists of three tabs:

- **General**

  The name and the programming language are displayed in this tab. You can also define the type of protection for the section.

- **Localization**

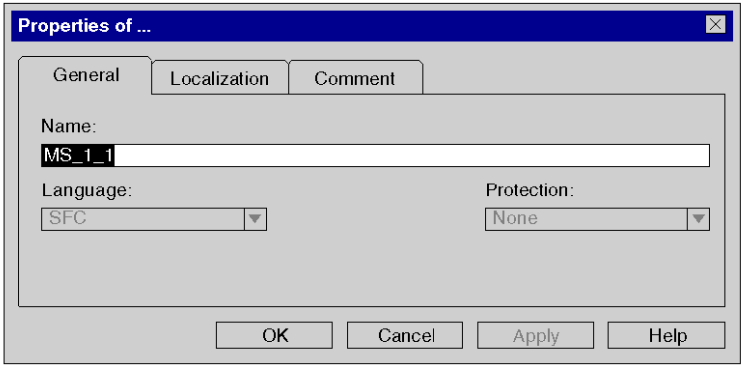
  You can enter the functional module of the section in this tab.

- **Comment**

  You can enter a comment in this tab.

General tab

Representation of the **General** tab:

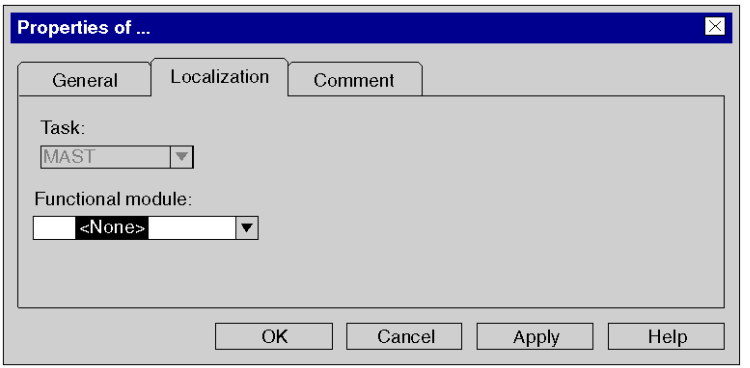


Elements of the **General** tab:

Element	Description
Name	The name of the section is displayed in this field. Note that when you modify the name, the name of a link between the section and the element calling (action, transition or macro section) is deleted.
Language	The programming language of the section is displayed in this list box. **Note:** The programming language can only be defined when creating the section. The programming language cannot be changed for existing sections.
Protection	Select the type of access protection from this list box. • **None** No protection • **Read-Only** Write protection • **No read & write** Read and write protection See also section Program Unit, Section and Subroutine Protection, page 192.
OK	Use this command button to accept all entries and close the properties dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Localization tab

Representation of the **Localization** tab:

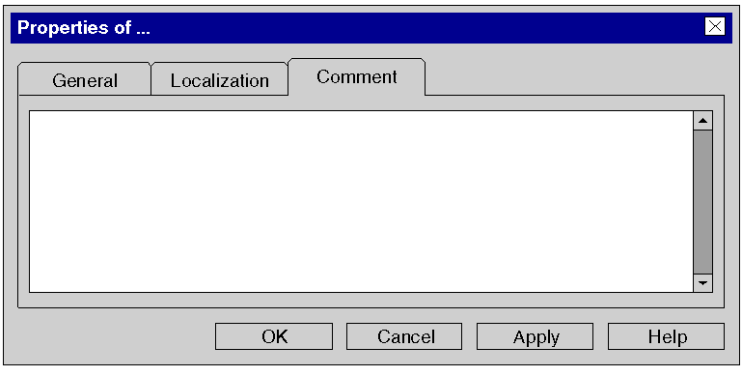


Elements of the **Localization** tab:

Element	Description
Task	The task of the section is displayed in this field.
Functional module	Select the functional module to which the section is to be allocated to in this list box. See also section Functional modules, page 237.
OK	Use this command button to accept all entries and close the properties dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box. **Note:** This command button is only available in the properties dialog box for existing sections.

Comment tab

Representation of the **Comment** tab:



Elements of the **Comment** tab:

Element	Description
Text box	A comment about the section can be entered in this text box.
OK	Use this command button to accept all entries and close the properties dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box. **Note:** This command button is only available in the properties dialog box for existing sections.

Create a Sub-Program Section (SR)

Introduction

Subroutine sections can be created:

- in offline and online mode

Subroutines are expansions to IEC 61131-3 and must be explicitly enabled via the **Tools > Project Settings** dialog box, in the **Language extensions** tab by activating the **Allow subroutines** check box.

Information for creating LD sections

An LD section contains 11-63 columns and 17-3998 lines.

The number of columns is set in the **Tools > Project Settings** dialog box, in the **Program > Languages > LD** tab, in the **Number of Columns** text box.

This setting only applies to sections to be created, not existing sections. Always define the number of columns before you create a new LD section.

(The number of lines can be increased or reduced during the creation of section contents.)

Creating subroutine sections

Creating subroutines:

Step	Action
1	Left-click the **SR Sections** directory for the target task, or place the cursor on this directory using the arrow keys.
2	Select (right-click or **Shift+F10** key combination) the **New SR Section** menu command from the shortcut menu. **Result:** The **New** dialog box for the section opens.
3	Enter the section name. The section name must be unique in the entire project and conform to the general naming conventions.
4	Select the programming language of the section.
5	Enter a comment in the **Comment** tab if desired.
6	If you wish to create additional sections, confirm the entries with **Apply**. In this case, the **New** dialog box will remain open and you can generate additional sections. If you do not wish to create additional sections, confirm the entries with **OK**. In this case, the **New** dialog box will close and the generated section will open automatically.

View or edit section properties

Displaying or modifying section properties:

Step	Action
1	Left-click on the desired section, or place the cursor on this section with the arrow keys.
2	Click the right mouse button or press **Shift+F10** to select the **Properties ...** menu command from the shortcut menu. **Result:** The **Properties of** dialog box for the section opens.

Step	Action
3	Implement the desired changes.
4	Confirm all entries with **OK**.

Importing/exporting sections

See sections:

- Import the Program, page 1701
- Exporting the program, page 1699

Property Dialog of the Sub-Program Sections

Structure of the properties dialog box

The properties dialog box consists of two tabs:

- **General**

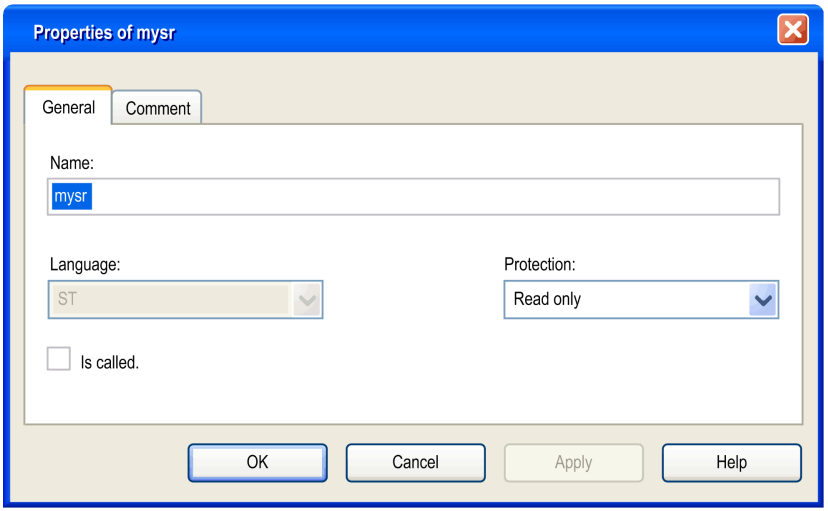
  Enter the name and the programming language of the section in this tab. The display indicates whether the subroutine is used in the project.

- **Comment**

  You can enter a comment in this tab.

General tab

Representation of the **General** tab:

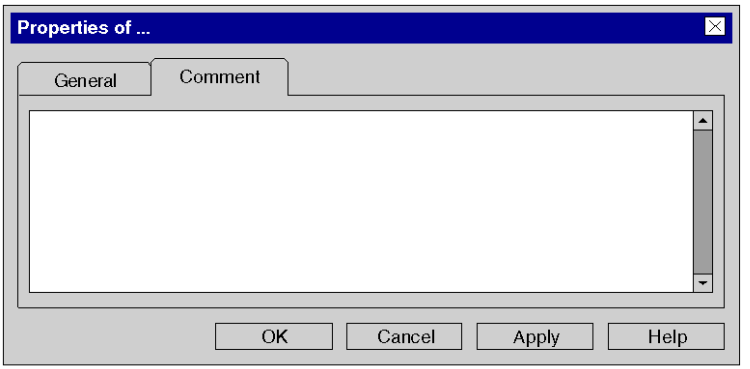


Elements of the **General** tab:

Element	Description
Name	Enter the name of the subroutine section in this text box. The section name must be unique in the entire project and conform to the general naming conventions.
Language	The programming language of the section is displayed in this list box. **Note:** The programming language can only be defined when creating the section. The programming language cannot be changed for existing sections.
Protection	Select the type of access protection from this list box. • **None** No protection • **Read-Only** Write protection • **No read & write** Read and write protection See also section Program Unit, Section and Subroutine Protection, page 192.
Is called	This check box indicates whether the subroutine is used in the project.
OK	Use this command button to accept the entries and close the properties dialog box.
Apply	Use this command button to accept the entries without closing the properties dialog box.

Comment tab

Representation of the **Comment**tab:



Elements of the **Comment** tab:

Element	Description
Text box	A comment about the section can be entered in this text box.
OK	Use this command button to accept the entries and close the properties dialog box.
Apply	Use this command button to accept the entries without closing the properties dialog box.

Creating a Timer Event Section

Introduction

Timer event sections can only be created in offline mode.

The following can be modified in offline mode:

- the number of the timer event,
- the time base,
- the unit of the preset value,
- the unit for the phase,
- the type of access protection,
- the functional module,
- the comment.

In online mode in stop mode, the following can be changed:

- the unit of the preset value,
- the unit for the phase,
- the type of access protection,
- the functional module,
- the comment.

Information for creating LD sections

An LD section contains 11-63 columns and 17-3998 rows.

The number of columns is set in the **Tools > Project Settings** dialog box, in the **Program > Languages > LD** tab, in the **Number of Columns** text box.

This setting only applies to sections to be created, not existing sections. Always define the number of columns before you create a new LD section.

(The number of rows and columns can be changed during the creation of section contents.)

Information for creating FBD sections

An FBD section contains 86,400 grids (number of rows x number of columns).

The number of rows and columns is set in the **Tools > Project Settings** dialog box, in the **Program > FBD** tab.

This setting only applies to sections to be created, not existing sections. Always define the number of rows and columns before you create a new FBD section.

(The number of rows and columns can be changed during the creation of section contents.)

Creating timer event sections

Creating timer event sections:

Step	Action
1	Left-click the **Timer Events** directory of the **Events** task, or place the cursor on this directory with the arrow keys.
2	Select (right-click or **Shift**+**F10** key combination) the **New Event Section** menu command from the shortcut menu. **Result:** The **New** dialog box for the section opens.

Step	Action
3	Enter the number of the timer event in the **Timer number** list box. The maximum number of timer events depends on the selected CPU, see also section Timer section (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.
4	Enter the time base for the preset value and phase in the **Time base** list box. **Note:** Select the largest time base to reduce system load.
5	Enter the units for the preset value in the **Preset** text box. The valid range is 0...1023. Preset value = **Preset** x **Timer base**
6	Enter the unit for the phase in the **Phase** text box. The valid range is 0...1023. Phase value = **Phase** x **Timer base**
7	Select the programming language of the section.
8	Select the type of access protection if necessary from the **Protection** list box: • **None** No protection • **Read-Only** Write protection • **No read & write** Read and write protection See also section Program Unit, Section and Subroutine Protection, page 192.
9	Select the functional module to which the section is allocated from the **Localization** tab if necessary. See also section Functional modules, page 237.
10	Enter a comment in the **Comment** tab if desired.
11	If you wish to create additional sections, confirm the entries with **Apply**. In this case, the **New** dialog box will remain open and you can generate additional sections. If you do not wish to create additional sections, confirm the entries with **OK**. In this case, the **New** dialog box will close and the generated section will open automatically.

Displaying or modifying the section properties

Displaying or modifying section properties:

Step	Action
1	Left-click on the desired section, or place the cursor on this section with the arrow keys.
2	Click the right mouse button or press **Shift+F10** to select the **Properties...** menu command from the shortcut menu. **Result:** The Properties of, page 529 dialog box for the section opens.
3	Implement the desired changes, page 526.
4	Confirm all entries with **OK**.

Importing/exporting sections

See sections:

- Import the Program, page 1701
- Exporting the program, page 1699

Properties dialog box for Timer Event Sections

Structure of the properties dialog box

The properties dialog box consists of three tabs:

- **General**

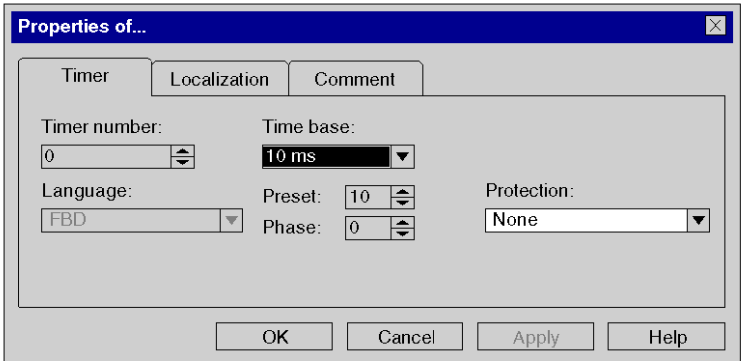
  In this tab, you can enter the timer event number, the timer time value, the programming language and the type of access protection for the timer event section, see also section Event processing (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.
- **Localization**

  You can enter the functional module of the section in this tab.
- **Comment**

  You can enter a comment in this tab.

General tab

Representation of the **General** tab:



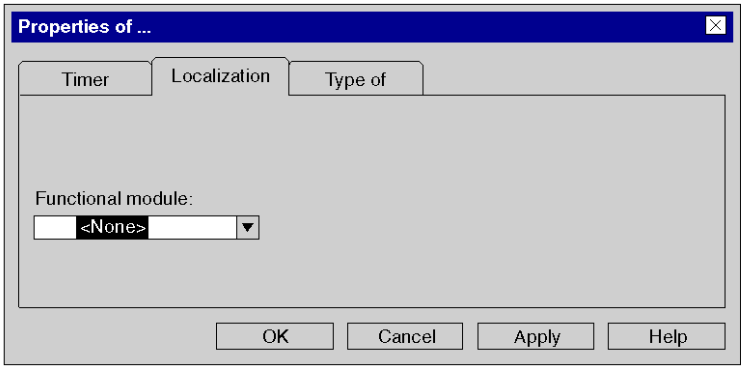
Elements of the **General** tab:

Element	Description
Timer number	Enter the number of the timer event in this list box. Note that the ITCNTRL function block (see EcoStruxure™ Control Expert, System, Block Library) is required to activate a timer event. The maximum number of timer events depends on the selected CPU, see also section Event processing (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.
Time base	Enter the time base for the preset value and phase in this list box. The following values are available: • 1 ms • 10 ms • 100 ms • 1 sec The default value is 10 ms. **Note:** Select the largest time base to reduce system load.
Preset	Enter the unit for the preset value in this text box. The valid range is 1...255. Preset value = **Preset** x **Timer base**
Phase	Enter the units for the phase in this text box. The valid range is 0...255. Phase value = **Phase** x **Timer base**
Language	The programming language of the section is displayed in this list box.

Element	Description
	Note: The programming language can only be defined when creating the section. The programming language cannot be changed for existing sections.
Protection	Select the type of access protection from this list box. • **None** No protection • **Read-Only** Write protection • **No read & write** Read and write protection See also Project Protection, page 192.
OK	Use this command button to accept the entries and close the properties dialog box.
Apply	Use this command button to accept the entries without closing the properties dialog box.

Localization tab

Representation of the **Localization** tab:

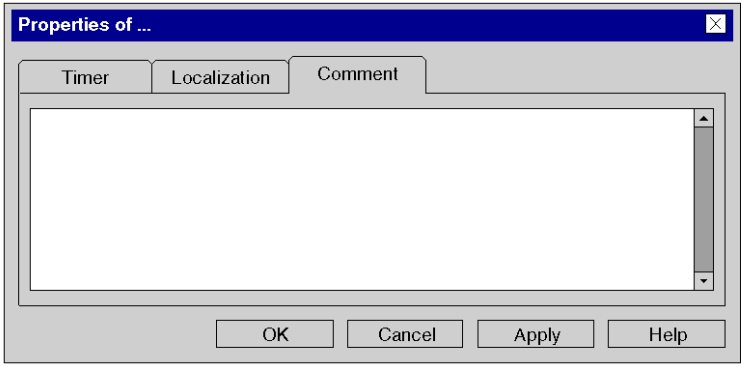


Elements of the **Localization** tab:

Element	Description
Functional module	Select the functional module to which the section is to be allocated to in this list box. See also Functional module, page 237.
OK	Use this command button to accept the entries and close the properties dialog box.
Apply	Use this command button to accept the entries without closing the properties dialog box.

Comment tab

Representation of the **Comment** tab:



Elements of the **Comment** tab:

Element	Description
Text box	A comment about the section can be entered in this text box.
OK	Use this command button to accept the entries and close the properties dialog box.
Apply	Use this command button to accept the entries without closing the properties dialog box.

Create an I/O Event Section

Introduction

I/O event sections can be created in offline and online mode.

Information for creating LD sections

An LD section contains 11-63 columns and 17-3998 rows.

The number of columns is set in the **Tools > Project Settings** dialog box, in the **Program > Languages > LD** tab, in the **Number of Columns** text box.

This setting only applies to sections to be created, not existing sections. Always define the number of columns before you create a new LD section.

(The number of rows and columns can be changed during the creation of section contents.)

Information for creating FBD sections

An FBD section contains 86,400 grids (number of rows x number of columns).

The number of rows and columns is set in the **Tools > Project Settings** dialog box, in the **Program > FBD** tab.

This setting only applies to sections to be created, not existing sections. Always define the number of rows and columns before you create a new FBD section.

(The number of rows and columns can be changed during the creation of section contents.)

Creating I/O event sections

Creating I/O event sections:

Step	Action
1	Left-click on the **I/O Events** directory of the **Events** task, or place the cursor arrow on this directory.
2	Use the right mouse button or the **Shift**+**F10** key combination to select the **New Event Section** menu command in the context menu. **Result:** The **New** dialog box for the section opens.
3	Enter the number of the I/O event in the **I/O Event number** list. The maximum number of I/O events depends on the selected CPU, see also the section on Tasks and Processes (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).
4	Select the programming language of the section.
5	Select the type of access protection if necessary from the **Protection** list: • **None** No protection • **Read-Only** Write protection • **No read & write** Read and write protection See also section Program Unit, Section and Subroutine Protection, page 192.
6	Select the functional module to which the section is allocated from the **Localization** tab if necessary. See also section Functional modules, page 237

Step	Action
7	Enter a comment in the **Comment** tab if desired.
8	If you wish to create additional sections, confirm the entries with **Apply**. In this case, the **New** dialog box remains open and you can generate additional sections. If you do not wish to create additional sections, confirm the entries with **OK**. In this case, the **New** dialog box closes and the generated section opens automatically.

Displaying or modifying section properties

Displaying or modifying section properties:

Step	Action
1	Left-click on the desired section, or place the cursor on this section with the arrow keys.
2	Use the right mouse button or press **Shift+F10** to select the **Properties** menu command. **Result:** The Properties of, page 534 dialog box for the section opens.
3	Implement the desired changes.
4	Confirm all entries with **OK**.

Importing/exporting sections

See sections:

- Import the Program, page 1701
- Exporting the program, page 1699

Property Dialog of I/O Event Sections

Structure of the properties dialog box

The properties dialog box consists of three tabs:

- **General**

  Enter the I/O event number, the programming language and the type of access protection for the I/O event section in this tab, see also section Event processing (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.

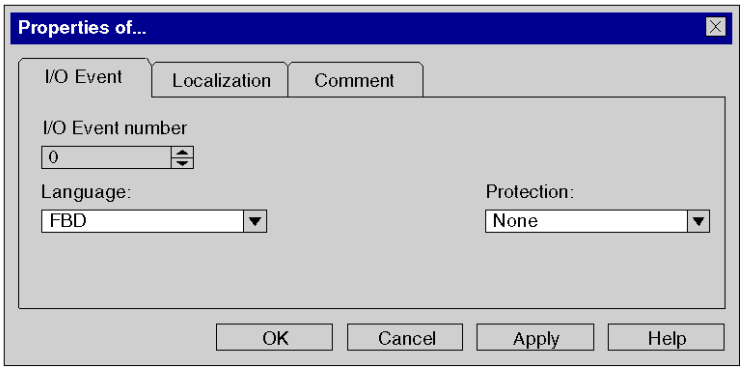
- **Localization**

  You can enter the functional module of the section in this tab.
- **Comment**

  You can enter a comment in this tab.

General tab

Representation of the **General** tab:



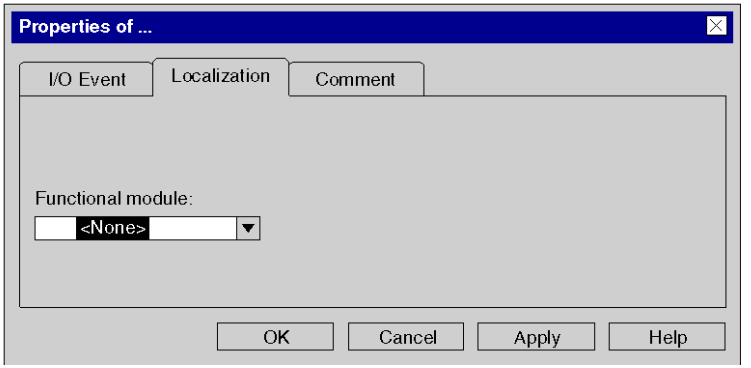
Elements of the **General** tab:

Element	Description
I/O Event number	You can enter the number of the I/O event in this list box. The maximum number of I/O events depends on the selected CPU, see also section Event section, page 532 in the Reference manual.
Language	The programming language of the section is displayed in this list box. **Note:** The programming language can only be defined when creating the section. The programming language cannot be changed for existing sections.
Protection	Select the type of access protection from this list box. • **None** No protection • **Read-Only** Write protection • **No read & write** Read and write protection See also section Program Unit, Section and Subroutine Protection, page 192.

Element	Description
OK	Use this command button to accept the entries and close the properties dialog box.
Apply	Use this command button to accept the entries without closing the properties dialog box.

Localization tab

Representation of the **Localization** tab:

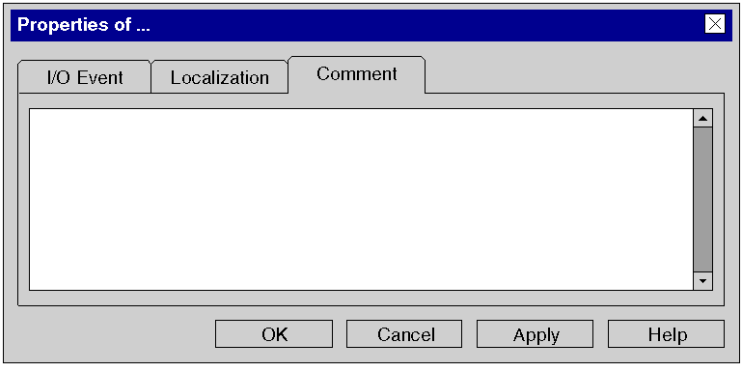


Elements of the **Localization** tab:

Element	Description
Functional module	Select the functional module to which the section is to be allocated to in this list box. See also section Functional modules, page 237.
OK	Use this command button to accept the entries and close the properties dialog box.
Apply	Use this command button to accept the entries without closing the properties dialog box.

Comment tab

Representation of the **Comment**tab:



Elements of the **Comment** tab:

Element	Description
Text box	A comment about the section can be entered in this text box.
OK	Use this command button to accept the entries and close the properties dialog box.
Apply	Use this command button to accept the entries without closing the properties dialog box.

Opening a Section

Opening a section

Opening a section:

Step	Action
1	Click the left mouse button on the section desired (section, subroutine, event), or place the cursor on this section with the arrow keys.
2	Select (right-click or **Shift+F10** key combination) the menu command **Open** from the shortcut menu. or Double click on the section. **Result:** The section is opened.

Deleting sections

Deleting sections

Deleting sections:

Step	Action
1	Left-click the target section (section, subroutine, event), or place the cursor on this section with the arrow keys.
2	Select (right-click or **Shift+F10** key combination) the **Delete** menu command from the shortcut menu. **Result:** The section is deleted.

Execution order of sections

Introduction

The execution of sections is carried out according to the order displayed in the structural view.

The arrangement of the actions, transition, macro steps, subroutines, and event sections has no influence on the execution order as their execution is program or event controlled.

The execution order of sections can be changed exclusively in the structural view.

The changes can be made:

- in offline mode
- in online mode in stop mode

Modifying the execution order

Modifying the execution order:

Step	Action
1	Click on the desired section with the left mouse button and hold the mouse button down.
2	Move the section to the required position within the current task or into another task.

Creating LL984 Segments and Networks

Introduction

The logic of LL984 is programmed in networks.

Several networks are grouped in segments.

Each network contains a matrix of 7 rows and 11 columns, where you can place contacts, coils, function blocks etc.

At the beginning of each segment's code execution the inputs of a RIO drop can be read and at the end the outputs of the same or another RIO drop can be written.

Recommendations

It is recommended not to use more than 64 segments and 5,000 networks in an application.

Restrictions

For creating LL984 segments and networks the following rules apply:

- LL984 segments can only be created when the **Ladder Logic 984 (LL984)** project setting is activated via **Tools > Project Settings > Program > Languages**.
- LL984 segments can only be created in the **MAST** task of the application, they are not allowed in the **FAST** and **AUX** tasks.
- Only one LL984 segment can be created inside the subroutine (**SR Sections**) node of the **MAST** task.
- LL984 segments can not be created under **Timer Events** or **I/O Events** nodes.
- LL984 segments can not be created in DFBs.
- For action and transition sections in SFC, LL984 is not allowed as language.

  Only ST, IL, FBD and LD are allowed as language.
- LL984 segments and sections with IEC languages can be mixed inside the **Sections** node of the **MAST** task. The same is valid inside the subroutine (**SR Sections**) node of the **MAST** task.
- LL984 networks can only be created as children of LL984 segments.
- Only LL984 networks are allowed as children of LL984 segments, no sections with IEC languages can be created there.

Creating a Segment

Step	Action
1	Left-click the **Sections** directory of the **MAST** task, or place the cursor using the arrow keys on this directory.
2	Click the right mouse button or use **Shift+F10** to select **New Section** in the context menu. **Result:** The properties dialog for the segment is opened.
3	Enter the section (segment) name. The section (segment) name must be unique in the entire project and conform to the general naming conventions of Control Expert.
4	Select **LL984 Segment** as programming **Language** of the segment.
5	Select the functional module to which the segment is attached from the **Localization** tab if necessary. Also refer to *Functional modules*, page 237.
6	Define a boolean variable in the **Condition** tab with which the segment can be activated/deactivated (segment active when variable=1) if required. You have the following options to select variables: • Enter the name of the variable/address directly or copy it from the clipboard. • Use the **...** button to open a **Data Selection** dialog.
7	**Only applies to Quantum projects:** Assign a read RIO drop and a write RIO drop to the segment in the **Remote IO** tab if required.
8	Enter a comment in the **Comment** tab if desired (max. 1024 characters).
9	If you wish to create additional segments, confirm the entries with **Apply**. In this case, the dialog will remain open and you can generate additional segments. If you do not wish to create additional segments, confirm the entries with **OK**. In this case, the dialog will close and the generated segment is displayed as the last element in the **Sections** directory. **NOTE:** The position of sections and segments inside the **Structural view** matches exactly the execution sequence and can be changed via drag-and-drop.

Creating a Network

Step	Action
1	Left-click an **LL984 Segment** in the **Sections** directory of the **MAST** task, or place the cursor using the arrow keys on this directory.
2	Click the right mouse button or use **Shift+F10** to select **New 984 Network** in the context menu.

Step	Action
	Result: The properties dialog for the network is opened.
3	Enter the network name. The network name must be unique in the entire project and conform to the general naming conventions of Control Expert.
4	**LL984 Network** is preselected for the **Language** of the network and can not be changed.
5	Define a boolean variable in the **Condition** tab with which the segment can be activated/ deactivated (segment active when variable=1) if required. You have the following options to select variables: • Enter the name of the variable/address directly or copy it from the clipboard. • Use the command button **...**to open a **Data Selection** dialog.
6	Enter a comment in the **Comment** tab if desired.
7	Confirm the entries with **OK**. The dialog will close and the generated network is displayed as the last element in the **LL984 Segment** directory. **NOTE:** The position of networks inside the **Structural view** matches exactly the execution sequence and can be changed via drag-and-drop.

A new network is normally inserted as the last element in the **LL984 Segment** directory.

To insert a new network directly before a selected network use **Insert Network …** from the context menu.

Creating a Network for Subroutines

Step	Action
1	Left-click the **SR Sections** directory of the **MAST** task, or place the cursor using the arrow keys on this directory.
2	Click the right mouse button or use **Shift**+**F10** to select **New SR Section** in the context menu. **Result:** The properties dialog for the SR section (**LL984 Segment**) is opened.
3	Enter the SR section (LL984 segment) name. The SR section (**LL984 Segment**) name must be unique in the entire project and conform to the general naming conventions of Control Expert.
4	Select **LL984 Segment** as programming **Language** of the segment. **NOTE:** There can be only 1 LL984 SR segment in a project. So when there is already 1 existing you can no longer select **LL984 Segment** as programming **Language** for a new SR section.
5	Enter a comment in the **Comment** tab if desired (max. 1024 characters).

Step	Action
6	Confirm the entries with **OK**. The dialog will close and the generated LL984 segment is displayed in the **SR Section** directory.
7	Left-click the **LL984 Segment** in the **SR Sections** directory of the **MAST** task, or place the cursor using the arrow keys on this directory.
8	Click the right mouse button or use **Shift**+**F10** to select **New 984 Network** in the context menu. **Result:** The properties dialog for the network is opened.
9	Enter the network name. The network name must be unique in the entire project and conform to the general naming conventions of Control Expert.
10	**LL984 Network** is preselected for the **Language** of the network and can not be changed.
11	Enter a comment in the **Comment** tab if desired.
12	Confirm the entries with **OK**. Tthe dialog will close and the generated network is displayed as the last element in the **LL984 Segment** directory. **NOTE:** The position of networks inside the **Structural view** matches exactly the execution sequence and can be changed via drag-and-drop.

Displaying or Modifying Segment/Network Properties

Step	Action
1	Left-click on the desired segment/network, or place the cursor on this section/network with the arrow keys.
2	Click the right mouse button or use **Shift**+**F10** to select **Properties** in the context menu. **Result:** The **Properties** dialog for is opened.
3	Implement the desired changes.
4	Confirm all entries with **OK**.

Exporting/Importing Segments/Networks

Please refer to Exporting/Importing LL984 Segments/Networks, page 1725.

Modifying LL984 Segment Properties

Introduction

This section describes the specifics of the LL984 Editor referring to modifying LL984 segment properties.

For standard features please refer to *Accessing Sections*, page 502.

Protection

The **Protection** property is not available at the moment.

Localization

You can only attach LL984 segments to a functional module and **not** LL984 networks. The networks of a segment are implicitly attached to the functional module of their segment.

You can attach an LL984 segment to a functional module either in the **Localization** tab of its properties dialog or in the **Project Browser** via drag-and-drop from the **Structural View** to the **Functional View**.

You can detach an LL984 segment from a functional module either in the **Localization** tab of its properties dialog or by selecting **Detach** from the segment´s context menu.

Condition

You can modify the **Condition** of an LL984 segment in the **Condition** tab of its properties dialog.

When you configure a boolean variable in this tab, the state of this variable is checked at the beginning of the LL984 segment's code execution.

When the variable´s state is 0, the segment and all its networks are not executed.

When the variable´s state is 1 or when no condition variable is defined for the segment, it depends on the condition variables of the networks whether the networks are executed or not.

Remote I/O

You can synchronize the Remote I/Os of an LL984 segment in the **Remote I/O** tab of its properties dialog.

Here you can configure which Remote Quantum drop is read before and which one is written after execution of the LL984 segment.

Drops which are not yet configured are displayed in red inside the list box.

Not configured drops can be selected here, but when they are not configured before next **Build**, the build will not be successful and a diagnostic message will be displayed.

NOTE: This property is only available for Quantum projects.

Comment

You can enter a comment to an LL984 segment in the **Comment** tab of its properties dialog.

The maximum size of a comment is 1024 characters.

The LL984 Subroutine Segment

You can only have 1 LL984 segment under the **SR sections** node of the **MAST** task.

The properties dialog of this LL984 subroutine segment only provides the **General** and the **Comments** tabs.

The **Is called** check book in the **General** tab is greyed and not checked.

This is done because in LL984, subroutines can be called via a memory word and so you only know at run time whether a subroutine is called or not.

You can not define a condition variable for the LL984 subroutine segment.

Modifying LL984 Network Properties

Introduction

This section describes the specifics of the LL984 Editor referring to modifying LL984 network properties.

For standard features please refer to *Accessing Sections*, page 502.

Protection

The **Protection** property is not available at the moment.

Localization

You can only attach an LL984 segment to a functional modules. The networks of a segment are implicitly attached to the functional module of their segment.

Condition

You can modify the **Condition** of an LL984 network in the **Condition** tab of its properties dialog.

When you configure a boolean variable in this tab, the state of this variable is checked at the beginning of the LL984 network's code execution.

When the variable´s state is 0, the variable and its networks are not executed.

When the variable´s state is 1 or when no condition variable is defined for the network, it depends on the condition variables of the networks whether the networks are executed or not.

You can define a condition variable for each network.

NOTE: Condition variables do not have influence on the execution of an `L9_SKP` block, that means when you have a skip over 4 networks, you skip the remainder of the current network and the next 3 networks independent of their condition variables.

Remote I/O

You can synchronize the Remote I/Os only for LL984 segments and not for LL984 networks.

Comment

You can enter a comment to an LL984 network in the **Comment** tab of its properties dialog.

The maximum size of a comment is 1024 characters.

User Directory/Hyperlink

For LL984 networks you can add hyperlinks by selecting **Add Hyperlink** from the network´s context menu.

For details please refer to chapter *Hyperlinks*, page 1774.

You can not add user directories to LL984 networks.

LL984 Networks in Subroutine Segments

The properties dialog of an LL984 network in a subroutine (SR) segment only provides the **General** and the **Comments** tabs.

The **Is called** check box in the **General** tab is greyed and not checked, because in LL984 subroutines can be called via a memory word and so you only know at run time whether a subroutine is called or not.

You can not define a condition variable for an LL984 network in a subroutine (SR) segment.

Scheduling LL984 Segments and Network

Introduction

Sections and LL984 segments (and also LL984 networks inside segments) are executed in the sequence in which they are displayed in the **Structural View** of the **Project Browser**.

To change this sequence, you can drag-and-drop sections, LL984 segments or LL984 networks inside the **Structural View**.

Restrictions

For dragging-and-dropping LL984 segments and networks the following rules apply:

- Sections (ST, LD, …) can not be dropped into LL984 segments.
- LL984 segments can not be dropped into other LL984 segments.
- LL984 networks can only be dropped into LL984 segments.

  LL984 networks can be dropped into the LL984 subroutine segment under the **SR Sections**) node of the **MAST** task.

  After dropping the networks, they are re-analyzed latest during next **Build**, because e.g. `L9_LAB` function blocks are only allowed in networks of the LL984 segment in the **SR Sections** node.

- When an LL984 network has an activation condition and is dragged-and-dropped into the LL984 subroutine segment, the LL984 network loses its activation condition because subroutine networks do not have activation conditions.

Drag-and Drop Between Applications

You can not drag-and drop LL984 segments and networks from one Control Expert application to another.

To copy sections between applications you have to use the import/export feature of Control Expert.

Deleting, Cutting, Copying and Pasting LL984 Segments and Network

Deleting LL984 Segments and Networks

You can delete LL984 segments and networks in the **Structural View** of the **Project Browser**.

To delete an LL984 segment or network either select **Delete** from its context menu or press the **DEL** key.

If you want to delete an LL984 segment, a message is displayed that all its networks will be deleted, too.

You have to confirm this message to delete the LL984 segment.

Cutting, Copying and Pasting

You can not cut, copy and paste LL984 segments and networks in a Control Expert application.

Alternatively you can use the import/export feature on segment or network level.

Control Expert Software Options

What's in This Chapter

Project Settings 549
Options 595
Customize Dialog Box 610

Subject of this Chapter

This chapter describes the options in Control Expert software. The software provides 3 types of option:

- **Project Settings**: they are specific to the generated project and to how it is executed on the PLC (for example: acknowledgment of diagnostic functions, execution of SFC language, etc.).
- **Options..**: they are specific to the workstation and may therefore differ from one station to another for the same project (for example: the way in which errors are displayed, direction of movement of selection after data entry, etc.)
- **Custom..**: they are used to customize the ergonomics of the workstation (contents of toolbar etc.).

Project Settings

Overview

This section describes the **Tools > Project settings** dialog box tab.

General Description

Introduction

Importing an inappropriate project settings file can result in unintended operation of the machine.

⚠WARNING
UNINTENDED EQUIPMENT OPERATION When importing a project settings file, ensure that the intended file is used. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

The table below describes the general functions used with all Project Settings tabs:

Fields	Description
Import	This button allows the selection of a Project Setting .XSO file. All values in the file will be taken into account for the current opened project. After Import, click **OK** or **Apply** to validate the settings.
Export	This button allows the current values of the project settings to be saved in an .XSO file.
Reset All	This button is used to reset all the settings to their default values.
OK	This button is used to save the modifications and close the current tab.
Apply	This button is used to save the modifications.
Cancel	This button is used to cancel without saving the modifications on the current tab.
Help	This button is used to call the online help.

General Navigation

The table below describes the general key shortcut navigation used with all Project Settings tabs:

Fields	Description
TAB	This key is used to move from one control to another.
ENTER	This key is used to confirm the current action.
ESC	This key is used to cancel without saving the modifications in the current tab.
Up, Down, Left and Right Arrows	These keys are used to navigate within the tree control and the right list box.

Specific Navigation

The table below describes the Specific key shortcut navigation used with all Project Settings tabs:

Fields	Description
Space	This key is used to select/deselect an element.
F2	This key is used to activate the current line.
ESC	This key is used to deactivate the current line.
Up and Down Arrows	These keys are used to navigate in the list box in the active line.
+ and -	These keys are used to increase or decrease a value in the number field in the active line.

General Project Settings

At a Glance

Access these functions when you expand (**+**) the **General** menu in the **Project Settings** structure (**Tools > Project Settings**):

- **Management of build messages**
- **Build settings**
- **Project autosaving on download**
- **PLC embedded data**
- **PLC diagnostics**
- **PLC behavior**
- **Path**
- **Time**
- **Configuration**

The following tables describe these functions individually.

Manage Build Messages

This table describes the available **Management of build messages** fields and the actions that account for the modifications:

Fields	Description	
Variable not used generates	**Nothing** or a detected **warning** is displayed in the view window if variables defined in the project are not used in the program. Variables defined in the project but are not used in the program.	
	Modification:	Online possible
	Action:	No build necessary
Multiple writing of variables generates	Variables assigned several times to different places in the program.	
	Modification:	Online possible
	Action:	No build necessary
Parameters not assigned generates	EFB or DFB parameters not assigned.	
	Modification:	Online possible
	Action:	**Build Changes** (Offline)
Multiple use of FB instance generates	EFB or DFB instances used several times at different places in the program.	
	Modification:	Online possible
	Action:	No build necessary
DDT mapping compatibility generates	An alignment test on DDTs is done for each type of data structure (BOOL, BYTE, INT, DINT...) with specific rules for each platform. Control Expert displays a notification to identify the DDTs that you modify when the **Analyze Project**, **Build Changes**, or **Rebuild All Project** command is executed. If the data structures have the same alignment in each platform, communication between them is supported. Otherwise, a notification appears to identify DDTs that you modify to allow communications. Premium and Quantum have the same mapping rule so these two platforms are aligned for communication. M340, M580, and Momentum have different mapping rules than Premium and Quantum. So, confirm that the alignment of the exchanged DDTs is the same for each platform (M340 and Premium for example) and in this case modify the DDTs. For alignment rules, refer to the *DDT: Mapping Rules* topic in the EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual.	
	Modification:	Online possible
	Action:	No build necessary

Fields	Description	
Overlapping of address generates	If **nothing** is selected, no message is displayed in the view window even if there is overlapping of address of some variables. If a detected **warning** is selected, some address messages are displayed in the view window if there is overlapping of addresses of some variables. The project is generated. Select **error** to display some detected error messages when there are overlapping addresses for some variables. When errors are detected, the project is not generated. **NOTE:** • Control Expert does not detect the overlapping of topological addresses. It detects only flat addresses. • The overlapping can generate a high number of detected errors for clients using many located variables or LL984 converted code and so increase the build generation time.	
	Modification:	Online possible
	Action:	No build necessary
Missing gateway IP @ generates	A detected error is generated on Quantum RIO architecture when a 140NOC78000, 140NOC78100, or 140CRP31200 module is configured without gateway IP address defined (equal to 0.0.0.0).	
	Modification:	Online possible
	Action:	No build necessary
Control numbers of warnings	Select **Control numbers of warnings** and a value defined in the **Maximum of warnings allowed** field (the valid range is 100...2000). Then, when the number configured is reached, the next detected alerts are not displayed in the view window.	
	Modification:	Online possible
	Action:	No build necessary
Control numbers of errors	Select **Control numbers of errors** and a value defined in the detected **Maximum of errors allowed** field (the valid range is 100–2000). Then, when the number configured is reached, the next detected errors do not appear in the view window.	
	Modification:	Online possible
	Action:	No build necessary

Build Settings

This table describes the available **Build settings** fields:

Fields	Description	
Create New data block on line	Allows you to select the management of data memory during on line modification. This setting allows you to add/modify variables without limitation of the memory block. • When selected, the dynamic allocation of unlocated variables in a new data block can be done online and you are not prompted Control Expert does not prompt you to **Rebuild All Project**. • When deselected, Control Expert acts as per the free data memory setting, which appears below. **NOTE:** • For applications (STA, XEF, ZEF files, or an upload) that contain this project setting when opened the setting is not changed. • For applications that do not contain this project setting, when opened or uploaded it is set to the default value: Selected.	
	Modification:	Online possible
	Action:	**Rebuild All Project** (Offline) to deselect **Build Changes** (Online) to select
Free data memory (in kbytes)	Allows you to select the management of data memory during on line modification. This setting was the only way to manage data memory during on line changes with previous versions of Unity Pro V6.0. Unity Pro is the former name of Control Expert for version 13.1 or earlier. It is kept for compatibility with previous versions or when only a few modifications of variables on line are planned. When this setting is valid, only one memory block 1...64 K is reserved. Each time unlocated variables are created or modified, they are added to this memory block in the controller. When the memory allocated is full, you cannot perform additional online modifications (a detected error message is displayed during the online change session). A **Rebuild All Project** is mandatory to restart with this memory block empty and available for new on line modifications.	
	Modification:	Online possible
	Action:	**Rebuild All Project** (Offline)

Fields	Description	
Virtual connected mode	Modify a project on a non-connected PC, as if this PC was connected to the controller. Then, connect the PC to the controller and activate the command **Build > Build Changes** to have these modifications written to the controller. This transfer does not stop the controller and only the changes made are taken into account. The purpose of this mode is to report whether the intended online modification is valid. Analysis is permitted in this mode, but generation is not possible. The project can be regenerated at any time, but this exits the **Virtual connected mode**.	
	Modification:	Online possible
	Action:	**Rebuild All Project** (Offline)
Generate with LD link animation	Allows choosing to include or not information on ladder diagram link animation in the generated code.	
	Modification:	Online possible
	Action:	**Rebuild All Project** (Offline)
Optimized code	Reduce the size of the code generated by removing arithmetic overflows. **NOTE:** • On Modicon M580, M340, and Premium controllers and on Quantum 140CPU6•• controllers, the %S18 bit is not managed for arithmetic operations on elementary data types (EDT) and tables. • On Quantum 140CPU3••/4••/5•• controllers, the %S18 bit is not managed for arithmetic operations on EDT. The %S20 bit is not managed for table indexing (the threshold values are still checked).	
	Modification:	Online possible
	Action:	**Build Changes** (Offline)
Safe Signature management	Specify when a new **SourceSafeSignature** is calculated for a safety-related controller application: • Automatic: (default) A new SourceSafeSignature is calculated automatically at each build. • On user request: A new SourceSafeSignature is calculated only when the **Build > Update Safe Signature** command is executed in Control Expert.	
	Modification:	Online possible
	Action:	**Build Changes** (Offline)
Apply online modification	Allows choosing whether the application is to be downloaded automatically or on a user request after the **Build Changes** command.	
	Modification:	Online possible
	Action:	No build necessary

Fields	Description	
I/O Scanning mode	Defines the I/O scanning register size for EtherNet/IP and Modbus TCP devices: • **Legacy**: The default I/O scanning Device DDT structure created is an array of BYTE, aligned on 32 bits. This mode is used in applications created with Unity Pro versions 11.1 or earlier. Set this mode for these applications to keep EtherNet/IP and Modbus TCP device settings. Unity Pro is the former name of Control Expert for version 13.1 or earlier. Switching between the **Legacy** and **Enhanced** modes can effect the Device DDT structure of the configured DIO devices. • **Enhanced**: The default I/O scanning Device DDT structure created is an array of INT, aligned on 16 bits. This mode is valid for M580 controllers as of OS version 2.00. Use this mode for applications that are created with Control Expert and are not opened or modified by Unity Pro version 11.1 or earlier. If the controller DIO ports are used for I/O scanning, the controller uses OS version 2.00 or any subsequent supporting version(s). Refer to DDT: Mapping rules (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) for alignment rules. DDT alignment verification is set in the **DDT mapping compatibility generates**, page 552 project setting. **NOTE:** This function is applicable in a Modicon M580 system.	
	Modification:	Offline only
	Action:	**Rebuild All Project** (Offline)

Auto-Save Project on Download

This table describes the available **Project autosaving on download** fields:

Fields	Description	
save STA	If selected, the archive file is automatically saved each time it is transferred to the controller (generation in online mode or transfer of complete program).	
	Modification:	Online possible
	Action:	No build necessary
save STU	If selected, the project is automatically backed up (STU and auto.STA files saved if the application has changed since the last download) each time it is transferred to the controller (generation in online mode or transfer of complete program). **NOTE:** Each time the project is saved (STU file), if the project is in **BUILT** state, a file with AUTO.STA suffix is saved, even if the STU box is deselected.	
	Modification:	Online possible
	Action:	No build necessary
save XVM	If selected, the project variables are automatically backed up (XVM file) each time the project is transferred to the controller (generation in online mode or transfer of complete program).	
	Modification:	Online possible
	Action:	No build necessary

Embedded Controller Data

This table describes the available **PLC embedded data** fields:

<table>
<tr><th>Fields</th><th colspan="2">Description</th></tr>
<tr><td>Data dictionary</td><td colspan="2">Dictionary of symbolized variables with their characteristics (address, type,...) in the controller memory.

If this setting is enabled, the symbolized variables of the application are embedded in the controller memory. Any client (SCADA using OFS, HMI, embedded Web server...) can then animate or modify those variables. By default, the settings are disabled.

The Data dictionary is generated at build time. This block is updated only if it is used during an online modification.

NOTE: Comments of the variables are not stored.

These properties improve the performances of the Data dictionary online build:
• When you assess the Preload on build changes, a version of the data dictionary that reflects the changed variable definition is downloaded by applying the change of application in the controller. The controller client tools can then be informed of new changes of data.
• The Effective Build changes time-out (sec) is the delay (1...240 s) that Control Expert is waiting between the download of the new data dictionary and the online application change. The down counting of the time-out is displayed in the progress bar during the Build Changes operation.
• Select the appropriate Only HMI variables and identify the HMI variables in the Data Property table to help improve both the online-build performance and the memory usage of the controller. This setting means that the data dictionary embeds only variables or DFB/EFB instances that have the HMI attribute set.
• Select the Keep other variables in Read Only box to embed all non-HMI read-only variables in the controller module. In this case:
◦ You can access untagged HMI variables only in read-only mode.
◦ You can modify HMI variables.
NOTE: The Keep other variables in Read Only box is available only after you select the Only HMI variables box.
• The Usage of Process Namespace setting, only available for M580 safety-related projects, determines how an operator screen can access and read process variable. For more detailed information, refer to chapter M580 Safety Project Settings (see Modicon M580, Safety System Planning Guide).
Selecting the Include DFB/PU private variables setting adds these private variables in the Data dictionary.

NOTE: A Memory full detected error message could happen during a Build Changes if there is not enough free memory. A project with the Data dictionary enabled uses a large amount of free memory during a build.</td></tr>
<tr><td></td><td>Modification:</td><td>Online possible</td></tr>
<tr><td></td><td>Action:</td><td>Build Changes (Online)</td></tr>
</table>

Fields	Description	
Upload information	If selected, this non-executable code designed to restore the source of the project on a PC is simultaneously loaded with the executable code when transferring the project to the controller. If selected, program transfer from the controller to the PC cannot be done. In this non-executable code, the **Comments** and **Animation tables** check boxes include comments associated with the variables and types and animation tables respectively.	
	Modification:	Online possible
	Action:	**Rebuild All Project** (Offline)
Upload information management	If **Automatic** is selected, the online modifications are made on both the executable code and the non-executable code designed to restore the project on the PC. If **On user request** is selected, the online modifications are only made on the executable code in the controller. Update the non-executable code implicitly with a **controller > Update Upload Information** command. An indicator at the foot of the screen is displayed if the non-executable code has not been updated.	
	Modification:	Online possible
	Action:	No build necessary
Optimize data on-line change	Select this check box to reduce the time of the **Upload information** generation relating to variables and function blocks of the application in online mode. **NOTE:** Deselect this box to retain compatibility with Unity Pro versions before 4.1. Unity Pro is the former name of Control Expert for version 13.1 or earlier.	
	Modification:	Online possible
	Action:	**Build Changes** (Offline)
Data memory protect	Select this box to help protect some data against request modification. **NOTE:** Only available for: • M580 controllers as of OS version 3.20 • M340 controllers as of OS version 3.30	
	Modification:	Online possible
	Action:	**Rebuild All Project** (Offline)

Controller Diagnostics

This table describes the available **PLC diagnostics** fields:

Fields	Description	
Application diagnostics	Select this box to choose an **Application diagnostic level**.	
	Modification:	Online possible
	Action:	**Rebuild All Project** (Offline)
Application diagnostic level	Make a selection in the pull-down menu in the **Property value** column: • **No cause of fault:** The analysis of the diagnostic detected error origin is not performed. • **Local diagnostic:** The analysis of the diagnostic detected error origin is performed within the section using the diagnostic DFB. • **Global diagnostic:** The analysis of the diagnostic detected error origin is performed within the whole application.	
	Modification:	Online possible
	Action:	**Rebuild All Project** (Offline)
System diagnostics	Select this box to choose a **System diagnostic language** from the pull-down menu in the **Property value** column.	
	Modification:	Online possible
	Action:	**Build Changes** (Offline)
System diagnostic language	Language choice: • **English** (by default) • **German** • **Italian** • **Spanish** • **French**	
	Modification:	Online possible
	Action:	**Build Changes** (Offline)
Rack Viewer diagnostics information	If selected, it provides a view of the rack.	
	Modification:	Online possible
	Action:	**Build Changes** (Online)
Rack Viewer variable names	If selected, it provides a view of the rack names.	
	Modification:	Online possible
	Action:	**Build Changes** (Online)

Fields	Description	
Program Viewer information	The **Program Viewer** runs on a PC that is connected to an Ethernet module. If selected, it provides a view of the program structure (functional and structural) and displays the contents of sections and animation. **NOTE:** • Useful only when having a FactoryCast 140NOE7711•, TSYETY5103 in the controller local rack. • Select **Data dictionary**, without the **Only HMI variables** option. The program viewer relies on the data dictionary being fully present in the controller to animate the variables in the code. If the data dictionary is not present, or incomplete, the code is visible but not animated or only partially animated. Using the HMI variables only would limit the scope of the animated variables in the code.	
	Modification:	Offline only
	Action:	**Build Changes** (Offline)
Event Logging	Select this field to enable event logging for each possible originator of event logging in the architecture. The Syslog settings entered below are transferred to each event logging originator in the architecture. **NOTE:** • Event logging is enabled if this field is selected and a valid Syslog server IP address is entered in **SYSLOG server address** field below. • Control Expert event logging is configured in the **Profiles** tab of the **Security Editor** tool. For more information, refer to the description of the **Profiles** tab and the *Syslog Events* topic in the *EcoStruxure™ Control Expert, Security Editor, Operation Guide*.	
	Modification:	Offline only
	Action:	**Build Changes** (Offline)
SYSLOG server address	Sets the IP address of the Syslog server that stores the system events logging. **NOTE:** • The default value (0.0.0.0) does not allow events to be logged. A valid server address is used when the **Event Logging** field is selected. • The Syslog server address entered here is embedded in the application.	
	Modification:	Offline only
	Action:	**Build Changes** (Offline)
SYSLOG server port number	Sets the Syslog port number. Port number value: **Default**: 601 **Range**: 0...65535	
	Modification:	Offline only
	Action:	**Build Changes** (Offline)

Fields	Description	
SYSLOG server protocol	Sets the Syslog server protocol. TCP V1 is the default Syslog protocol.	
	Modification:	The Syslog protocol cannot be modified in Unity Pro 10.0. Unity Pro is the former name of Control Expert for version 13.1 or earlier.
	Action:	—
Disable SYSLOG only for Control Expert	Modification:	Ignore events raised by EcoStruxure Control Expert in the syslog server.
	Action:	—

Controller Behavior

⚠WARNING
UNINTENDED EQUIPMENT OPERATION Confirm that the use of the function **Reset %M on Stop->Run transition** is appropriate to the application and carefully tested before putting the application into service. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

This table describes the **PLC behaviour** field:

Field	Description	
Reset %M on Stop->Run transition	This function resets %M that is not used in LL984 sections during a STOP to RUN transition of the controller. **NOTE:** This function is available for Quantum controllers OS version 2.82 and any subsequent supporting version(s).	
	Modification:	Online possible
	Action:	**Build Changes** (Offline)

The action of the **Reset %M on Stop->Run transition** function depends on these **(Tools menu) > Project Settings** check boxes:

- **General > PLC behaviour > Reset%M on Stop->Run transition**

  This setting only impacts % M not used in LL984 language sections.
- **Program > Languages > Ladder Logic 984 (LL984)**

  This setting only impacts %M used with normal coils in LL984 language sections.

The results of a transition depend on which check boxes are validated:

Reset%M	LL984	Resulting Action on Stop > Run transition
Unselected	Unselected	No %M are reset.
	Selected	Reset of %M that are used with **normal** LL984 coils.
Selected	Unselected	Reset of %M.
	Selected	Reset the %M that are not used with **latched** LL984 coils.

If there are %M conflicts:

- For a %M used on an LL984 **latched** coil **and** on a **non-LL984** coil, the LL984 **latched** coil has the higher priority

- For a %M used on an LL984 **normal** coil **and** on a **non-LL984** coil, the LL984 **normal** coil has the higher priority

If a %M has its value forced, its forced value does not change after a **Reset %M on Stop->Run transition**.

Path Usage

This table describes the **Path** field:

Fields	Description	
Hyperlink Root Path	Allows choosing a path which is used for LL984 application, page 109 when the option to convert long comments to text files is chosen. In this case, the long comments are available as hyperlinks in the variable comments column of the data editor. Additionally you can define a path for the created hyperlink text files with one of these methods: • Define a relative path in the root directory of your application. • Use the **...** button to select a different path. The subfolder of the hyperlink files is displayed below the edit field. The path defined is written to the project settings of your new application: **Tools > Project Settings > General > Path > Hyperlink Root Path**	
	Modification:	Online possible
	Action:	No build necessary

Time Stamping

This table describes the available **Time** fields:

Fields	Description	
Time Stamping Mode	Defines the time stamping mode: • Applicative (see EcoStruxure™ Control Expert, Applicative Time Stamping, User Guide) • System (see System Time Stamping, User Guide)	
	Modification:	Online possible
	Action:	**Rebuild All Project** (Offline)
Max events stored	Only available for **System** time stamping mode, this value (from 0 to 32000) defines the maximum number of time stamping events stored in the controller memory. This maximum number defines the size reserved in the controller memory for time stamping storage. See the **PLC > Memory Consumption...** to get the memory size used and refine this value (if applicable). **NOTE:** When building the project, if this value is insufficient, an error message gives the exact value to select.	
	Modification:	Online possible
	Action:	**Rebuild All Project** (Offline)
Custom behaviourTimeZone	Select this box to define your own time zone with these considerations: • You cannot select a UTC time in **Time Zone**. • You can adjust the **Time Offset**. • You can configure **Automatically adjust clock for daylight saving**.	
	Modification:	Online possible
	Action:	**Rebuild All Project** (Offline)
Time Zone	Allows you to select the wanted time zone in the list of standard time zone.	
	Modification:	Online possible
	Action:	**Rebuild All Project** (Offline)
Time Offset	Allows you to add an offset to the **Time Zone** in the range from -12 to +13 hours.	
	Modification:	Online possible
	Action:	**Rebuild All Project** (Offline)

Fields	Description	
Automatically adjust clock for daylight saving	**Enabled**: In the **START Daylight Saving** and **END Daylight Saving** fields, enter the month, day of the week, occurrence range, hour, minute, second, and offset from the respective drop-down lists to enable this feature for M580 systems. **NOTE:** You can enable this feature for M580 systems only. If you enable this feature for Modicon Premium, Quantum, Momentum, MC80, or M340 projects, there is no effect on the clock adjustment. **Disabled** (by default): The **START Daylight Saving** and **END Daylight Saving** fields are disabled because their times are automatically changed in the spring and fall every year.	
	Modification:	Online possible
	Action:	**Rebuild All Project** (Offline)

NOTE: These time settings are used only if you configure a controller with **BMXNOR••••**, **BMXCRA••••**, **BMECRA••••**, or **BMXERT••••** modules. In other cases, time settings are managed by **NTP** or by your controller application. The Quantum and Premium Ethernet modules have their own time setting configuration screens.

Data-Type Configuration

This table describes the **Configuration** field:

Fields	Description	
M580 preferred I/O data type (Local I/O)	You can choose the default type of I/O for adding a new device in a Modicon M580 main local rack (containing the controller): • **Device DDT** (by default): unlocated data • **Topological**: located data	
	Modification:	Online possible
	Action:	No build necessary
M580 preferred I/O data type for Quantum EIO	You can choose the default type of I/O for adding a new Quantum I/O module in a Modicon M580 remote I/O: • **Device DDT** (by default): unlocated data • **State Ram**: located data	
	Modification:	Online possible
	Action:	No build necessary

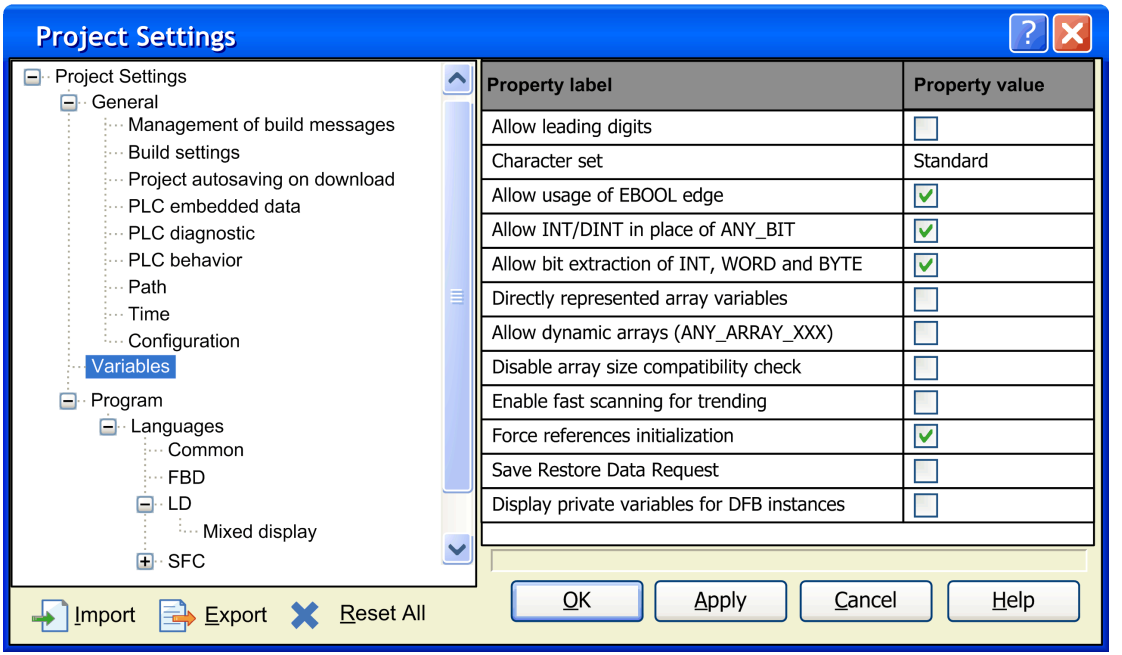
Variables

Introduction

This dialog box can be accessed with the **Tools > Project Settings > Variables**.

Representation

Representation of the **Variables** section:



Elements of the **Variables** section:

Element	Description	
Allow leading digits	When activating this check box, leading digits in names can be used (for example, section names, variable names, step names, and so on). Verify that the names contain at least one letter in addition to the leading digits.	
	Modification:	Online possible
	Action required:	**Rebuild All Project** (Offline)
Character set	**Standard**: When activating this optional button, the standard (IEC) character set in names can be used (for example, section names, variable names, step names, and so on). It is not possible to use accented characters (for example 'é'). **Extended**: When activating this optional button, the ASCII character set in names can be used (for example, section names, variable names, step names, and so on).It is possible to use accented characters. **Unicode**: When you activate this optional button, the unicode character set in names can be used (for example, section names, variable names, step names, and so on). This option is necessary to create elements in not-roman alphabets like Chinese.	
	Modification:	Online possible

Element	Description	
	Action required:	**Rebuild All Project** (Offline)
Allow usage of EBOOL edge	When you activate this check box, the `EBOOL` data type for edge recognition can be used.	
	Modification:	Online possible
	Action required:	**Build Changes** (Offline)
Allow INT/DINT in place of ANY_BIT	When activating this check box, for **generic** parameters of data type `ANY_BIT`, the parameters of data type `INT` or `DINT` can be used (not `UINT` and `UDINT`). **For example:** **Allowed:** `AND (AnyBitParam := IntVar1, AnyBitParam2 := IntVar2):` **Not allowed:** `AND_WORD (WordParam1 := IntVar1, WordParam2 := IntVar2);` (In this case, use `AND_INT`. `AND_ARRAY_WORD (ArrayInt, ...)` (In this case, perform an explicit type conversion using `INT_ARR_TO_WORD_ARR (...)`.	
	Modification:	Online possible
	Action required:	**Build Changes** (Offline)
Allow bit extraction of INT, WORD and BYTE	When activating this check box, the individual bits from an `INT`, `WORD`, or `BYTE` can be extracted, see Single bits in words (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).	
	Modification:	Online possible
	Action required:	**Build Changes** (Offline)
Directly represented array variables	When this check box is checked, it is possible to: • Declare references via indexed addresses (for example, `%MW1[3]`). See References using indexed addresses (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) • Reference bits and words as tables (Array) (for example, %MW1:16). See Table of bits and words (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) • Use a combination of both of them (for example, %MW1[3]:16).	
	Modification:	Online possible
	Action required:	**Build Changes** (Offline)
Allow dynamic arrays [ANY_ARRAY_XXX]	When activating this check box, the dynamic arrays can be used, see Tables (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).	
	Modification:	Online possible

Element	Description	
	Action required:	**Build Changes** (Offline)
Disable array size compatibility check	When activating this check box, no compatibility check is made on the arrays when one array is assigned to another, see also section Tables (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual. **Note:** This option applies only to one-dimensional array.	
	Modification:	Online possible
	Action required:	**Build Changes** (Offline)
Enable fast scanning for trending	When activating this check box, a fast scan is made on variables. The analysis of trending variables is improved.	
	Modification:	Online possible
	Action required:	**Rebuild All Project** (Offline)
Force references initialization	When activating this check box (default state), variables of Reference type (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) have to be initialized when they are created in the Data Editor, otherwise errors messages will be displayed during build. When the check box is not activated, if the variable of Reference type are not initialized in the Data Editor, they are initialized by default with the NULL value, then they have to be assigned by the `REF (see EcoStruxure™ Control Expert, Standard, Block Library)` function in the application program. **NOTE:** The application performance is better if the checkbook is activated.	
	Modification:	Online possible
	Action required:	**Rebuild All Project** (Offline)
Save Restore Data Request	When activating this check box, **Save data from PLC to file** box is displayed to propose saving CPU data, page 127 in a file when Control Expert is disconnected, page 91 from the PLC. A dialog box related to PLC init values and local init values is displayed when a build online, page 85 is performed.	
	Modification:	Online possible
	Action required:	No build necessary
Display private variables for DFB instances	When activating this check box, private variables for DFB instances are displayed to allow setting default values in data editor.	
	Modification:	Online possible
	Action required:	No build necessary

Program

Introduction

This configuration section can be accessed **Tools > Project Settings > Program**. It contains common languages, FBD, LD, SFC, and ST section display settings.

Languages

Element	Description	
Function Block Diagram (FBD)	When this check box is selected, the FBD language can be used in your project.	
	Modification:	Online possible
	Action required:	**Build Changes** (Offline)
Ladder (LD)	When this check box is selected, the LD language can be used in your project.	
	Modification:	Online possible
	Action required:	**Build Changes** (Offline)
Sequential Function Chart (SFC)	When this check box is selected, the SFC language can be used in your project.	
	Modification:	Online possible
	Action required:	**Build Changes** (Offline)
List (IL)	When this check box is selected, the IL language can be used in your project.	
	Modification:	Online possible
	Action required:	**Build Changes** (Offline)
Structured Text (ST)	When this check box is selected, the ST language can be used in your project.	
	Modification:	Online possible
	Action required:	**Build Changes** (Offline)
Ladder Logic 984 (LL984)	When this check box is selected, the LL984 language can be used in your project. The LL984 is not available for Modicon M580 system.	

Element	Description	
	NOTE: For modifying the Ladder Logic 984 (LL984) setting, the following restrictions apply: • This setting is disabled for applications using Quantum Safety PLCs, as LL984 segments and networks can only be created for non-safety applications. • You have to go offline.	
	Modification:	Online possible
	Action required:	**Build Changes** (Offline)

Common

Element	Description	
Allow procedures	When this check box is selected, the procedures can be used in your project.	
	Modification:	Online possible
	Action required:	**Build Changes** (Offline)
Allow subroutines	When this check box is selected, the subroutines can be used in your project.	
	Modification:	Online possible
	Action required:	**Build Changes** (Offline)
Allow nested comments	When this check box is selected, the comments can be nested without it being reported as an error.	
	Modification:	Online possible
	Action required:	**Build Changes** (Offline)
Allow multi assignment [a:=b:=c] (ST/LD)	When this check box is selected, the multi-assignments (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) can be used in ST.	
	Modification:	Online possible
	Action required:	**Build Changes** (Offline)
Allow empty parameters in non-formal call (ST/IL)	When this check box is selected, it is not necessary to assign a value to all formal parameters when making informal calls of functions (ST) and function blocks (IL/ST). See Informal call (ST) (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) and Informal call (IL) (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).	

Element	Description	
	Modification:	Online possible
	Action required:	**Build Changes** (Offline)
Usage of ST expressions (LD/ FBD)	When this check box is selected, the ST expressions can be used as actual parameters on FFB inputs and contacts.	
	Modification:	Online possible
	Action required:	**Build Changes** (Offline)
Maintain output links on disabled EF (EN=0)	When this check box is selected, the behavior of link values can be defined in case an EF is called conditionally (EN input connected). The default value of this option for a new application is "unchecked". After changing this setting a **Rebuild All Project** of the application is necessary, because it influences the generated code. For detailed information, refer to *Maintain Output Links on Disabled EF*, page 581. **NOTE:** This setting allows keeping the same behavior as a Concept application.	
	Modification:	Online possible
	Action required:	**Rebuild All Project** (Offline)
Display complete comments of structure element	Displays the complete comments of structured elements in the mixed mode of LD and in the tooltips for variables in FBD/LD/SFC sections, page 636.	
	Modification:	Offline only
	Action required:	No build necessary
Enable implicit type conversion	When this check box is selected, you do not need to apply explicit data type conversions for the subset of Control Expert implicit conversions in this table (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).	
	Modification:	Online possible
	Action required:	**Rebuild All Project** (Offline)

FBD

This area is used to define the number of rows and columns for Function Block Diagrams (FBD).

Element	Description
Number of rows	From 240 to 1440 (steps of 10).

Element	Description	
	Modification:	Offline only
	Action required:	No build necessary
Number of columns	From 60 to 360 (steps of 10). The section area (number of rows x number of columns) cannot exceed the maximum size of 86,400 grids: for example, 240 x 360 or 1,440 x 60. The two input fields are impacting one another. For example, the number of rows is determined by the internal calculation 86,400/number of columns (limited to the lower tenth).	
	Modification:	Offline only
	Action required:	No build necessary

LD

Element	Description	
Right-justify coils	If this check box is selected, coils, page 786 are automatically placed on the right power rail.	
	Modification:	Offline only
	Action required:	No build necessary
Number of columns	Define the number of columns in an `LD` section in this box. From 11 to 63. **Note:** These settings are only valid for new `LD` sections and not for existing sections.	
	Modification:	Offline only
	Action required:	No build necessary
Single scan edge detection for EBOOL	When an `EBOOL` is written only one time (through OFS, HMI, by program or forced by Control Expert): • If this check box is selected, writing or forcing an `EBOOL` activates the edge detection for one cycle. • If this check box is not selected, writing or forcing an `EBOOL` activates the edge detection until the next writing of this `EBOOL`.	
	Modification:	Offline only
	Action required:	**Rebuild All Project** (Offline)

LD > Mixed display

This area is used to define the available area for variable comments lines height, symbol lines height, and address lines height in mixed display mode, page 753.

Element	Description	
Comments lines height	Sets the maximum number of comment lines shown in mixed display mode. From 0 to 20 lines.	
	Modification:	Offline only
	Action required:	No build necessary
Symbol lines height	Sets the maximum number of symbol lines shown in mixed display mode. From 0 to 20 lines.	
	Modification:	Offline only
	Action required:	No build necessary
Address lines height	From 0 to 5 lines. **NOTE:** These parameters are adjustable only on the **Project Settings**.	
	Modification:	Offline only
	Action required:	No build necessary

SFC

Element	Description	
Allow macro sections	When this check box is selected, the macro sections (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) can be used in your project.	
	Modification:	Online possible
	Action required:	**Build Changes** (Offline)
Allow multiple token	When this check box is selected, the sequential control works in the Multi-token (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) operating mode.	
	Modification:	Online possible
	Action required:	**Build Changes** (Offline)

SFC > multi token

The SFC multi token option is for PL7 migration and to support the GRAFCET specification that used to be supported by Telemecanique.

Element	Description	
Max. # of active steps/section	This box is only available if the **Allow multiple token** check box activated. The maximum number of active steps per section can be defined (including all their macro sections) in this box. From 1 to 100.	
	Modification:	Online possible
	Action required:	**Build Changes** (Offline)
Max. # of active actions/section	This box is only available if the **Allow multiple token** check box activated. The maximum number of active actions per section can be defined (including all their macro sections) in this box. From 1 to 100.	
	Modification:	Online possible
	Action required:	**Build Changes** (Offline)
SetSteps: keep previous step(s) in active state	This box is only available if the **Allow multiple token** check box activated. When this check box is selected, all currently active steps remain active when you set new steps (Animation Panel or function block). When this check box is not selected, all currently active steps are deactivated when you set new steps (Animation Panel or function block).	
	Modification:	Online possible
	Action required:	**Build Changes** (Offline)
Parallel branches: allow jump in/ out	This box is only available if the **Allow multiple token** check box activated. When this check box is selected, jumps can be made into or out of a parallel sequence, see Jump into a parallel sequence (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) and Jump out of a parallel sequence (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).	
	Modification:	Online possible
	Action required:	**Build Changes** (Offline)

Element	Description	
Allow multiple evolutions by divergence of sequence selection	This box is only available if the **Allow multiple token** check box activated. When this check box is checked, all transitions of an alternative branch are processed in parallel (x-out-n-select), see x-out-n-select (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual). When this check box is not checked, all transitions of an alternative branch are processed in sequence (1-out-n-select), see Alternative sequences (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).	
	Modification:	Online possible
	Action required:	**Build Changes** (Offline)

NOTE: Confirm that you optimize the number of active steps and active actions at the right number to get the project to fit into the available memory.

SFC > Single token

This feature is only available for M340 with minimum OS version V3.30 and M580 with minimum OS version V3.20.

Element	Description	
SFC action behavior: Final Scan logic activated	When this check box is selected, the final scan is activated in SFC Action Section (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).	
	Modification:	Online possible
	Action required:	**Build Changes** (Offline)

NOTE:

- An application using the SFC final scan option cannot be opened with Control Expert V14.1 and earlier.
- The execution of two simultaneous actions in parallel can create uncertainty on the . t. In this case, the .x remains valid to indicate that the action is executing.

ST

Element	Description	
Allow jump and label	When this check box is selected, you can use jumps and labels (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in ST.	
	Modification:	Online possible
	Action required:	**Build Changes** (Offline)

LL984

Element	Description	
SKP enabled	This LL984 setting is enabled only, if the LL984 library setup was installed. When this check box is selected, the L9_SKP function block can be used in your project. **NOTE:** Modifying this setting leads to a reanalysis of the LL984 networks. This is also possible in online mode.	
	Modification:	Online possible
	Action required:	**Build Changes** (Online)

LL984 > Network Display

Element	Description	
Printing View	Using this list you select **view1** to **view4** defined under items **View 1** to **View 4** (see below). The default view is **view4**. The view you select for the **Printing View** setting is used for printing the LL984 networks.	
	Modification:	Online possible
	Action required:	No build necessary
Column Width	Using this field you can enter the column width as a floating point value (**1.0** to **3.0**. For example, a value of 3.0 means a 3 times larger column width. The default width is **1.0**. **NOTE:** Modifying the **Column Width** setting does not lead to the need of a new **Build**.	

Element	Description	
	Modification:	Online possible
	Action required:	No build necessary
View 1 to **View 4**	The LL984 editor can display various information concerning variables attached to coils, contacts, and function block nodes. According to the settings for **View 1** to **View 4** the LL984 editor shows the following information: • Address (for example, %M100) • 984-address (for example, 400100 • Variable name • Variable comment • Descriptors 1 to 9 • Topological address For further information, refer to *Items of the View 1 to View 4 Settings* (see EcoStruxure™ Control Expert, LL984 Editor, Reference Manual).	
	Modification:	Online possible
	Action required:	No build necessary

Maintain Output Links on Disabled EF (EN=0)

Introduction

The **Maintain output links on disabled EF (EN=0)** project setting allows you define the behavior of link values, connected to output pins of EFs, in case an EF is called conditionally (EN input connected).

The default value of this project setting for new applications is deactivated (unchecked).

NOTE: After changing this setting you have to make a **Rebuild All Project** of the application.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

Verify the operation of the application when using the Maintain output links on Disable EF function.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

NOTE: The behavior of the application will change after modifying this project setting that has an impact on the code execution.

Changing this Project Setting

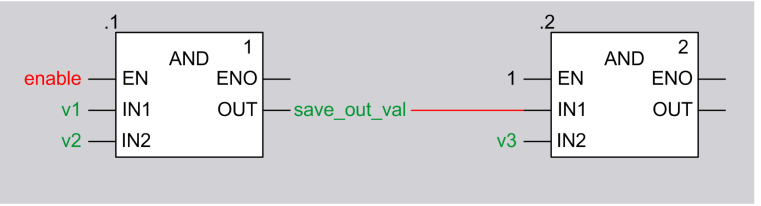
You can change the **Maintain output links on disabled EF (EN=0)** project setting in Control Expert under **Tools > Project Settings > Program > Languages > Common**.

Behavior in Case of Deactivated Project Setting

In case of a falling edge of `EN` input, the values of output variables stay unchanged, whereas output links are reset to 0.

For the example shown in the graphic below:

- `EN` -> 0
- save_out_val = 1
- link = 0 (red)

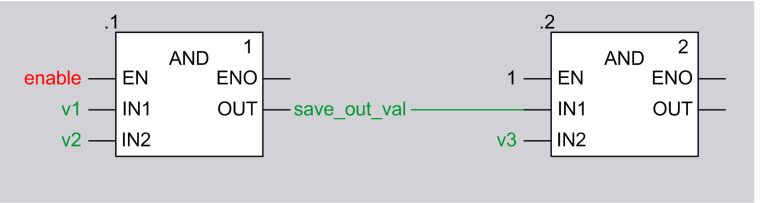


Behavior in Case of Activated Project Setting

In case of a falling edge of `EN` input, the values of output variables and output links stay unchanged.

For the example shown in the graphic below:

- `EN` -> 0
- save_out_val = 1
- link = 1 (green)



Unity Pro Compatibility (v4.0, v4.1)

NOTE:

Unity Pro is the former name of Control Expert for version 13.1 or earlier.

The following table describes the compatibility of Unity Pro version 4.0 (or earlier) and Unity Pro 4.1 (and any subsequent supporting version(s)) in regard to:

- transferring from the controller to the PC (upload)
- opening XEF file on the PC (import)
- opening STA file on the PC (archived project)

The compatibility depends on whether the project setting **Maintain output links on disabled EF (EN=0)** is activated or deactivated.

	Unity Pro Version	
Prerequisite	**4.0 and earlier**	**4.1 and subsequent supporting version (s)**
Generated with Unity Pro version 4.0 or earlier. Project setting not available in Unity Pro 4.0.	possible	possible
Generated with Unity Pro 4.1 or any subsequent supporting version(s) for an application version 4.0. Project setting deactivated.	possible (1)	possible
Generated with Unity Pro version 4.1 or any subsequent supporting version(s) for an application version 4.0. Project setting activated.	not possible	possible
Generated with Unity Pro version 4.1 or any subsequent supporting version(s) for an application version 4.1 or any subsequent supporting version(s), using M340 V2.1 controller, …) Project setting deactivated.	not possible	possible

	Unity Pro Version	
Prerequisite	**4.0 and earlier**	**4.1 and subsequent supporting version (s)**
Generated with Unity Pro version 4.1 or any subsequent supporting version(s) for an application version 4.1 or any subsequent supporting version(s) (using M340 V2.1 controller, …) Project setting activated.	not possible(2)	possible
(1) Importing a 4.1 XEF file into Unity 4.0 may require the external tool XEFDowngrader. **(2)** The controller is not recognized.		

Project Settings after Upload, Open XEF, Open STA File

NOTE:

Unity Pro is the former name of Control Expert for version 13.1 or earlier.

The following table describes the status of the **Maintain output links on disabled EF (EN= 0)** project setting after

- transferring from the controller to the PC (upload)
- opening XEF (or ZEF) file on the PC (import)
- opening STA file on the PC (archived project)

	Unity Pro Version	
Prerequisite	**4.0 and earlier**	**4.1 and subsequent supporting version (s)**
Generated with Unity Pro version 4.0 or earlier. Project setting not available in Unity Pro 4.0.	not available	not activated
Generated with Unity Pro version 4.1 or any subsequent supporting version(s) for an application version 4.0. Project setting deactivated.	not available	not activated
Generated with Unity Pro version 4.1 or any subsequent supporting version(s) for an application version 4.0 Project setting activated.	not available	activated

	Unity Pro Version	
Prerequisite	**4.0 and earlier**	**4.1 and subsequent supporting version (s)**
Generated with Unity Pro version 4.1 or any subsequent supporting version(s) for an application version 4.1 or any subsequent supporting version(s), using M340 V2.1 controller, …) Project setting deactivated.	not available	not activated
Generated with Unity Pro version 4.1 or any subsequent supporting version(s) for an application version 4.1 or any subsequent supporting version(s) (using M340 V2.1 controller, …) Project setting activated.	not available	activated

In case of uploading, importing or opening a Unity Pro application with Unity Pro 4.1 or any subsequent supporting version(s), created with Unity Pro 4.0 or earlier, the **Maintain output links on disabled EF (EN=0)** project setting is deactivated.

After a **Rebuild All Project** and download the behavior is the same as the application had in the environment of the earlier version of Unity Pro.

If you want to use the **Maintain output links on disabled EF (EN=0)** project setting, you must:

- open or upload an application,
- activate the project setting,
- **Rebuild All Project**.

Undoing EF Deletion

If the **Maintain output links on disabled EF (EN=0)** project setting is activated and an EF is called conditionally, static memory will be allocated for each output pin of the EF, to save the output values during runtime.

In case you delete this EF in the editor the static memory will be released.

If you perform an UNDO later on, the EF will be re-created and new static memory will be allocated.

After a following **Build Changes**, these mechanisms occur:

- **Maintain output links on disabled EF (EN=0)** project setting deactivated:

  No static memory is assigned for the output pins as maintaining the last link values is not needed.

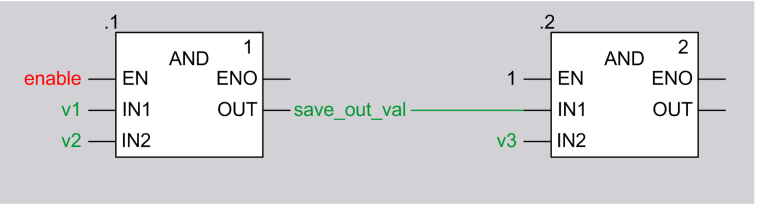
  - If EN is 0 the link is reset to 0.
  - If EN is 1 the link is written with the value of the OUT.
- **Maintain output links on disabled EF (EN=0)** project setting activated:

  New static memory is allocated for the output pins because the old memory has been released. The link´s new static memory is initialized to 0.

  - If EN is 0 the link's value remains initialized to 0.
  - If EN is 1 the EF is executed and the value of the OUT is written to the link.

Example:

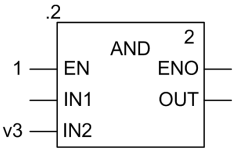
Project setting activated, EN not connected



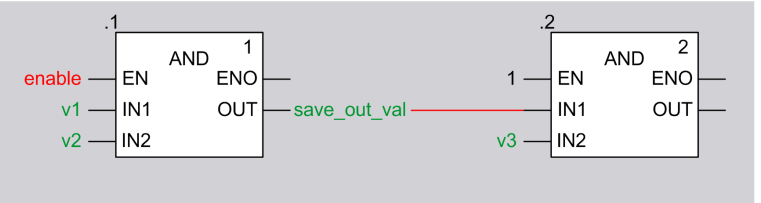
Conditions before **Build Changes**:

- EN = 0
- save_out_val = 1
- link = 1

Delete EF.



Perform UNDO and **Build Changes**.



Result:

- save_out_val = 1
- link = 0

With the next activation of the changed EF the link's new static memory will be written and works as expected on the next deactivation.

This behavior is in principle the same by using other EF types (ADD, XOR…).

EN Not Connected

If EN is not connected or not displayed (box **Show EN/ENO** not checked), no preparation is done to maintain the last link values.

If the EN input is connected later and then **Build Changes** is performed, the following mechanisms occur:

- **Maintain output links on disabled EF (EN=0)** project setting deactivated:

  No static memory is assigned as maintaining the last link values is not needed.

  After falling edge of EN, link is reset to zero, variable save_out_val is unchanged.

- **Maintain output links on disabled EF (EN=0)** project setting activated:

  New static memory is allocated in the controller to maintain the last link values for the output links. This static memory is initialized to 0.

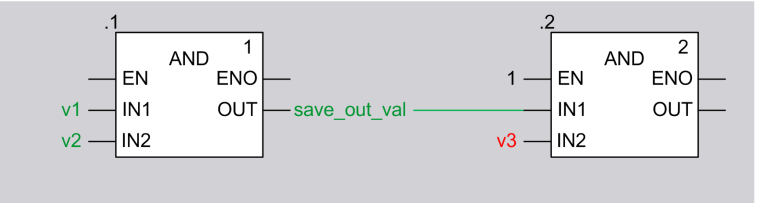
  If EN is 1 after a **Build Changes**, the EF is executed and the output data is assigned to the connected link's static memory.

  The link's static memory will be consistent at the end of the first controller working cycle after a **Build Changes**.

  NOTE: If the EN is 0 after the **Build Changes**, the static memory, representing the link, keeps its initialization state (0), which may be different to the value before the download change.

Example:

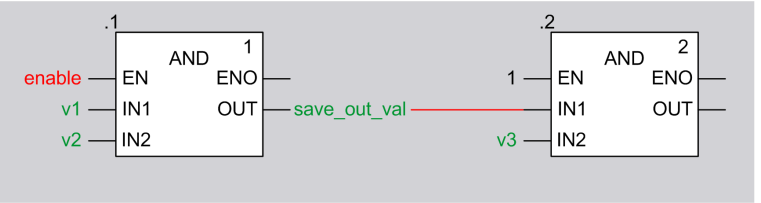
Option activated, EN not connected



Conditions before **Build Changes**:

- EN of first AND is not connected, i.e. EF AND is invoked
- save_out_val = 1
- link = 1

Connect enable to EN of first AND, set enable to 0, perform **Build Changes**.



Result:

- save_out_val = 1
- link = 0

With the next activation of the changed EF the link's new static memory will be written and works as expected on the next deactivation.

This behavior is in principle the same by using other EF types (ADD, XOR…).

Use Cases

The following use cases show that

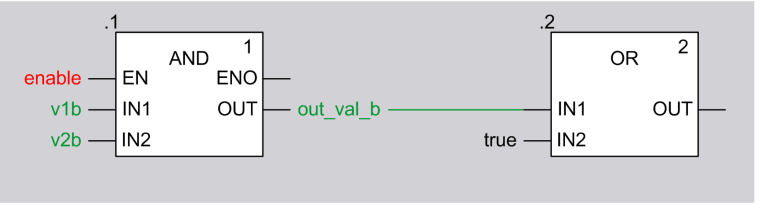
- replacing EFs by other EFs leads to initialization of output link values due to new memory allocation.
- changing output links on not changed EFs leads to maintaining the link values due to using of already allocated memory that belongs to the EFs.

The following use cases apply to all EF types.

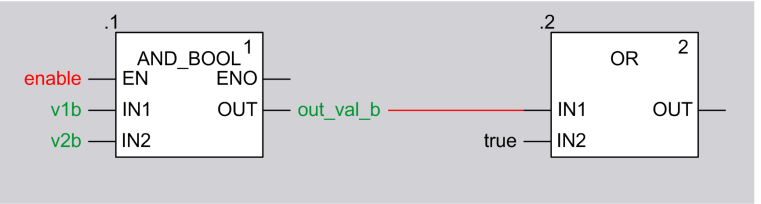
Replacing AND by AND_BOOL

Preconditions

- **Maintain output links on disabled EF (EN=0)** project setting is activated
- variable enable = 0.
- output link = 1 (due to enable was 1 in the history)



Replace AND by AND_BOOL and perform **Build Changes**.



Result: output link = 0

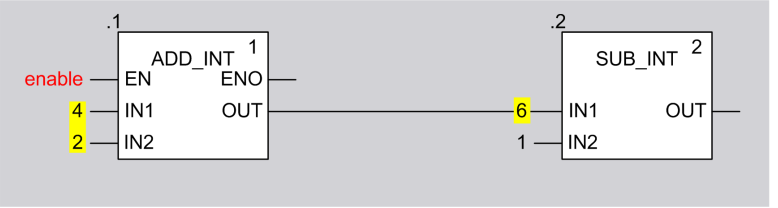
Reason for output link is 0 after Build Changes:

- The memory to hold the output values of the deleted EF (AND) has been released.
- Assigning a new EF (AND_BOOL) causes allocation of new memory to hold its output link values.
- The new memory is initialized to 0.

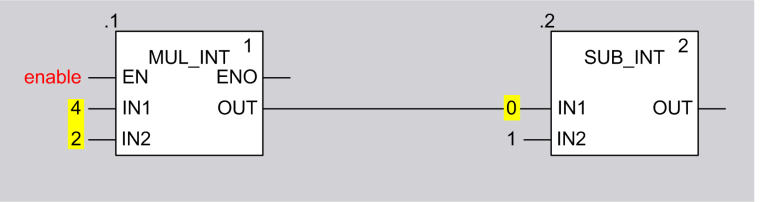
Replacing ADD by MUL

Preconditions

- **Maintain output links on disabled EF (EN=0)** project setting is activated
- variable enable = 0.
- output link = 6 (due to enable was 1 in the history)



Replace ADD_INT by MUL_INT and perform **Build Changes**.



Result: output link = 0

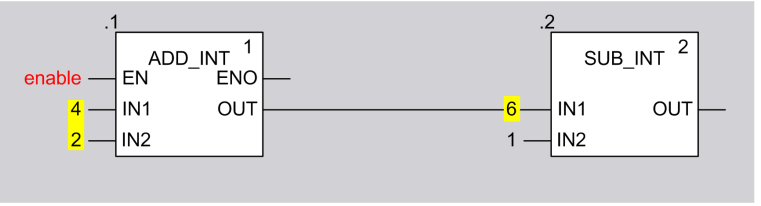
Reason for output link is 0 after Build Changes:

- The memory to hold the output values of the deleted EF (ADD_INT) has been released.
- Assigning a new EF (MUL_INT) causes allocation of new memory to hold its output link values.
- The new memory is initialized to 0.

Deleting Output Link and Performing UNDO

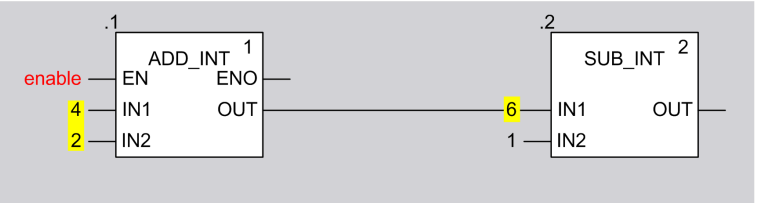
Preconditions

- **Maintain output links on disabled EF (EN=0)** project setting is activated
- variable enable = 0.
- output link = 6 (due to enable was 1 in the history)



Delete output link.

Undo deletion and perform **Build Changes**.



Result: output link = 6

Reason for output link is 6 after Build Changes:

- Deleting an output link and connecting a new output link to the same EF output pin does not cause the allocating of new memory to hold the values of the output link.
- As this memory belongs to the EF, it will not be released until deletion of the EF.
- In this use case the new link gets the memory of the deleted link (value = 6).

Configuration Settings for Operator Screens

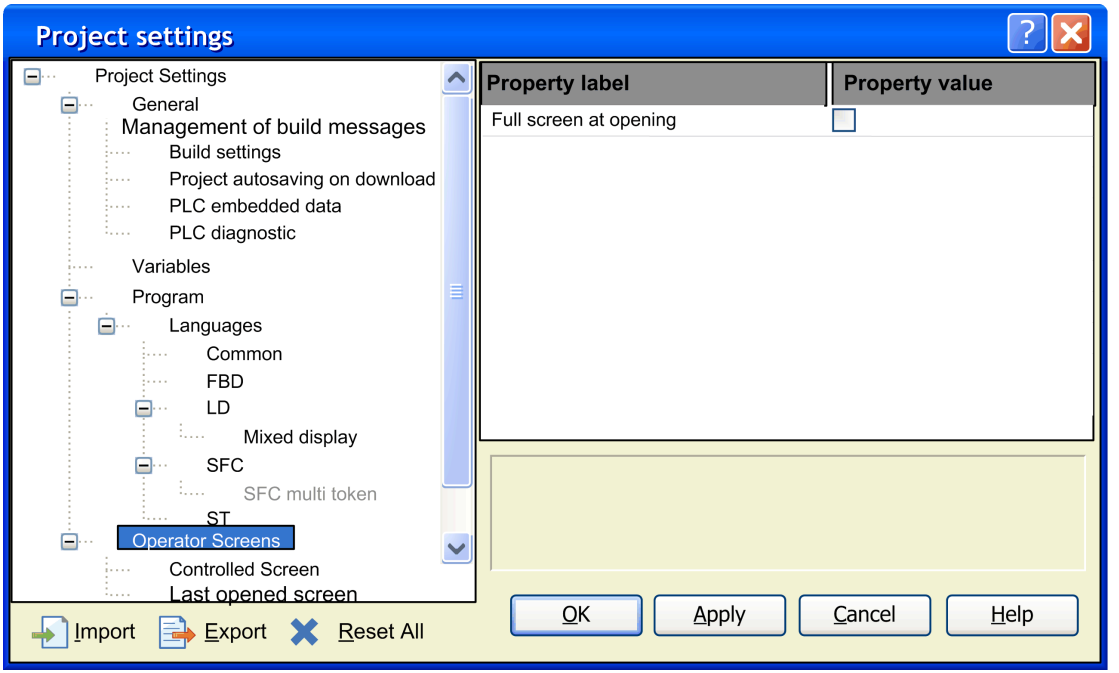
At a Glance

The configuration section accessible from **Tools > Project Settings > Operator Screens** is used to:

- Manage the control of screens by the PLC,
- Manage the detection of displayed screens,
- Manage the display or masking of screen numbers in the project browser,
- Manage the display of full screen at opening.

Representation

Representation of the **Operator Screens** section:



Description

Elements of the **Operator screens** section:

Field	Description	
Operator Screens		
Full screen at opening	When this box is checked, the operator screens open in **Full screen** mode. The menu bar is replaced by the drawing toolbar.	
	Modification:	Online possible
	Action required:	No build necessary
Controlled Screen		
Variable	**Variable**: This field is used to enter the screen control variable. In online mode, this variable enables the PLC to select the screen to be displayed. To the right	

Field	Description	
	of this field is the  button, which can be used to access the variable instance selection window. **Type**: Type of the selected variable. **Comment**: Comment on the selected variable. **NOTE:** Only variable with type EBOOL, BOOL, byte, INT, or DINT can be selected.	
	Modification:	Offline only
	Action required:	No build necessary
Displaying screens controlled via the PLC	When this box is checked, displaying of screens in online mode is controlled via the PLC. It is the value of the control variable that determines the screen displayed. This setting can also be changed via the shortcuts F2/F3 or their corresponding buttons in the toolbar  when an operator screen is open.	
	Modification:	Online possible
	Action required:	No build necessary
Display screen number in the project browser	When this box is checked, the screen number is displayed to the left of the screen name in the project browser.	
	Modification:	Online possible
	Action required:	No build necessary
Last opened screen		
Variable	**Variable**: This field is used to enter the DINT type variable containing the number of the last screen opened by the user or the PLC. To the right of this field is the  button, which can be used to access the variable instance selection window. **Type**: Type of the selected variable. **Comment**: Comment on the selected variable. **NOTE:** Only variable with type DINT can be selected.	
	Modification:	Offline only
	Action required:	No build necessary

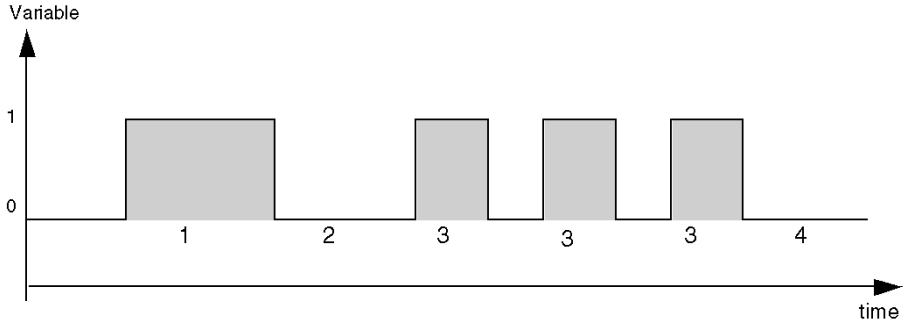
Control Variable

The control variable enables the PLC to select the screen to be displayed. This variable can be binary or numeric.

Binary Control Variable

A binary control variable allows controlling the operator screen display with only one button connected to the PLC.

The diagram below gives an example of the trend of the binary control variable:



For each of the steps of the trend diagram, the control variable operates as follows:

Step	Operation
1	Its value is equal to 1 for more than 1 second: the screens are highlighted one after another (1 second per screen) according to their position in the project browser. The PLC selects the screens by scrolling through the list from top to bottom.
2 and 4	Its value is equal to 0 for more than 1 second: the screen currently highlighted in the **Project Browser** (n = current screen) is displayed.
3	Its value is equal to 1 for less than 1 second: on each pulse, the PLC selects the screen (n+1) in the list of operator screens. The pulses are used to display the screens one by one.

NOTE: As long as this binary control variable has value 1, the operator screens are selected automatically in project browser one after the other. In this case, the context menu cannot be used in the project browser.

Numeric Control Variable

When the control variable is numeric, its value corresponds to an operator screen number. Therefore, the screen editor displays (if it exists) the screen whose number is equal to the variable value.

Options

Overview

This section describes the **Tools > Options** dialog box tab.

General Description

Introduction

Functionalities, key shortcuts and specific key-shortcut navigation are used for all Options tabs.

General Description

The table below describes the general functions used with all options tabs:

Fields	Description
Import	This button allows the selection of a Options .XOO file. All values in the file is taken into account immediately. After Import, click on OK or Apply to validate the options. **NOTE:** If a project is opened, the **Custom Library** and **Working Drive** are updated in the next Control Expert launching.
Export	This button allows to save the current values of the options in a .XOO file.
Reset All	This button is used to reset all the options to their default values.
OK	This button is used to save the modifications and close the current tab.
Apply	This button is used to save the modifications.
Cancel	This button is used to cancel without saving the modifications on the current tab.
Help	This button is used to call the online help.

General Navigation

The table below describes the general key shortcut navigation used with all Options tabs:

Fields	Description
TAB	This key is used to move from control to another.
ENTER	This key is used to confirm the current action.
ESC	This key is used to cancel without saving the modifications in the current tab.
Up, Down, Left and Right Arrows	These keys are used to navigate within the tree control and the right list box.

Specific Navigation

The table below describes the specific key shortcut navigation used with all Options tabs:

Fields	Description
Space	This key is used to check / uncheck an element.
F2	This key is used to activate the current line.
ESC	This key is used to deactivate the current line.
Up and Down Arrows	These keys are used to navigate in the list box in the active line.
+ and -	These keys are used to increase or decrease a value in number field in the active line.

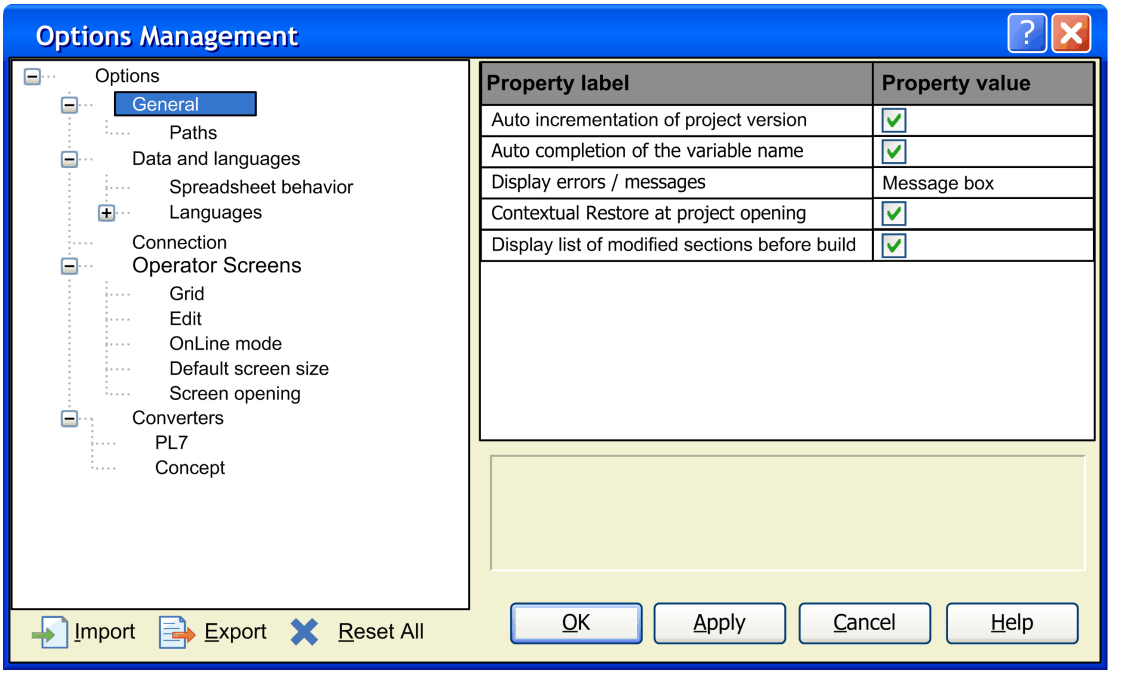
General Options

At a Glance

This configuration section, accessible from **Tools > Options > General**, contains a set of general options.

General Options Dialog Box

Representation of the **General** section:



Description

Elements of the **General** section:

Fields	Description
General	
Auto incrementation of project version	When this option is selected, the project version is automatically incremented on each build.
Auto completion of the variable name	This option is useful when you start the data selection to enter the name of a variable: for example, a pin of a block in FBD. When it is selected, you can enter the first characters of a variable to select one of these that matches with the entered string. If the entered string has been already used before, the auto completion feature completes the string with the last variable that was used.
Display errors / messages	If **Message box** is selected, all detected error messages are displayed in dialog boxes.

Fields	Description
	If **Output window** is selected, all detected error messages are displayed in the output window.
Contextual Restore at project opening	If this box is selected, the windows have the same arrangement whereas the project is opened or closed. A .ZTX file is generated, which saves the context.
Display list of modified sections before build	When this box is checked, the modified sections are listed in the **List of modified sections** dialog box after activating the **Build Changes** command in the **Build** menu.
General > Paths	
Project path	Path where the .STU project files are saved or from where they are retrieved when opening.
Import/Export file path	Path where the import/export source files are kept: for example, .XEF files.
XVM path	Path where the .XVM file is stored. This file is automatically generated to back up the variables of the project every time this one is transferred to the PLC. The default path is the same as the default **Project path**.
Project settings templates path	Path where the **Project Settings** templates are located.
Project settings template file	Project settings template file that is proposed in the **Settings File** during the **New Project**, page 74 creation.
Custom Library path	Path where the custom libraries are located. Changing the custom library path is taken into account at next open project or new project.
Working drive	Workspace used by the software.
Log files path	Path where the workspace is used by the software.

NOTE: The libraries provided by Schneider Electric are located in a Control Expert system folder. You cannot change the location of these libraries.

To use a previously created custom library after changing the **Custom Library Path**, move the library file to the new location.

Data and Languages

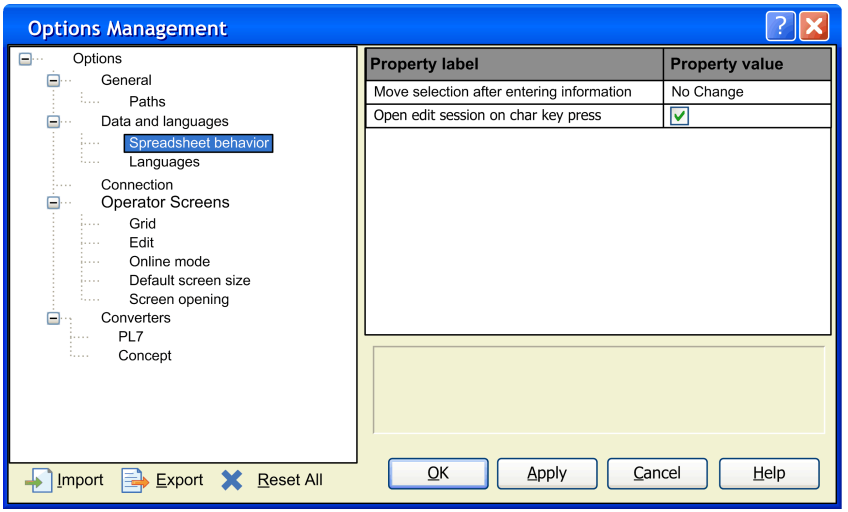
Introduction

The **Data and Languages** section contains settings for entering data.

NOTE: The changes made in this section become active when the respective dialog box (for example, data editor) is closed and then opened again.

Representation

Representation of the **Data and Languages** section:



Items of the **Data and Languages** section:

Element	Description
Spreadsheet behavior	
Move selection after entering information	This options button allows specifying which neighboring cell becomes active when the **Enter** key is pressed.
Open edit session on char key press	If this box is checked, the input mode is automatically opened for the currently active cell when you press a character key. The contents of the active cell are overwritten with the new contents.
Languages	
Automatically assign a variable to a new graphical object	If this box is checked, the properties dialog box (for example, for contacts, coils, steps, transitions) or the FFB input assistant (for example, for functions or function blocks) is opened. It assigns the formal parameter of the object when you place a graphical object.
Reuse windows on refine	If this box is checked, then: • Only 1 window is used for **Refine** command on DFBs • Only 1 window is used for **Refine** command on actions • Only 1 window is used for **Refine** command on transitions.

Element	Description
	If this box is not checked, for each **Refine** command on DFBs, actions or transitions, a new window is opened.
LL984	
Show latched coils as	Use this list to select letter L (by default) or letter M that is displayed inside a latched coil. **NOTE:** Modifying this setting leads to a refresh of the editor window of all opened LL984 networks.
Reuse 984 editor window when network is opened	If this box is checked, only 1 window is used for **Open** commands on LL984 networks. If this box is not checked, for each **Open** command on an LL984 network, a new window is opened. **NOTE:** Modifying this setting will only take effect after next time an LL984 editor is opened.
Mnemonics	The **Property label** column lists all contacts and coils and an assortment of function blocks available for LL984 networks. In the **Property value** column, you can enter a mnemonic for each listed element. A mnemonic can consist of up to 4 alphanumeric signs. Refer to Using Mnemonics, page 1157.
LD	
Mnemonics	The **Property label** column lists an assortment of objects available for LD sections. In the **Property value** column, you can enter a mnemonic for each listed element. A mnemonic can consist of up to 4 alphanumeric signs. Refer to Using Mnemonics, page 769.

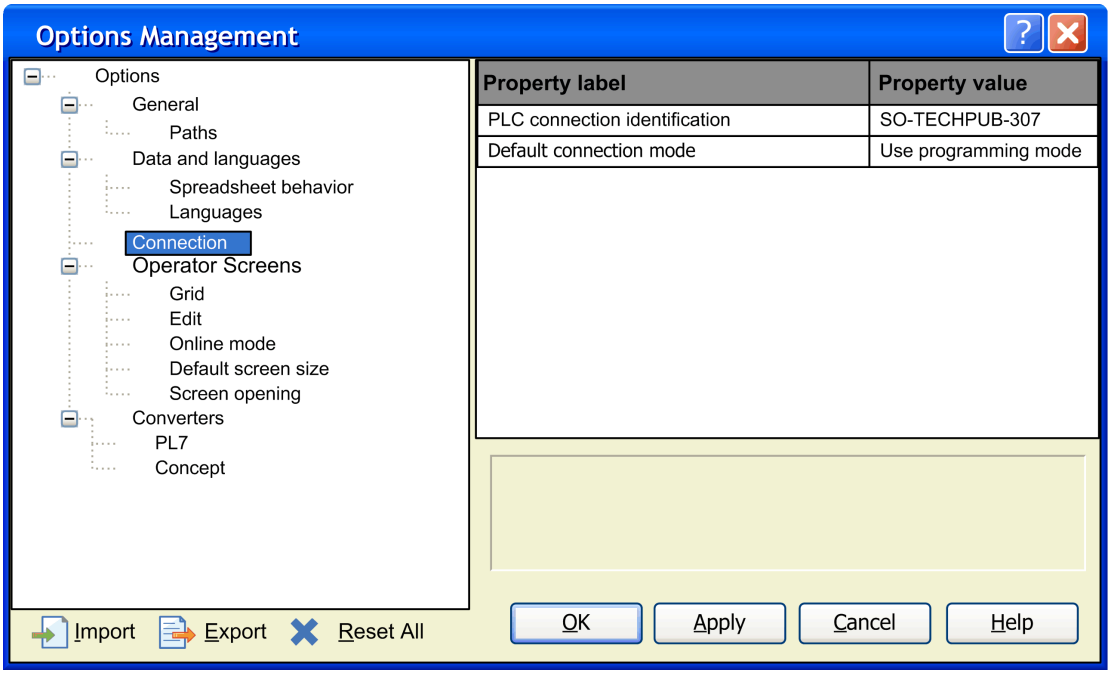
Connection Options

At a Glance

This configuration section, accessible from **Tools > Options > Connection**, contains the options relating to the connection, page 83 of the terminal to the PLC.

Connection Options Dialog Box

Representation of the **Connection** section:



Description

Items of the **Connection** section:

Fields	Description
PLC connection identification	When several terminals simultaneously access the same PLC, this information can be used to identify which terminal holds the PLC reservation. By default, the network name of the terminal is proposed; this name can be changed.
Default connection mode	If **Use programming mode** is selected, the terminal connects, by default, in programming mode. If **Ask for connection mode** is selected, the terminal asks, on each connection to a PLC, in which mode the connection is to be made: programming mode or monitoring mode. If **Use monitoring mode** is selected, the terminal connects, by default, in monitoring mode (read-only access). Note: In monitoring mode links between EFs are not animated. If **Mandatory programming mode** is selected, the terminal can only connect in programming mode. If the PLC is reserved, a connection refusal then follows.

The Operator Screens Option

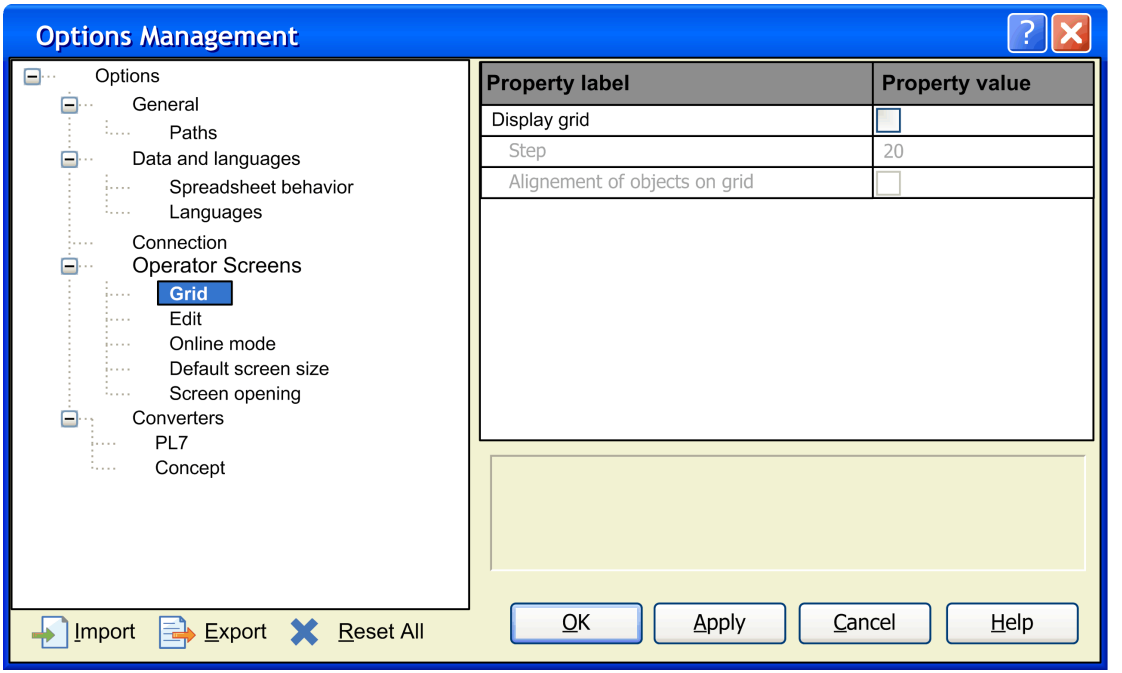
At a Glance

The configuration section, accessible from **Tools > Options > Operator Screens**, is used to:

- Enable or prevent use of the magnetic grid,
- Display the mouse position,
- Enable edit in online mode,
- Enable selection of an object in online mode,
- Display or hide the vertical scroll bar,
- Display or hide the horizontal scroll bar,
- Specify the default size of new operator screens.

Representation

Representation of the register section **Operator Screens**:



Elements of the register section **Operator Screens**

Field	Description
Grid	
Display grid	When this box is checked, the grid is displayed according to the step defined.
Step	Used to define the step of the grid, i.e. the space between the horizontal and vertical lines of the grid. From 5 to 100 (steps of 5 pixels).
Alignment of objects on grid	When this box is checked, the objects are automatically positioned on the points of the grid when they are handled (creation, size-modification, movement, etc.). If this box is not checked, the objects are not aligned on the grid.
Edit	
Display mouse position	When this box is checked, the position of the mouse is displayed in the left side of the status bar. If an object is selected, the display indicates the position of its top left corner, its dimensions, and its animation settings (if they exist), see the following example:

Field	Description
	[x:60,y:180 – cx:100,cy:60][Text][Animation : %MW100]
Vertical scroll bar	When this box is checked, the vertical scroll bar is displayed when the screen height is larger than the current window.
Horizontal scroll bar	When this box is checked, the horizontal scroll bar is displayed when the screen width is larger than the current window.
Online mode	
Enable edit in online mode	When this box is checked, this authorizes the editing of screens that are not currently controlled by the PLC in online mode (creation, modification, and deletion of screens). **NOTE:** If the screens are controlled by the PLC **Description**, page 592, you cannot edit them in Online Mode even if the **Enable edit in online mode** is checked.
Enable selection in online mode	When this box is checked, this enables the selection of an object in online mode.
Default screen size	
Width	These two fields allow the definition of the default size of new screens. The values entered do not change the size of screens already created.
Height	
Screen opening	
Screen opening	If the Mono-Instance box is checked, when a new screen opens, the latter replaces the previous one in the display window. If the Multi-Instance box is checked, each time a new screen is opened, the screen is displayed in a new window. In online mode, if several screens are opened simultaneously, this can reduce performance.

Converters

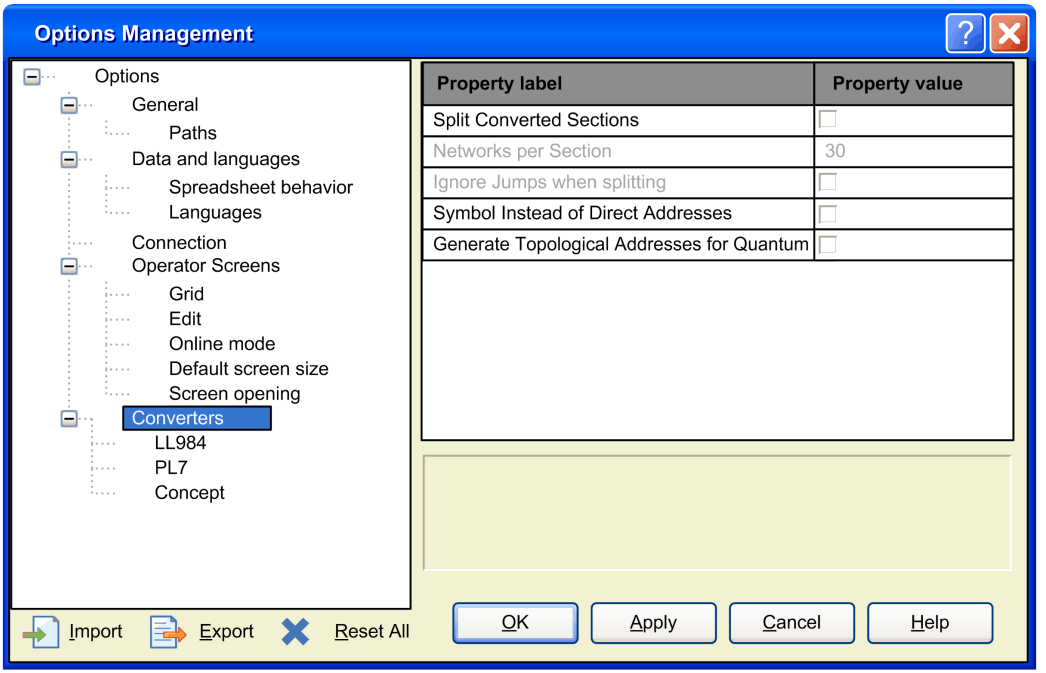
Introduction

This configuration section, accessed from **Tools > Options > Converters**, contains options for the conversion of legacy projects to Control Expert.

The section is only available for Quantum, Premium, and Atrium.

Representation

Representation of the register section **Converters**:



Elements of the register section **Converters**:

Element	Description
Converters	
Split Converted Sections	This separates ProWORX 32, Modsoft, and Concept LL984 segments into smaller units. Separation is only applied to MAST task sections. Grafcet and Post, FAST task sections and subroutines must be present as a complete unit and cannot be separated into smaller sections: • When this check box is enabled, the options **Networks per Section** and **Ignore Jumps when splitting** are enabled and can be modified. • When this check box is disabled, the options **Networks per Section** and **Ignore Jumps when splitting** are disabled. The separation of segments into smaller units does not take place, if a jump is present in the segment and its target has not yet been reached, and becomes active again after the jump target.
Networks per Section	Allows the definition of the number of networks in a section. From 1 to 30 networks per section.

Element	Description
Ignore Jumps when splitting	A section may contain jumps to labels. A jump is located inside the section. The converter takes jumps inside the section into account since it is not allowed to have a jump from one section to another. Depending on how the jump is validated inside the section (active or not), the automatic splitting at network bounds is inhibited (active). The feature of recognizing such jump scopes can be suppressed by this check box. When this check box is enabled, automatic section splitting always takes place regardless of jumps. Therefore many error messages concerning missing jump targets are generated at analyze time.
Symbols Instead of Direct Addresses	• When this check box is enabled, discrete variables are converted with their symbols instead of the direct addresses. When there is no symbol in the source application, a symbol is created automatically (for example, %IO.2.3.4 -> IO_2_3_4). • When this check box is disabled, the variables are converted with their direct addresses.
Generate Topological Addresses for Quantum	• When this check box is enabled, a topological address is created (for example, %M\2.3\4.5.6). • When this check box is disabled, a **flat address** is generated (for example, % M1234). During the conversion, the Concept addresses are converted into internal flat addresses 1:1. However, if topological addressing is desired, the flat address is searched for in the I/O address tables of the I/O modules. If the associated address is found, the address of the module link point is determined and linked with the address of the I/O module to a topological address. **NOTE:** Flat addressing is only possible for Quantum and for Modicon M340. For Premium, topological addressing is always used regardless of the option.
Converters > LL984 These options are present only if LL984 converter is installed.	
Suppress Schedule	When this check box is enabled, the association of I/O drops (Remote I/O) is suppressed during conversion. This can be done to avoid analyzing errors in Control Expert.
Suppress Graphs	When this check box is enabled, networks only containing links are erased during conversion. This can be done to avoid analyzing errors in Control Expert.
Attach coils to open ends	When this check box is enabled, coils without assigned variables are positioned at the open ends automatically. This can be done to avoid analyzing errors in Control Expert.
Generate ConvError Hints	• When this check box is enabled, **ConvError objects** are generated in the LD programs during the conversion to draw attention to special issues.

Element	Description
	• When this check box is disabled, **ConvError objects** are **not** generated in the LD programs.
Include 984 Execution Corrections	• When this check box is enabled, the converter modifies the section so that the result in Control Expert is the same as in ProWORX 32, Modsoft, or Concept. • When this check box is disabled, **no ConvError objects** are generated during the conversion.
Unpowered LD objects to Left Power	When this check box is enabled, unpowered LD objects are connected to the left power rail automatically with a normally opened contact without assigned variable.
Comment Generation Rule	When this check box is enabled, is possible to composite a variable comment from different source strings. A blank character is generated between all the different source strings.
Symbol Generation Rule	When this check box is enabled, is possible to composite a variable comment from different source strings. An underline character is generated between all the different source strings. Possible parameters for the string generation rules are: • **Sx**: short comments while x could be 1…4, • **Dx**: descriptors while x could be 1...9, • **L**: long comment, • **F**: flat address as string without % character. **NOTE:** Variable name source priority: • If Symbol is defined for a State Ram Data, it is used as variable name, • Else, if ISA Symbol is defined for a State Ram Data, it is used as variable name, • Else, a data name is generated with the **Symbol Generation Rule**.
Max. Symbol Length	Limitation of the generated symbol length.
Map Compact to Premium	• When this check box is enabled, the Compact modules are converted to Control Expert Premium modules. • When this check box is disabled, Concept Compact applications are converted to Modicon M340 applications.
Converters > PL7	
Transition Optimization	• When this check box is enabled, in order to improve the performances at execution time, some SFC transitions are converted to variables. • When this check box is **not** enabled, the transitions are converted as they were in PL7.
Step State Explicit Definition	

Element	Description
	• If this check box is enabled, the macro steps that are not used on the PL7 program are declared on the Control Expert project when the conversion of PL7 application to Control Expert is done. • If this check box is disabled, the macro steps that are not used on the PL7 program are not converted on the Control Expert project.
Symbol As Step Name	• When this check box is enabled, the PL7 steps %Xi are converted with the PL7 step symbol. For example, on PL7 %X2 with the symbol Motor_1 is converted into Motor_1. • When this check box is **not** enabled, the PL7 steps %Xi are converted on X_i. For example on PL7 %X2 is converted into X_2.
Converters > Concept	
Variables at empty pins	• When this check box is enabled, empty link points are filled with variables created by the converter. • When this check box is disabled, empty link points are **not** filled with variables created by the converter. In Control Expert, it is necessary to fill provided inputs and outputs for derived data types or I/O parameters (this is not necessary in Concept). If these types are not generic, the converter fills these initially empty inputs and outputs with variables created by the converter.
Disable FB Output Multi-Assignment detection	• When this check box is enabled, the converter disables the multiple assignment check. • When this check box is disabled, the converter enables the multiple assignment check. Control Expert and Concept behavior differ on DFB/FB output parameters in FBD or LD sections. This different behavior exists with elementary variables or derived data type components. The converter searches all potential locations where such elementary variables or derived data type components are written and marks these locations with a Converter Error Hint. Then, you can check the affected code concerning different behavior and if necessary modify the application. If the variable is written only once, no Converter Error Hint is generated.
Generate Generic EFs	• When this check box is enabled, the converter removes the added data type from the name of the function. • When this check box is disabled, the converter leaves the added data enter the name of the function. There are only a few generic elementary functions in Concept (EFs) for example, MOVE, SEL, MUX. With many other functions, the elementary data type is added to the name of the function. In Control Expert, many of these functions are used without the elementary data type added to the name (as defined in IEC 61131). The converter removes the added data type from the name of the function.

Element	Description
	In some cases, the use of generic functions in Control Expert leads to analytic errors. In these cases, disable the check box.
Separate LD networks	• When this check box is enabled, recognized networks are moved vertically, which prevents overlapping. • When this check box is disabled, recognized networks are **not** vertically moved. The original vertical arrangement of the graphics is maintained, but error messages may occur due to overlapping. IEC LD sections contain many independent graphic areas (networks). During the conversion of IEC LD sections, additional columns are added to the networks to avoid undesired automatically generated links in Control Expert. If the additionally inserted columns were to extend across the entire section, the original graphic would be modified too much. Therefore, the sections are divided in networks during the conversion and additional columns are only inserted for the associated network. Inserting additional columns may cause a network to exceed its maximum section width and it is then wrapped into the next line. If this causes networks to vertically overlap, the overlapping of the logic can lead to undesired automatic links in Control Expert.
Substitute Procedures in ST/IL	• When this check box is enabled, the instance name of the Concept call is replaced with the type name. • When this check box is disabled, a **DFB is created**, which accesses the procedure. Some EFBs from Concept are implemented as procedures in Control Expert without instance names.
Substitute %QD by %MF	• When this check box is enabled, **%QD** variables are converted to **%MF** variables. • When this check box is **not** enabled, **%QD** variables are converted to **MW** variables. Variables that are directly addressed in Concept with **%QD** can be initialized floating point constants or dual word constants. When floating point constants appear, this check box should be enabled.
LD Column Break	The number entered here determines the column after which a network is wrapped to the next column. Inserting additional columns may cause a network to exceed its maximum section width and it is then wrapped into the next line. For more information, see *Wrapping networks that are too wide* in the *Concept Converter* documentation.

Customize Dialog Box

Overview

This section describes the dialog box for customizing menu toolbars.

Customize Dialog Box

Call the Dialog Box:

You have the following options to call the dialog box:

- Execute the **Tools > Customize** command, or
- Use the **Toolbar** menu command from a toolbar shortcut menu or the toolbar itself.

Dialog Structure

The dialog box consists of two tabs:

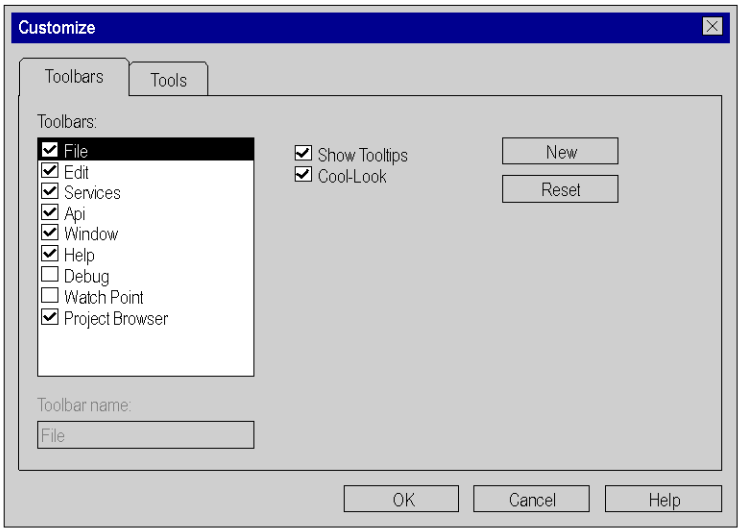
- **Toolbars**

  In this tab, you can show/hide toolbars, create new toolbars, and define the appearance of the toolbars.
- **Tools**

  In this tab, you can define the menu commands for starting external applications.

Toolbars Tab

Illustration of the **Toolbars** tab:

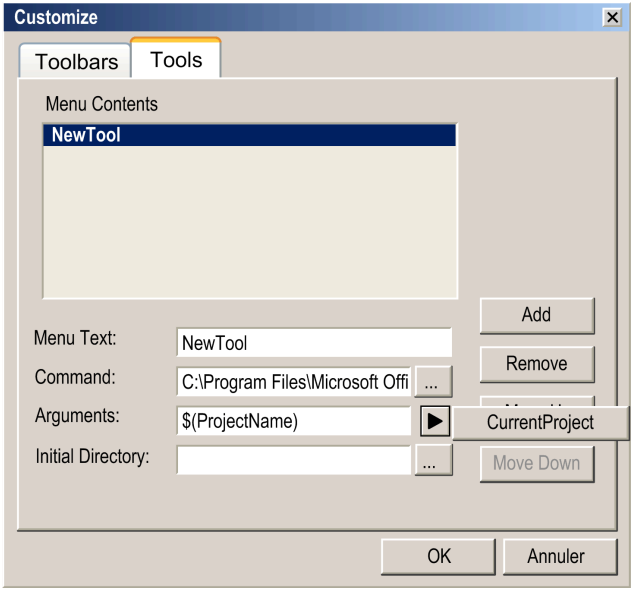


Elements of the **Toolbars** tab:

Element	Description
Toolbars	Displays the currently available toolbars. To display a toolbar, activate the respective check box. To hide a toolbar, deactivate the respective check box.
Toolbar name	Displays the name of the selected toolbar.
Show Tooltips	When you activate this check box, tooltips are displayed for each toolbar symbol.
Cool Look	When you activate this check box, the toolbar symbols are displayed in flat form.
New...	Use this button to open the dialog box for defining new toolbars, see also Creating your own toolbars, page 1840.
Reset	Use this button to restore the original state of the toolbar selected in the **Toolbars** list.

Tools Tab

Display of **Tools** tab:



Elements of the **Tools** tab:

Element	Description
Menu Contents	Displays the defined menu commands.
Menu Text	Enter the name of the new command, also see Starting external applications, page 1834.
Command	Enter the path of the file to be executed, also see Starting external applications, page 1834.
Arguments	Enter the arguments to be transferred while launching the application Starting external applications, page 1834. By clicking **arrow > CurrentProject**, the argument is `$(ProjectName)`. When the application is launched, `$(ProjectName)` is replaced with the complete STU file path of the opened application. **NOTE:** If the application has never been saved, the project file path is undefined.
Initial Directory	Enter the path to the working directory of the application Starting external applications, page 1834.
Add	Use this button to define a new menu command for starting an external application, see also Starting external applications, page 1834.
Remove	Use this button to delete the selected menu command.

Element	Description
Move Up	Use this button to move the selected menu command up by 1 position.
Move Down	Use this button to move the selected menu command down by 1 position.

Common Function for graphical language

What's in This Chapter

Using Bookmarks in Graphical Languages Sections 615
Display grid in a FBD/LD/SFC section 621
Zoom in a FBD/LD/SFC section 621
Tracking links in a FBD/LD/SFC section..... 624
Print Current Section in a FBD/LD/SFC section 625
Undo and redo changes in a FBD/LD section 626
Using connectors in a FBD/LD section..... 627
Inspect windows in a FBD/LD section 632
Tooltips for Variables in a FBD/LD/SFC Section..... 636
Replace Variables in a FBD/LD Section 638
Initialize search in a FBD/LD section..... 643
Initialize Animation Table in a FBD/LD section 644
Go to in a FBD/LD section 645
Go to dialog box in a FBD/LD section 647

Overview

This chapter describes the common menus and dialogs of the FBD, LD and SFC editors.

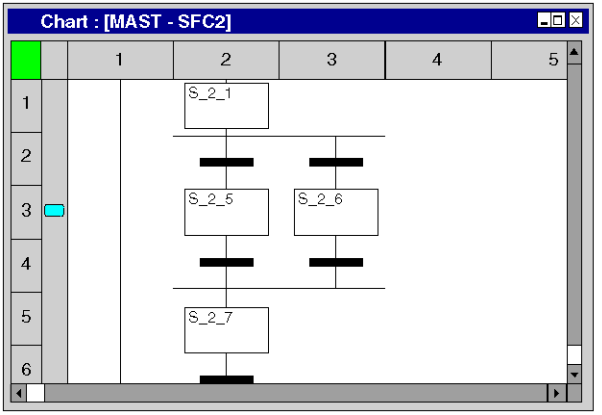
Using Bookmarks in Graphical Languages Sections

Introduction

Bookmarks are used to mark positions in the margin (gray column on the left edge of the section/network) and allow the marked positions to be easily found in different language sections/networks.

You can navigate through a list of bookmarks, displayed in the **Tools > Bookmarks** dialog.

An example of a bookmark in SFC:



The current bookmark is indicated using a horizontal arrow ( ).

Bookmark are displayed in different colors.

Created	Enabled/Disabled	Color
manually	enabled	blue
manually	disabled	light gray
automatically	enabled	gray
automatically	disabled	light gray

Setting Bookmarks

You have the following options to set a bookmark:

- Execute the command **Edit > Bookmarks > > Toggle Bookmark**. (There must not be any current bookmark in the selected area).

  The bookmark is inserted in the margin (gray column on the left edge of the section/ network) and is marked as the current bookmark.

- Press the **Ctrl+F2** key combination. (There must not be any current bookmark in the selected area).

  The bookmark is inserted in the margin (gray column on the left edge of the section/ network) and is marked as the current bookmark.

  or

- Place the mouse pointer in the margin (gray column on the left edge of the section/network) at the position to be marked and use **Toggle Bookmark** from the context menu for the margin (right mouse click).

  The bookmark is inserted at the mouse pointer and marked as the current bookmark.

Removing Single Bookmarks

You have the following options to remove a bookmark:

- Select the bookmark to be deleted as the current bookmark (e.g. with **Next Bookmark**) and use the **Edit > Bookmark > > Toggle Bookmark** command.
- Select the bookmark to be deleted as the current bookmark (e.g. with **Next Bookmark**) and press the **Ctrl+F2** key combination.

  or
- Place the mouse pointer on the bookmark to be removed and use **Toggle Bookmark** from the context menu of the bookmark (right mouse click).
- You can also remove a bookmark using the context menu of the **Bookmarks** dialog.

Removing All bookmarks

You have the following options to remove all bookmarks in the current section/network:

- Use the **Edit > Bookmark> > Clear all Bookmarks** menu command.

  or
- Press the **Ctrl+Shift+F2** key combination.
- You can also remove all bookmarks using the context menu of the **Bookmarks** dialog.

Searching for Bookmarks (Search Down)

You have the following options to search for bookmarks (search down):

- Use the **Edit > Bookmark> > Next Bookmark** menu command.
- Use the **Edit > Go to > Bookmark > Next** menu command.

  or
- Press the **F2** key.

If a current bookmark is selected, a jump is made to the next one and it is marked as the current bookmark.

The bookmarks are traversed according to their order of appearance in the **Bookmarks** dialog.

Searching for Bookmarks (Search Up)

You have the following options to search for bookmarks (search up):

- Use the **Edit > Bookmark> > Previous Bookmark** menu command.
- Use the **Edit > Go to > Bookmark > Previous** menu command.

  or
- Press the **Shift+F2** key combination.

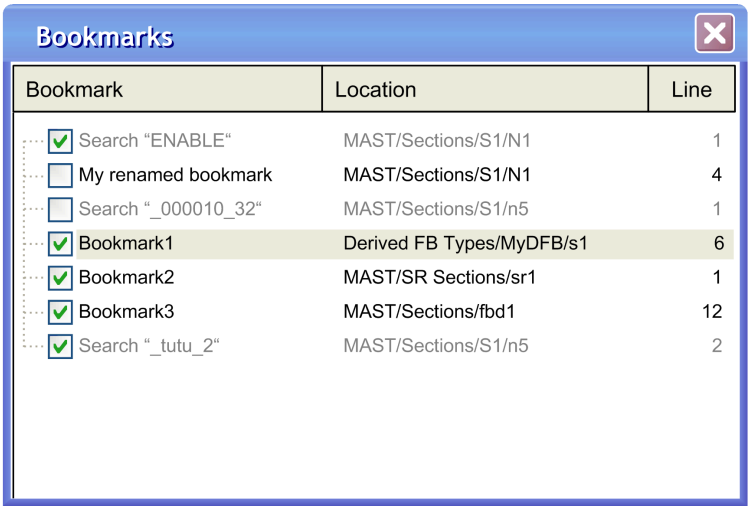
If a current bookmark is selected, a jump is made to the previous one and it is marked as the current bookmark.

The bookmarks are traversed according to their order of appearance in the **Bookmarks** dialog.

Bookmarks Dialog

To open the bookmarks dialog use **Tools > Bookmarks**.

A list of the project´s bookmarks is displayed in the **Bookmarks** dialog.



Element	Description
Bookmark column	List of the project´s bookmarks. Automatically created bookmarks (see below) are displayed in gray and can not be renamed.
Location column	Location of the bookmark (section, network or DFB) referring to the **Project Browser**.
Line column	Line in the section/network where the bookmark is located.
check boxes in the **Bookmark** column	Activate this check box to be able to jump to this bookmark by means of **Next Bookmark**/**Previous Bookmark**. By default for newly created bookmarks the check box is activated.
Delete/Delete all	Right click in the dialog and select from the context menu: • **Delete**: to delete the selected bookmark • **Delete all**: to delete all bookmarks in the project

Creating bookmarks is done in the language editors.

Deleting and renaming bookmarks can be done in the **Bookmarks** dialog, too.

NOTE: Renaming of bookmarks can only be done with created bookmarks. Automatically created bookmarks can not be renamed.

Bookmark Commands in Language Editors

The bookmark commands in language editors affect the **Bookmarks** dialog as follows:

- **Toggle Bookmark** creates a bookmark, visible also in the **Bookmarks** dialog.

  It will be appended to the list of bookmarks.
- **Next Bookmark** jumps to the next bookmark in the list of bookmarks
- **Previous Bookmark** jumps to the previous bookmark in the list of bookmarks.
- **Delete Bookmark** deletes the selected bookmark.
- **Delete all Bookmarks** deletes only the bookmarks inside the current section/network.

NOTE: The **Bookmark** tab of the **Edit > Go To** dialog only displays the bookmarks of the current section.

Naming Bookmarks

Created bookmarks are named as **Bookmark1**, **Bookmark2** and so on.

In the **Bookmarks** dialog you can rename the bookmarks.

Automatically Created Bookmarks

Automatically created bookmarks are named as **Search"<Search String>"**.

Automatically created bookmarks are displayed in gray and can not be renamed.

The maximum number of automatically created bookmarks is 30.

Automatic creation of another bookmark will insert the new bookmark as the first one in the bookmark list and will delete the bookmark at the end.

For automatically created bookmarks please refer to Tracing Variables / Addresses, page 1252.

Navigating to Bookmarks

In the **Bookmarks** dialog double-click a line in the list of bookmarks to open the respective section/network with the selected bookmark.

If the check box in front of a bookmark line is not activated, you can not jump to this bookmark by means of **Next Bookmark**/**Previous Bookmark**.

Saving Bookmarks

Bookmarks are saved with your application.

Deleting Bookmarked Lines

If you delete a section/network or a bookmarked line in a section/network, the corresponding bookmarks are deleted, too.

Undo will not restore the bookmarks.

Display grid in a FBD/LD/SFC section

Displaying the grid

The editor window has a grid background as default which splits the window into lines and columns.

Use the **View > Grid** menu command to show or hide the grid.

The active mode is indicated by way of a check-symbol before the menu command.

Zoom in a FBD/LD/SFC section

Introduction

Several Zoom functions are available to provide an optimal display of the elements in the current section (overview of the entire contents or display of details).

Zoom factor in % (FBD/LD/SFC)

The Zoom factors given in percentages increase or decrease the view of the section in proportion to its normal size. To see the section in its normal size, select **100%**.

NOTE: For LD sections, the column width can be adjusted using the |S| |M| |L| buttons, so that more text can be displayed keeping the same zoom factor (same line height).

The following Zoom factors are available:

- **50%**

  Decreases the section contents to half their normal size.
- **75%**

  Decreases the section contents to three-quarters of their normal size.
- **100%**

  Displays the section contents in normal size.
- **200%**

  Increases the section contents to double their normal size.

The menu command to set the Zoom factor is available:

- via the **View > Zoom** menu.
- with right click and menu **Zoom** for the section.

You can see which mode is active by way of a check-symbol before the menu command.

Custom... (FBD/LD/SFC)

This dialog box has two functions:

- Displaying the current Zoom factor as a percentage

  When the dialog box is opened, the **Zoom %** list displays the current Zoom factor.
- Defining any Zoom factor as a percentage

  You can define the Zoom factor by entering the Zoom factor as a percentage or by selecting a percentage from the list of already used Zoom factors.

The menu commands to set the user-defined Zoom factor is available:

- via the menu **View > Zoom > Custom...**.
- with right click and menu **Zoom > Custom...** for the section.

You can see which mode is active by way of a check-symbol before the menu command.

Zoom to Fit (FBD/LD/SFC)

For FBD and SFC sections the contents are minimized horizontally and vertically so that the whole section can be seen.

For LD sections the contents are minimized horizontally so that the whole section width can be seen. Adjusting the column width using the 

buttons double (|M|) or triple (|L|) the number of visible lines.

If the size of the section is changed, the section contents are automatically adapted to the new size.

The menu command to set the window size is available:

- via the menu **View > Zoom > Zoom to fit**.
- with right click and menu **Zoom > Zoom to fit** for the section.
- using the key combination **Ctrl+*** (* = multiplication sign of numeric pad)
- using the  symbol.

You can see which mode is active by way of a check-symbol before the menu command.

Zoom to selection (FBD/LD/SFC)

Decreases or increases the section contents so that the selected element(s) is (are) displayed as large as possible and in focus.

The menu command to zoom to the selection is available:

- via the menu **View > Zoom > Zoom to selection**.

You can see which mode is active by way of a check-symbol before the menu command.

Zoom out

To reduce current display by 25 %:

- select the menu **View > Zoom > Zoom out**.
- right click in the section and select **Zoom > Zoom out**.
- press the key combination **Ctrl**+**Minus (-)**.
- using the mouse: press **Ctrl**+**Rotate the mouse wheel**.
- click the  symbol.

Zoom in

To zoom in on current display by 25 %:

- select the menu **View > Zoom > Zoom in**.
- right click in the section and select **Zoom > Zoom out**.
- press the key combination **Ctrl**+**Plus (+)**.
- using the mouse: press **Ctrl**+**Rotate the mouse wheel**.
- click the  symbol.

Fullscreen

To display the application window in full-screen mode:

- select the **View > Full screen** menu.

- press the key combination **Ctrl**+**F8**.
- click the  symbol.

To end the full screen mode press **Esc** or use the toolbar menu.

Tracking links in a FBD/LD/SFC section

Introduction

There are different view functions available to provide optimum display of links (display source, display destination, display entire link).

Displaying the source of a link

Displaying the source (start point) of a link:

Step	Action
1	Select the link whose source you want to display.
2	Use the **Edit > Link > Go to Source** menu command. or Use the **Go to Source** menu command from the shortcut menu. **Result:** The focus of the section is changed to display the source (start point) of the link. The current zoom factor remains unchanged.

Displaying the destination of a link

Displaying the destination (end point) of a link:

Step	Action
1	Select the link whose destination you want to display.
2	Use the **Edit > Link > Go to destination** menu command. or Use the **Go to destination** menu command from the shortcut menu. **Result:** The focus of the section is changed to display the destination (end point) of the link. The current zoom factor remains unchanged.

Displaying the entire link

Displaying the entire link:

Step	Action
1	Select the link that should be displayed in its entirety.
2	Use the **Edit > Link > Zoom** menu command. or Use the **Zoom** menu command from the shortcut menu. **Result:** Decreases or increases the section contents so that the selected links are displayed as large as possible and in focus.

Print Current Section in a FBD/LD/SFC section

Introduction

There are the following ways to print the current section:

- Print the section using the menu command
- Print the section using the print documentation

NOTE: To view the breaks on the current section (for printing purpose), click **View > Page Break**. Click again to hide page breaks.

Print the section using the menu command

Carry out the following steps to print the current section using the menu command:

Step	Action
1	Open the print dialog using: • the menu command **File > Print**, • or the key combination **Ctrl+P** or • the symbol  .
2	Press **OK** to start printing the current section.

Print the section using the print documentation

Carry out the following steps to print the current section using the print documentation:

Step	Action
1	Double-click on **Documentation** in the project browser. **Result:** An overview of your system is open, see also Documentation, page 1660
2	Select **Program > Tasks > MAST/FAST/AUX > ...** to print the section.
3	Select the menu command **Print** from the sections context menu.
4	Press **OK** to start printing the current section.

Undo and redo changes in a FBD/LD section

Undo changes

You have the following options to undo the recent changes:

- Use the **Edit > Undo** menu command.
- Press the **Ctrl+Z** key combination.

  or
- Select the  symbol.

One change is undone each time the function is executed.

The function can be executed a maximum of 10 times.

Redo changes

Changes that have been undone using **Undo** can be restored:

- Use the **Edit > Redo** menu command.
- Press the **Ctrl**+**Y** key combination.

  or

- Select the  symbol.

One undone change is restored each time the function is executed.

The function can be executed a maximum of 10 times.

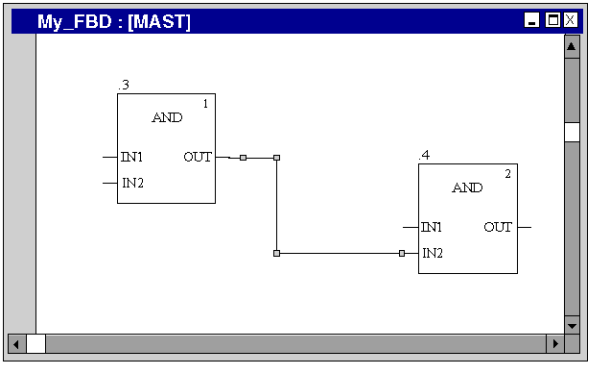
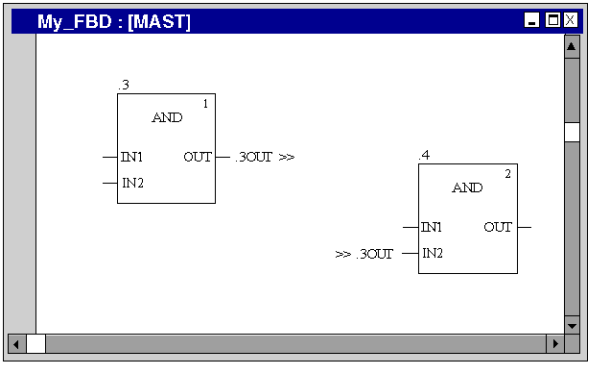
Using connectors in a FBD/LD section

Introduction

To avoid links overlapping each other, they can also be represented in the form of connectors, see also section Links (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.

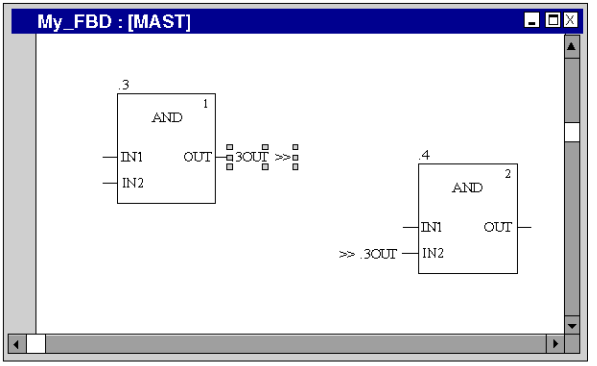
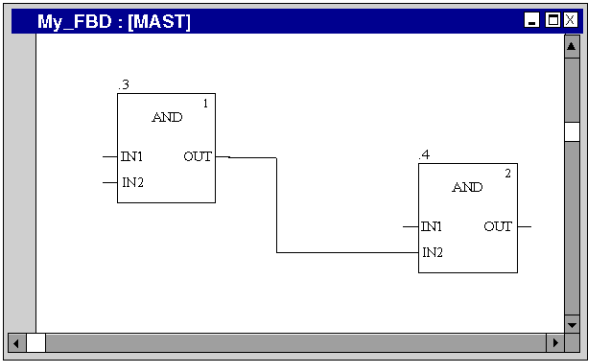
Displaying links as connectors

Displaying links as connectors:

Step	Action
1	Select the desired link. Example: 
2	Use the **Edit > Link > Show as Connector** menu command. or Use the **Show as Connector** menu command from the shortcut menu. **Result:** The link is displayed as a connector. Example: 

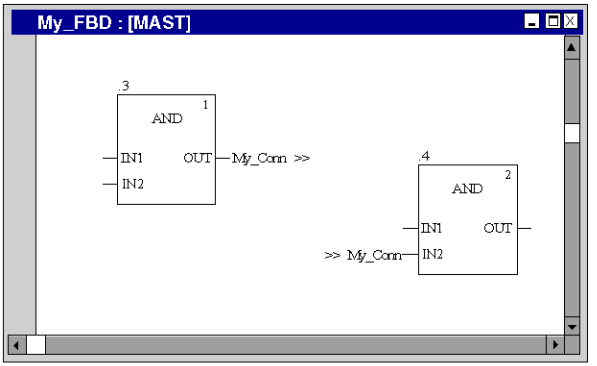
Displaying connectors as links

Displaying connectors as links:

Step	Action
1	Select the desired connector. Example: 
2	Use the **Show as Link** menu command from the shortcut menu. **Result:** The connector is displayed as a link. Example:  **Note:** Although the connector is now displayed as a link, the name of the connector is still occupied and cannot be used for another link. If the connector name is used for another link, the connector must be explicitly deleted, see Deleting connectors, page 630.

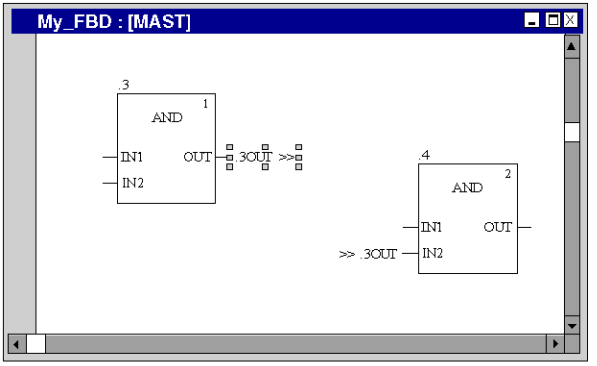
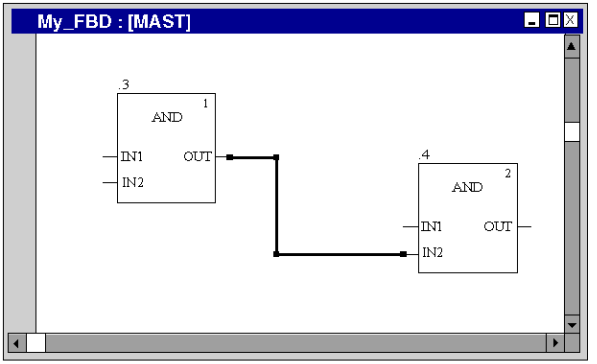
Renaming connector

Renaming connector:

Step	Action
1	Double-click the target connector (source). **Result:** The editor mode is activated.
2	Enter the new name. **Note:** The name must be unique within the current section and may not already be used by a connector, this also applies to connectors that are to be displayed as links. To enable an occupied connector name, the connector must be explicitly deleted, see Deleting connectors, page 630. Confirm the entry using **Enter**. **Result:** The connector (source and target) is renamed. Example: 

Deleting connectors

Deleting connectors:

Step	Action
1	Select the desired connector. Example: 
2	Use the **Show as Link** menu command from the shortcut menu. **Result:** The connector is displayed as a link. Example: 
3	Use the **Delete Connector** menu command from the shortcut menu. **Result:** The connector is deleted.

Inspect windows in a FBD/LD section

Introduction

In offline mode, the name, address (if available) and the comment (if available) of the associated variable are displayed in inspect windows.

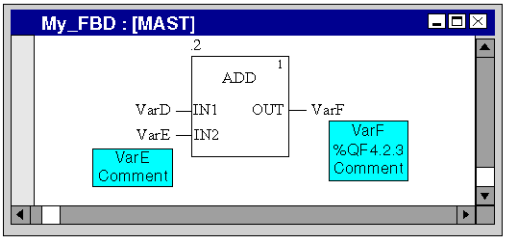
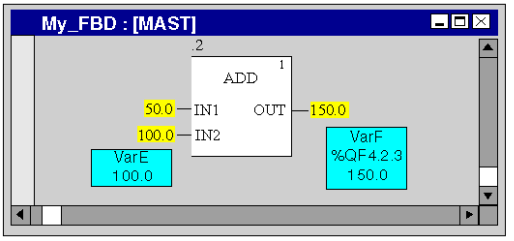
In online mode, the name, address (if available) and the current value of the variable are displayed in inspect windows.

The current values can be displayed:

- Decimal, e.g. `15`
- Binary, e.g. `00001111`
- Hexadecimal, e.g. `F`
- Floating point, e.g. `15.4`
- Exponent, e.g. `1.3e+003`
- Literal, e.g. `'otto'`

Generating an inspect window

Generating an inspect window:

Step	Action
1	**In FBD:** Select the desired FFB pin. **In LD:** Select the desired contact, the desired coil or the desired FFB pin.
2	You have the following options to generate inspect windows: • Use the **Edit > New > Inspect window** menu command. • Use the **Place Inspect** menu command from the shortcut menu. • Press the **F9** key. or • Select the  symbol. **Result:** The inspect window is generated. Example (offline):  Example (online): 

Resizing

Position the mouse pointer on the right or left edge of an inspect window (the correct position is indicated by the ↙ or ↘ mouse pointer symbol), click the left mouse button and drag the mouse right or left.

Changing the position

Position the mouse pointer on the inspect window, click the left mouse button and move the inspect window to the target position.

Anchoring the position

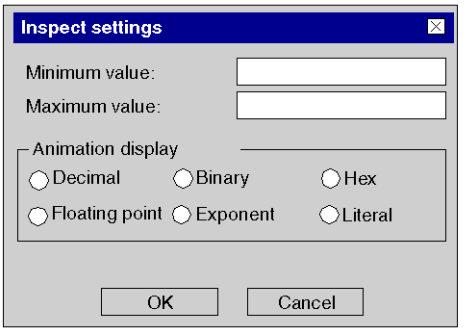
You can anchor all inspect windows of a section to their relative positions within the section (the inspect window stays in its onscreen position even if the page is scrolled), or to their absolute position (the inspect window is scrolled along with the rest of the section contents).

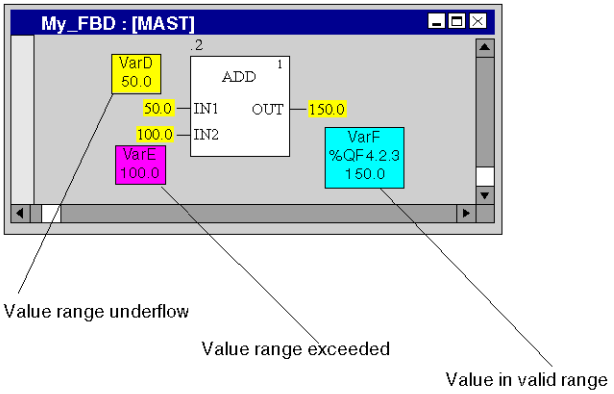
To switch between modes, select the **View > Floating Inspect Windows** menu command.

You can see which mode is active by way of a check-symbol in front of the menu command.

Setting a zone monitor

Setting a zone monitor:

Step	Action
1	Position the mouse pointer on the inspect window.
2	Use the **Settings** menu command from the shortcut menu. **Result:** The properties dialog box for the inspect window is opened. Example: 
3	Enter a minimum and a maximum value for the variable to be monitored.

Step	Action
4	Enter the format of the variable displayed in the section.
5	Confirm the entries with **OK**. **Result:** • If the current value is within the defined range, the inspect window is displayed in cyan. • If the current value is outside the defined range, the inspect window is displayed in yellow. • If the current value is above the defined range, the inspect window is displayed in magenta. Example: 

Hiding inspect windows

To hide and show the inspect windows select the **View > Hide Inspect Windows** menu command.

You can see which mode is active (hide inspect windows) by way of a check-symbol in front of the menu command.

Deleting inspect windows

Position the cursor on the inspect window to be deleted and select the **Delete Inspect** menu command from the shortcut menu.

Tooltips for Variables in a FBD/LD/SFC Section

Introduction

In offline and online mode the following information of a variable can be displayed by tooltip:

- type
- name
- address (if available)
- comment (if available)
- Alias of (if available)

Information is displayed for variables and expressions associated to:

- pins of FFBs (FBD/LD)
- contacts, coils, operation and comparison blocks (LD)
- transitions (SFC)

Concatenated Comments

To show concatenated comments of DDT expressions, set the option **Display complete comments from structure element** via **Tools > Project Settings** (for the current project) or via **Tools > Options** (for future projects).

Example, when the option is set.

DDT expression	Tooltip
ddt1.b1	ddt1.b1 DDT1 : ddt1 @ %MW100 | 'Comment for ddt1' BOOL : ddt1.b1 @ %MW100 | 'Comment for ddt1.Comment for b1'
ddt2.ddt1.b1	ddt2.ddt1.b1 DDT2 : ddt2 @ %MW200 | 'Comment for ddt2' BOOL : ddt2.ddt1.b1 @ %MW200 | 'Comment for ddt2.Comment for nested ddt1. Comment for nested b1'

Example, when the option is **not** set.

DDT expression	Tooltip
ddt1.b1	ddt1.b1 DDT1 : ddt1 @ %MW100 | 'Comment for ddt1' BOOL : ddt1.b1 @ %MW100 | 'Comment for b1'
ddt2.ddt1.b1	ddt2.ddt1.b1 DDT2 : ddt2 @ %MW200 | 'Comment for ddt2' BOOL : ddt2.ddt1.b1 @ %MW200 | 'Comment for nested b1'

Variable/Pin

Position the mouse pointer over a variable (e.g. `b1`) or expression to display type, name, address and comment of the variable.

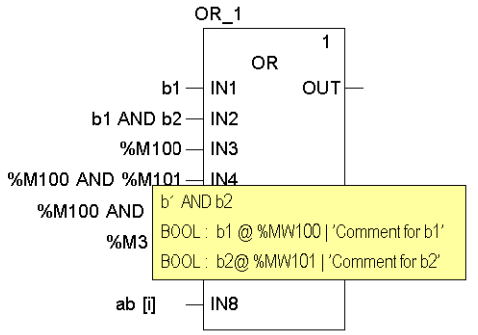
Position the mouse pointer over an FFB pin name (e.g. `IN1`) to display type, name and comment of the pin. In this case the information is taken from the FFB type and not from the FB instance.

Detected Error Tooltip

There are analyze errors and animation errors.

- Analyze errors are displayed in offline mode in a tooltip, when you position the mouse pointer over a connectable object (e.g. FFB, contact, coil etc.).
- Animation errors are displayed in online mode in a tooltip, when you position the mouse pointer over the `#Err!` value, which is shown in this case as animation value.

Using the Variable Tooltip

Step	Action
1	Position the mouse pointer over the variable or expression.
2	The tooltip is displayed. Example: 

Replace Variables in a FBD/LD Section

Overview

To replace variables assigned to elementary functions (EF) or to function blocks (EFB, DFB) in FBD/LD sections use the **Replace Variables** dialog box.

To find and replace variables connected to functions and/or function blocks you can use placeholders.

NOTE:

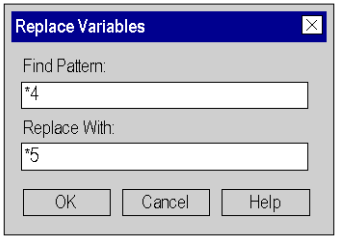
- Structure elements of multi-element variables (DDTs) cannot be replaced.
- If the original variable does not exist, the replacement variable is not possible.

Dialog Box

You have the following options to display the **Replace Variables** dialog box:

- Select the **Edit > Replace Variables** menu command.
- Select the **Replace Variables** command in the context menu (right-click on one or more EF/EFBs in the section).

Dialog box for replacing functions/function blocks using placeholders



Find and Replace Pattern

The find and replace patterns are applied to all variables connected to the selected functions/function blocks.

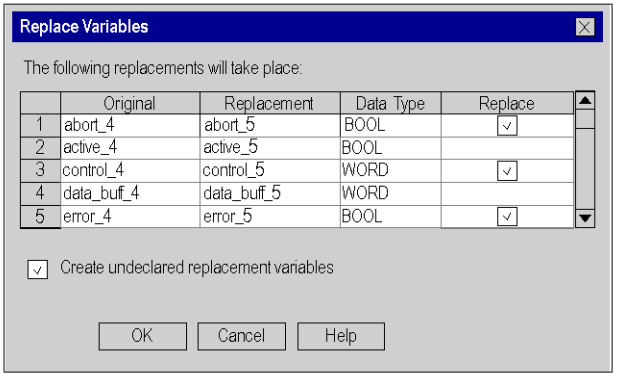
The following placeholders can be used:

- ***** to search/replace any string
- **?** to search/replace any character

Rules for find and replace patterns see below.

Replace Variables

All variables matching **Find Pattern** will be listed in a dialog box:



Elements of the dialog box

Element	Description
Original	displays the names of the found variables The entries in this column are **not** editable.
Replacement	displays the strings that will replace the names of the found variables The entries in this column are editable.
Data Type	displays the type of the original variable (not that on of the replacement variable, which could already exist and be of another type) The entries in this column are **not** editable.
Replace	provides the option to enable/disable the variable replacement via a check box By default **Replace** is enabled for all rows. The entries in this column are editable.
Create undeclared replacement variables	If this check box is selected, the replacement variables will be created in the Data Editor automatically. If such a replacement variable already exists in the Data Editor, the creation will be skipped. **Note:** If the replacement variable already exists, but has another type than the original one, since creation is skipped the replacement variable may or may not match the type of the connected pin afterwards. **Note:** If the check box is not selected, no variable creation will take place (some of the replacement variables may be undeclared afterwards).

Undo/Redo

Replace Variables supports Undo/Redo, which means that the original state will be recreated on Undo (of course without deleting the newly created replacement variables from Data Editor) and the state after replacement on Redo.

Online Mode

Replace Variables is supported in online mode.

Rules for Find and Replace Patterns

Rules for find and replace patterns:

- **?** can be used many times in pattern
- * can be used at maximum 2 times in pattern
- * can be first and last character in pattern only
- no * only and no ** can be used
- no ?* and no *? can be used (always another letter between them)
- numbers of * and ? in both (find and replace) pattern must be equal
- if two * and at least one ? are used the find and replace pattern must be structural identical (same length, same order of normal letters and placeholders)

Examples

Examples of simple replacements

Find Pattern:	Found Variable	Replace With:	Variable is Replaced With
abc123jmk	abc123jmk	abc223jmk	abc223jmk
abc123jmk	abc123jmk	abc124123jmk	abc124123jmk

Examples for replacements with constant characters

Find Pattern:	Found Variable	Replace With:	Variable is Replaced With
ab?1????k	abc123jmk	ab?2????k	abc223jmk
abc???12??mk	abc124123jmk	abc???022?4?mk	abc12402234jmk

Examples for replacements with constant series of characters

Find Pattern:	Found Variable	Replace With:	Variable is Replaced With
*123jmk	abc123jmk	*223jmk	abc223jmk
123	abc123jmk	*223*	abc223jmk
123	abc123jmk	*2234*	abc2234jmk

Examples of complex replacements

Find Pattern:	Found Variable	Replace With:	Variable is Replaced With
abc???12??m*	abc124123jmk	abc???022?4?m*	abc12402234jmk
*1???mk	abc123jmk	*2???mk	abc223jmk
*2??jmk	abc223jmk	*1240?2?4jmk	abc12402234jmk
1?3	abc123jmk	*2?3*	abc223jmk

Examples of array replacements

Find Pattern:	Found Variable	Replace With:	Variable is Replaced With
array[5*	array[5]	array[6*	array[6]
array5[5*	array5[5]	array5[6*	array5[6]
5	array5[5]	*6*	array6[5]
array5[i*	array5[i5]	array5[i6*	array5[i65]
5	array5[i5]	*6*	array6[i6]

Not Possible

The following complex replacements are **not** possible

Entry in text field search pattern	Found variable	Entry in text field Replace With
1?3	abc123jmk	*2?34*
1?3	abc123jmk	*02?34*
1241?3	abc124123jmk	*12402?34*
12?1?3	abc124123jmk	*12?02?34*

Initialize search in a FBD/LD section

Introduction

The **Initialize Search** function transfers the name of the selected element (search text) to the **Element** text box in the **Cross-References** window.

Selecting the search text

Process the following steps to select the search text:

Step	Action
1	Select the element to search for (search text). The following elements can be searched for: • Variable (To search for a variable, select the variable.) • EFB/DFB Instance (To search for an EFB/DFB instance, select the EFB/DFB.)
2	Use the **Services > Initialize Search** menu command. or Select the menu command **Initialize search** from the sections shortcut menu. or Press the **Ctrl+U** key combination. **Result:** The search text is transferred to the **Element** text box in the **Cross-References** window.

Initialize Animation Table in a FBD/LD section

Introduction

The **Initialize Animation Table** transfers the selected variables to the Animation table, page 1481.

Selecting the variables

Process the following steps to select variables:

Step	Action
1	Select the variables to transfer to the Animation table. The following items can be selected: • Variable (To transfer one or more variables, select the variable(s) desired). • FFB (To transfer all variables of an FFB, select the FFB). • Contacts, Coils, Operate and Compare Blocks (LD) (To transfer all variables associated to Contacts, Coils, Operate and Compare Blocks, select these objects).)
2	Use the **Services > Initialize Animation table** menu command. or Select the menu command **Animation Table** from the sections context menu. or Press the **Ctrl**+**T** key combination. **Result:** The variable(s) is/are transferred to the Animation table, page 1481.

Initialize Animation Table

Using **Initialize Animation Table**(**Ctrl**+**T**) you create an animation table with a default name e.g. `Table[FBD Editor – Mixer : [MAST]]`. You can rename the table.

All the variables you selected are listed in the animation table.

Selecting a new set of variables and using **Initialize Animation Table** again, all new variables are added to this animation table (if you did not rename it).

Initialize New Animation Table

Using **Initialize New Animation Table** (**Ctrl**+**Shift**+**T**) you create a new animation table with an indexed default name e.g. `Table[FBD Editor - Mixer : [MAST]1]`. You can rename the table.

All the variables you selected are listed in the table.

But if you click anywhere in the section and do not select any variable, you create a new empty animation table.

Every time you use **Initialize New Animation Table** you create a new animation table.

Go to in a FBD/LD section

Introduction

The **Go to** function is used to jump to:

- a specific location in the current section,
- a specific bookmark in the current section or
- a specific mark in the current section.

Open the dialog

The following options are available to open the **Go to** dialog box.

- Use the menu command **Edit > Go to**,
- select the menu command **Go to** from the sections shortcut menu,
- Press the key combination **Ctrl**+**G**.

  or

- Choose the  symbol.

Go to a location

Go to a certain position in the current section by carrying out the following steps:

Step	Action
1	Open, page 645 the **Go to** dialog box.
2	Select the **Position** tab.
3	Enter the line or column number.
4	Confirm the entry using the command button **Go to**. **Result:** The desired position is displayed.

Go to a bookmark

Go to a certain bookmark in the current section by carrying out the following steps:

Step	Action
1	Open, page 645 the **Go to** dialog box.
2	Choose the**Actions**tab.
3	Select a bookmark in the list.
4	Double click on the selected bookmark or press the **Go to** command button. **Result:** The bookmark is shown and marked as the current bookmark, page 615.

Go to a location

Go to a certain location in the current section by carrying out the following steps:

Step	Action
1	Open, page 645 the **Go to** dialog box.
2	Select the **Location**tab.
3	Select a location in the list.
4	Double click on the selected location or press the **Go to** command button. **Result:** The desired location is displayed.

Go to dialog box in a FBD/LD section

Call the dialog box:

see Open the dialog, page 645

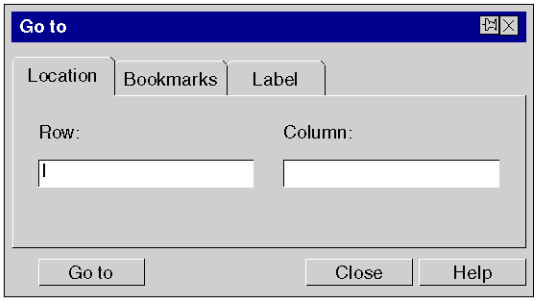
Call the dialog box

The **Go to**dialog box is made up of three tabs:

- **Location**
  Go to a certain position in the current section using this tab:
- **Bookmarks**
  Go to a certain bookmark in the current section using this tab:
- **Label**
  Go to a certain location in the current section using this tab:

Position tab

Representation of the tab **Position**:



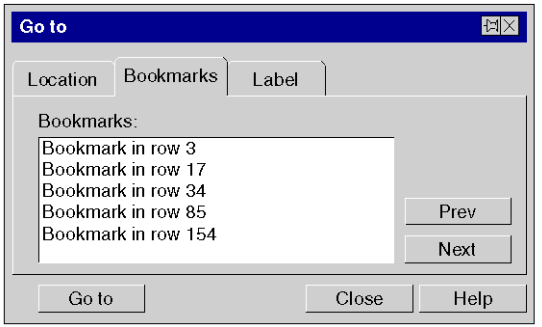
Elements of the **Position** tab

Element	Description
	For symbols that are not printed, the dialog box is closed after every search. For printed symbols the dialog box remains open until you close it.
Row	The number of the row to be displayed can be entered in this text box.

Element	Description
Column	The number of the column to be displayed can be entered in this text box.
Go to	Display the current position using the button desired position, page 646.
Close	Use this button to close the dialog.
Help	Use this button to call up the dialog help.

Register tab Bookmarks

Representation of the**Bookmark** tab:



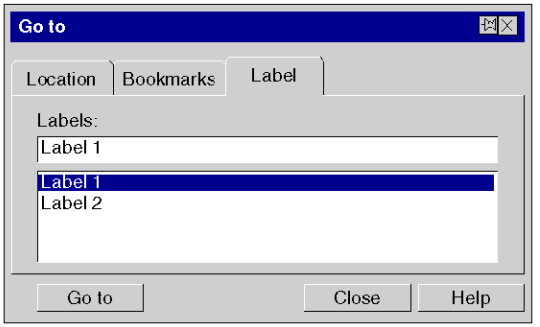
Elements of the tab **Bookmark**:

Element	Description
 	For symbols that are not printed, the dialog box is closed after every search. For printed symbols the dialog box remains open until you close it.
Bookmark:	Select the bookmark to be displayed from this list box.
Prev	If a current bookmark, page 615 is selected, a jump is made to the previous one and the last one is marked as the current bookmark. If no current bookmark is selected, the margin is search from the bottom to the top for the next bookmark, which is marked as the current bookmark. When the start of the section is reached, the search starts again from the bottom.
Next	If a current bookmark, page 615 is selected, a jump is made to the previous one and the last one is marked as the next bookmark. If no current bookmark is selected, the margin is search from top to bottom for the next bookmark, which is marked as the current bookmark. When the end of the section is reached, the search starts again from the top.

Element	Description
Go to	Display the current position using the button desired bookmark, page 646.
Close	Use this button to close the dialog.
Help	Use this button to call up the dialog help.

Register tab Location

Representation of the tab **Label**:



Elements of the tab **Bookmark**:

Element	Description
	For symbols that are not printed, the dialog box is closed after every search. For printed symbols the dialog box remains open until you close it.
Locations:	Select the location to be displayed from this text box. You can enter the name of the location in the text box with the keyboard or using the drop-down list by clicking with the mouse.
Go to	Display the current position using the button desired location, page 646.
Close	Use this button to close the dialog.
Help	Use this button to call up the dialog help.

FBD Editor

What's in This Chapter

Creating a program with FBD programming language 651
Editing FFBs 669
Returning from a Subroutine or DFB 706
Calling a Subroutine 708
Jumps within the Current Section 713
Definition of the Jump Target (Jump Labels) 718
Editing Links 723
Entering Comments 737
Online functions 739
FBD Reference Data Type 739
Export/Import 740
Customize FBD Editor 741

Overview

This chapter describes the menus and dialogs of the FBD editor.

For a description of the syntax for the FBD programming language, please refer to the Function Block Diagram FBD (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) chapter in the Reference manual.

Creating a program with FBD programming language

Overview

This section describes the basics for creating a program in the FBD programming language.

Structure of an FBD program (function block language)

Properties of an FBD program

Properties of an FBD program:

- FBD sections have a grid behind them.
- A grid section consists of ten grid units. A grid unit is the smallest possible space between two objects in an FBD section.
- An FBD section can be configured in number of cells (horizontal grid coordinates and vertical grid coordinates).
- The FBD programming language is not cell oriented but the objects are still aligned with the grid units.
- The execution order is determined by the position of the FFBs in the section (executed from left to right and from top to bottom). If the FFBs are connected to a network using graphical links, the execution order is determined by the signal flow, see also section Execution order of the FFBs (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.

  The execution order can be influenced in several ways, see also section Changing the execution order (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.
- A syntax and semantics check, page 654 is performed immediately after the statement is entered. The result of this check is displayed in colored text and objects.
- Syntactically or semantically incorrect sections can be saved.

FBD program objects

The objects of the programming language FBD (Function Block Diagram) help to divide a section into a number of:

- EFs and EFBs, page 674 (Elementary Functions and Elementary Function Blocks),
- DFBs, page 674 (Derived Function Blocks) and
- Procedures, page 674

These objects, combined under the name FFBs, can be linked with each other by:

- Links, page 723 or
- Actual Parameters, page 679.

Comments for the logic of the program can be added using text objects, page 737.

Edit and view functions

Edit and view functions of the FBD Editor:

- Selecting objects, page 656
- Deleting objects, page 658
- Cutting, page 659, copying, page 659 and pasting, page 660 objects

- Moving, page 662 objects (also between different FBD/LD sections)
- Undo, page 626 and Redo, page 627
- Using bookmarks, page 615
- Searching and replacing, page 1239 variables and function blocks and DFB instances.

Input assistants

Input assistants for the FBD Editor:

- Function input assistants, page 689 for functions, function blocks and procedures

Online functions

Online functions of the FBD Editor:

- Display the current value, page 632.
- Setting breakpoints, page 1430
- Setting watchpoints, page 1436
- Step by Step, page 1433

Creating an FBD Program

Creating an FBD program

Carry out the following procedures to create an FBD program:

Step	Action
1	Creating an FBD section., page 502
2	Place the desired FFB in the section, see also Calling an FFB via the data selection, page 674.
3	Assign the necessary current parameters to the FFB, also see Assign actual parameters, page 679. **Note:** As soon as the text is entered several checks are made, such as syntax/semantic error checks, correct spelling of variable names etc. A detailed description of syntax is found in chapter Function Block Diagram FBD (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual. The results of the checks are indicated by a color folder, see also Syntax and Semantics Check During Programming, page 654.
4	Create a graphical link between FFBs, also see Placing a Link, page 724.

Syntax and Semantics Check During Programming

Introduction

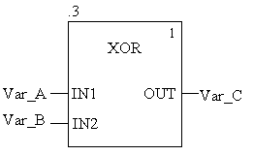
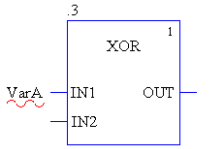
Syntax and semantics check is performed directly when creating the program.

The result of this check is displayed in three formats:

- directly in the program section with colored text
- as tooltip if the cursor is placed on incorrect text.
- In the output window, if **Generate > Analyze** is selected.

Representation

Meaning of colors and labels:

Color	Description	Example
Black	Syntax and semantics are correct	
Blue	Possible causes: • Associated variable is not declared • Data type of the variable does not match the data type of the pin	
Red wavy line	Erroneous text, e.g. variables which are not declared, variables with an incorrect data type, ...	

Tooltips

If the cursor is placed over an incorrect object or incorrect text, the tooltip is displayed with a brief description of the cause of the error. It will also be displayed in the output window after the analysis.

Error Message in the Output Window

Error message: `Accessed object is not a function block.`

Cause of error	Error Correction
Name of the function block was already used for a variable.	Rename the already used variable.

Navigating with the keyboard

Using the keyboard

The following keys and key combinations are provided for navigation:

Key combinations	Motion
Cursor left	Moves the cursor to the left
Cursor right	Moves the cursor to the right
Cursor up	Moves the cursor up
Cursor down	Moves the cursor down
Ctrl+Cursor left	Moves the cursor one grid point to the left
Ctrl+Cursor right	Moves the cursor one grid point to the right
Ctrl+Cursor up	Moves the cursor one grid point up
Ctrl+Cursor down	Moves the cursor one grid point down
Home	Shows the beginning of the current line
End	Shows the end of the current line
Ctrl+Home	Shows the top left cell of the section (the position of the cursor is not affected by this function)
Ctrl+End	Shows the bottom right cell of the section (the position of the cursor is not affected by this function)
Page Up	Scrolls one page up (the position of the cursor is not affected by this function)
Page Down	Scrolls one page down (the position of the cursor is not affected by this function)
Ctrl+Page Up	Scrolls one page left (the position of the cursor is not affected by this function)

Key combinations	Motion
Ctrl+Page Down	Scrolls one page to the right (the position of the cursor is not affected by this function)
Ctrl+Alt+Page Up	Displays the previous section (including via the **View > Previous Section**menu).
Ctrl+Alt+Page Down	Displays the next section (including via the **View > Next Section**menu).
Spacebar	Selects the object at the cursor position.
Ctrl + space bar	Adds the object at the cursor position to the current selection (multiple selection).
Enter	Opens the properties dialog box for the selected pin.
Alt+Enter	Opens the properties dialog box for the selected object.

Selecting objects

Select and placement mode

Objects are selected in select mode.

Select mode can be activated using:

- the **Edit > Select Mode** menu command
- the  symbol

  or
- the **Esc** key

The  cursor symbol indicates that select mode is active.

Selecting an object

Selecting an object:

Using the mouse	Using the keyboard
Left-click the object you want to select.	1. Move the cursor to the object to be selected, see also Navigating with the keyboard, page 655. 2. Press the **spacebar**.

Selecting several objects

Selecting several objects:

Using the mouse	Using the keyboard
1. Click the left mouse button and keep it pressed. 2. Drag the mouse across the objects you want to select. or 1. Left-click the first object you want to select. 2. Press the **Ctrl** key and keep it pressed. 3. Left-click the next object you want to select. 4. Repeat these steps until the desired objects are selected.	1. Move the cursor to the first object to be selected, see also Navigating with the keyboard, page 655. 2. Press the **spacebar**. 3. Move the cursor to the next object to be selected 4. Press the **Ctrl**+**spacebar** key combination. 5. Repeat these steps until the desired objects are selected.

Selecting by rows

Selecting by rows:

Using the mouse	Using the keyboard
Selecting the contents of a row: 1. Left-click the number of the row you want to select in the vertical ruler. Selecting the contents of several rows: 1. Left-click the number of the first row you want to select in the vertical ruler. 2. Press the **Shift** key and keep it pressed. 3. Left-click the number of the last row you want to select in the vertical ruler. **Note:** Pressing the **Shift** key makes it possible to combine this procedure with the procedure for selecting columns.	-

Selecting by columns

Selecting by columns:

Using the mouse	Using the keyboard
Selecting the contents of a column: 1. Left-click the number of the column you want to select in the vertical ruler. Selecting the contents of several columns: 1. Left-click the number of the first column you want to select in the vertical ruler. 2. Press the **Shift** key and keep it pressed. 3. Left-click the number of the last column you want to select in the vertical ruler. **Note:** Pressing the **Shift** key makes it possible to combine this procedure with the procedure for selecting rows.	-

Selecting all (entire contents of the section)

Selecting all (entire contents of the section):

Using the mouse	Using the keyboard
Use the **Edit** > **Select All** menu command.	Press the **Ctrl**+**A** key combination.

Deselecting objects

Deselecting objects:

Using the mouse	Using the keyboard
Left-click an empty space in the section.	1. Move the cursor to an empty space in the section, see also Navigating with the keyboard, page 895. 2. Press the **spacebar**.

Deleting, cutting, copying, pasting, duplicating, and moving objects

Deleting objects

Deleting objects:

Using the mouse	Using the keyboard
1. Select, page 656 the object to be deleted. 2. Use the **Edit** > **Delete** menu command.	1. Select, page 656 the object to be deleted. 2. Press the **Delete** key.
Result: The selected object is deleted. The actual parameters (and the links for FFBs) are deleted together with the object, even if they are not selected explicitly. Please also pay careful attention to the Notes, page 697 on deleting from FFBs **using an execute after** reference.	

Cutting objects

Cutting objects:

Using the mouse	Using the keyboard
1. Select, page 656 the object to be cut. 2. Use the **Edit** > **Cut** menu command. or Use the menu command **Cut** in the shortcut menu (right mouse button). or Click the  symbol.	1. Select, page 656 the object to be cut. 2. Press the **Ctrl**+**X** key combination.
Result: The selected object is cut from the section and copied to the clipboard. This is also the case for the actual parameters. Links are also cut. But they are only copied to the clipboard if they (and their partner objects) are selected explicitly. Please also pay careful attention to the Notes, page 697 on deleting from FFBs **using an execute after** reference.	

The cut object can be inserted, page 660 in any other position (also in another FBD section).

Copying objects to the clipboard

Copying objects to the clipboard:

Using the mouse	Using the keyboard
1. Select, page 656 the object to be copied. 2. Use the **Edit > Copy** menu command. or Use the **Copy** menu command from the shortcut menu (right-click). or Click the  symbol.	1. Select, page 656 the object to be copied. 2. Press the **Ctrl+C** key combination.
Result: The selected object is copied to the clipboard. This is also the case for the actual parameters. Links are only copied to the clipboard if they (and their partner objects) are selected explicitly. Please also pay careful attention to the Notes, page 697 on copying from FFBs **using an execute after** reference.	

The copied object can be inserted, page 660 in any other position (also in another FBD section).

If the copied object is an EFB or a DFB instance, it can also be duplicated, page 661 in any other available position (also in another LD or FBD section).

Pasting objects from the clipboard

Pasting objects from the clipboard:

Using the mouse	Using the keyboard
1. Use the menu command **Edit > Insert**. or Use the menu command **Paste** in the shortcut menu (right mouse button). or Click the  symbol. 2. Left-click the target position.	1. Move the cursor to the target position using the arrow keys. 2. Press the **Ctrl+V** key combination. 3. Press **Enter**.

Note: This presents the behavior of the **Copy**, **Cut**, **Paste** on a function block instance. It only applies to graphical languages as **FBD** and **LD**.

- Using the **Paste** function after a **Copy** of an object:

  As a result of a **Copy**, a new function block instance (**FBI**) is used. The **Paste** function creates a new FBI whenever it is repeated. Accordingly, the FBI is incremented.

- Using the **Paste** function after a **Cut** of an object:

  As a result of a **Cut**, the same instance of the function is used. The **Paste** function uses the same FBI whenever it is repeated. Accordingly, the FBI is identical.

Note: Please also pay careful attention to the Notes, page 697 on copying from FFBs **using an execute after** reference.

NOTE: Using same instance of a function block is not recommended because it can lead to incorrect execution of the function blocks. A function block instance uses contextual variables that are owned by each instance usage and cannot be applied to any other. For instance, each current value is specific to one timer instance and one timer usage (Current value of a timer named TON_1 cannot be shared with another timer named TON_2).

Duplicating an EFB or a DFB instance from the clipboard

Duplicating an EF instance is not possible. If the clipboard contains several objects with different types (EF, EFB and DFB), the EF is copied while the EFB and DFB instances are duplicated.

Duplicating an EFB or a DFB instance from the clipboard:

Using the mouse	Using the keyboard
1. Use the menu command **Duplicate** in the shortcut menu (right mouse button). 2. Move the mouse to the desired target position. **Result:** A dotted frame appears and the mouse pointer changes its symbol to  for a valid position. 3. Left-click the target position.	1. Move the cursor to the target position using the arrow keys. 2. Press the **Ctrl+W** key combination. **Result:** The target position appears as a dotted frame with the symbol  in its center for a valid position. 3. Press **Enter**.

The duplicate operation is also possible across sections between different opened LD or FBD sections in the same or another Control Expert instance.

NOTE: If the EFB or DFB instances to be duplicated do not already exist in the other Control Expert instance, these instances are copied.

Moving objects

Moving objects:

Using the mouse	Using the keyboard
1. Select, page 656 the object to be moved. 2. Position the mouse pointer on the selected object. (on one of the selected objects if several are selected). **Result:** The mouse pointer changes its symbol to  . 3. Click the left mouse button and keep it pressed. **Result:** The mouse pointer changes its symbol to  . 4. Drag the object to the new position and release the mouse button.	1. Position, page 656 the cursor on the object to be moved. 2. Press the **spacebar**. 3. Press the **Shift** key and keep it pressed. 4. Move the object to the target position using the arrow keys.
Result: The selected object is moved from its original position to the target position. The object's actual parameters are moved together with the object. FFBs still have their links and they are adjusted to the new position of the FFB. **Note:** The move operation is also possible across sections between different opened FBD sections. **Note:** When objects are moved beyond the visible area of the editor window, automatic scrolling of the window will not begin until you reach the edge of the window with the mouse pointer, not as soon as the objects are moved beyond the edge.	

Copying objects via drag & drop

Copying objects via drag & drop:

Using the mouse	Using the keyboard
1. Select, page 656 the object to be copied. 2. Position the mouse pointer on the selected object. **Result:** The mouse pointer changes its symbol to  . 3. Click the left mouse button and keep it pressed. 4. Press the **Ctrl** key and keep it pressed. **Result:** The mouse pointer changes its symbol to  . 5. Drag the object to the new position. **Note:** This is also possible across sections between different opened FBD sections. 6. Release the mouse button. **Result:** A copy of the selected object is pasted at the target position. The object's actual parameters (variable/address) are copied together with the object. A new instance is automatically created for FFBs. Links are only copied if they (and their partner objects) are selected explicitly. Please also pay careful attention to the Notes, page 697 on copying from FFBs **using an execute after** reference.	-

Copying, pasting and adjusting pin variables

Copying variables to the clipboard

Copying variables to the clipboard:

Using the mouse	Using the keyboard
1. Select, page 656 the pin variable to be copied. 2. Use the **Edit > Copy** menu command. or Use the **Copy** menu command from the shortcut menu (right-click). or Click the  symbol.	1. Select, page 656 the pin variable to be copied. 2. Press the **Ctrl+C** key combination.
Result: The selected variable is copied to the clipboard.	

Pasting variables from the clipboard

Pasting variables from the clipboard:

Using the mouse	Using the keyboard
1. Use the menu command **Edit > Paste variable**. or Use the menu command **Paste variable** in the shortcut menu (right mouse button). or Click the  symbol. 2. Left-click the target position.	1. Press the **Ctrl**+**Shift**+**V** key combination. 2. Move the cursor to the target position using the arrow keys. 3. Press **Enter**.

Adjusting variables font size

You can increase or decrease the size of the font for the FFB pins (inputs and outputs) using the following buttons on the toolbar  .

NOTE: The text inside the blocks and the instance name cannot be changed.

Inserting Rows and Columns

Rules for Inserting Rows

The following rules apply when inserting rows/columns:

- The maximum size of 86,400 grids cannot be exceeded.
- The maximum number of rows is 1440.
- The maximum number of columns is 360.
- The size of one inserted row or column is like the ruler size (that is 10 grids).
- The first row/column selected must not intersect with an object, that has an origin before this row/column.
- The same number of rows/columns are inserted as selected in the vertical/horizontal ruler.
- The new rows/columns are inserted above/left of the first selected row/column (i.e. all objects of the selected rows/columns are moved down/right).

- When rows/columns are inserted within existing links, these links are kept and are adjusted to the new position of the objects.

NOTE: The section area (number of rows x number of columns) cannot exceed the maximum size of 86,400 grids, e.g. 240 x 360 or 1,440 x 60. The number of rows and columns will impact one another, e.g. the number of rows is determined by the internal calculation 86,400 / number of columns (limited to the lower tenth).

Inserting Rows

Step	Action
1	Select, page 656 the desired number of rows in the vertical ruler (click the row numbers).
2	Use the **Insert rows** menu command from the shortcut menu of the vertical ruler or the **Edit > Insert rows** command. **Result:** The number of selected rows is inserted above the first selected row.

Inserting Columns

Step	Action
1	Select, page 656 the desired number of columns in the horizontal ruler (click the column numbers).
2	Use the **Insert columns** menu command from the shortcut menu of the horizontal ruler or the **Edit > Insert columns** command. **Result:** The number of selected columns is inserted in front of the first selected column.

Deleting Rows and Columns

Rules for Deleting Rows

The following rules apply when deleting rows/columns:

- The minimum number of rows (240) respective columns (60) cannot be fallen short.
- The size of one deleted row or column is like the ruler size (that is 10 grids).
- The same number of rows/columns are deleted as selected in the vertical/horizontal ruler.
- When rows/columns are deleted within existing links, these links are kept and are adjusted to the new position of the objects.

- The objects of the selected rows/columns are deleted and the objects of the following rows/columns are moved up/left by the respective number of rows/columns.

Deleting Rows

Deleting rows:

Step	Action
1	Select, page 656 the desired number of rows in the vertical ruler (click the row numbers).
2	Use the **Delete rows** menu command from the shortcut menu of the vertical ruler or the **Edit > Delete rows** command. **Result:** The selected number of rows is deleted.

Deleting Columns

Deleting columns:

Step	Action
1	Select, page 656 the desired number of columns in the horizontal ruler (click the columns numbers).
2	Use the **Delete Columns** menu command from the shortcut menu of the horizontal ruler or the **Edit > Delete Columns** command. **Result:** The selected number of columns is deleted.

Displaying the properties

Displaying the object properties

You have the following options to display the object properties dialog box:

- Double-click the object.
- Select, page 656 the object and use the **Edit > Properties... dialog box.**.
- Select, page 656 the object and use the **Properties...command.**from the shortcut menu.
- Select, page 656 the object and then press the **Alt**+**Enter** key combination.

When the properties dialog box is called without an object being selected, the section properties dialog box , page 505 is opened.

If several objects are selected, the commands do not work.

View data properties

You have the following options to display the section properties dialog box, page 1202:

- using the data editor, page 348

1. Selecting one or several lines in a list editor
2. Using the shortcut menu select the **Properties**command.

- via the FBD section

1. Selecting one or several element(s) in the FBD section.
2. From the context menu select the command **Data properties** or press **Ctrl** + **Enter**.

Refining DFBs and subroutines

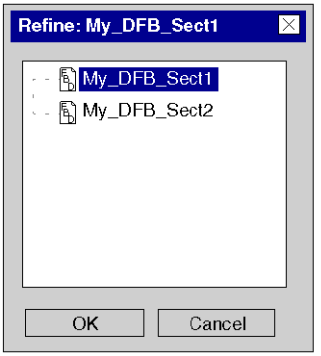
Introduction

This function makes it possible to "look into" a DFB or subroutine. This makes it possible to see the internal state of the DFB/subroutine when animation is used, or to make changes to the logic very quickly.

NOTE: The only limitation occurs when the displayed variable is in an ARRAY with at least 2 dimensions and when the index is not a constant. In this case an error message is displayed and the variable value cannot be displayed.

Refining DFBs

Process the following steps to refine DFBs:

Step	Action
1	Select the DFB.
2	You have the following options to refine DFBs: • Use the **Services > Refine** menu command. • Use the **Refine** menu command from the shortcut menu. or • Press the **Ctrl**+**Q** key combination. **Result:** If the DFB contains only one section, this will be opened automatically. If the DFB contains several sections, a selection dialog box is opened. Example: 
3	Select the desired DFB section. **Result:** The DFB section is opened. **Note:** With nested DFBs, several **refine** processes are possible, but do not open simultaneously the refine editors or it could lead to an incorrect data display.

Refining subroutines

Process the following steps to refine subroutines:

Step	Action
1	Select the subroutine block, page 709.
2	You have the following options to refine subroutines: • Use the **Services > Refine** menu command. • Use the **Refine** menu command from the shortcut menu. or • Press the **Ctrl**+**Q** key combination. **Result:** The subroutine is opened. **Note:** With nested subroutines several **refine** processes are possible.

Editing FFBs

Overview

This section describes the various possibilities to edit elementary functions, elementary function blocks, derived function blocks and procedures in the FBD programming language.

General information about calling an FFB

Introduction

FFB is the generic term for:

- Elementary Function (EF)
- Elementary Function Block (EFB)
- Derived Function Block (DFB)
- Procedure

  (Procedures are an expansion to IEC 61131-3 and must be enabled explicitly in the **Tools > Project Settings** dialog box, on the **Languages > Common** tab by selecting the **Allow procedures** checkbox).

FFBs can be placed in any free space.

If an FFB is placed in a space that is already occupied by an object, an error message is returned.

For more information, refer to chapter *Elementary Functions, Elementary Function Blocks, Derived Function Blocks and Procedures (FFBs)* (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).

Declaring block instances

Function blocks (elementary and derived) have internal states. If the inputs have the same values, the value on the output can have another value during the individual executions. For example, with a counter the value on the output is incremented.

The instance name serves as a unique identification for the function block in a project.

Instances can be declared:

- By selecting a function block type and its placement in a section.
- By the declaration of the block instance in the FFB input assistant, page 1199 its placement in the section.
- By the declaration of the block instance in the Data editor, page 380.

an FFB call.

FFBs can be called in the following ways:

- Via the FFB Input Assistant..., page 671 menu command
  - Use the **Edit > FFB Input Assistant...** menu command.
  - Use the menu command **FFB Input Assistant...** from the shortcut menu.

    or
  - Press the **Ctrl+I** key combination.
- Via the data selection, page 674
  - Use the **Edit > Data Selection...**.
  - Use the menu command **Data Selection...** from the shortcut menu.
  - Press the **Ctrl+D** key combination.

    or
  - Select the  symbol.
- Via the **Types library browser** per drag & drop
  - Use the **Tools > Types Library Manager** menu command.

    or
  - Press the **Alt+3** key combination.

NOTE: You can customize the FBD toolbar to call directly a maximum of 20 EF, EFB or DFB via the FBD Favorite Toolbar, page 743.

Calling an FFB via the FFB Input Assistant.

Activating the FFB input assistant

You have the following options to activate the FFB input assistant:

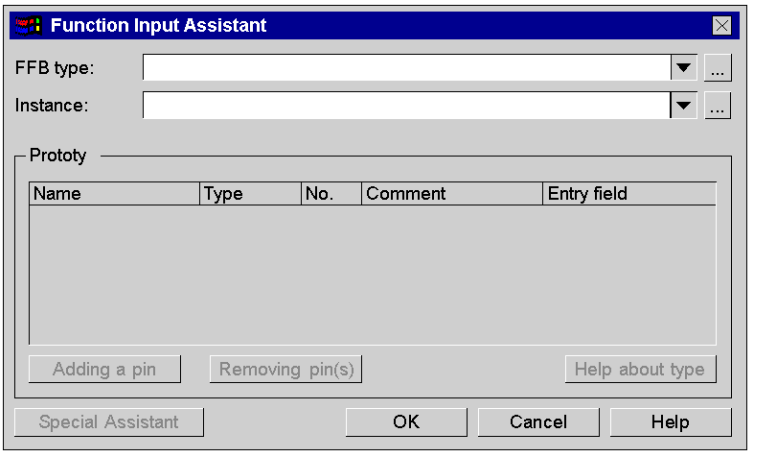
- Use the **Edit > FFB Input Assistant... menu command.** (no object may be selected).
- Use the **menu command** from the shortcut menu (no object may be selected).

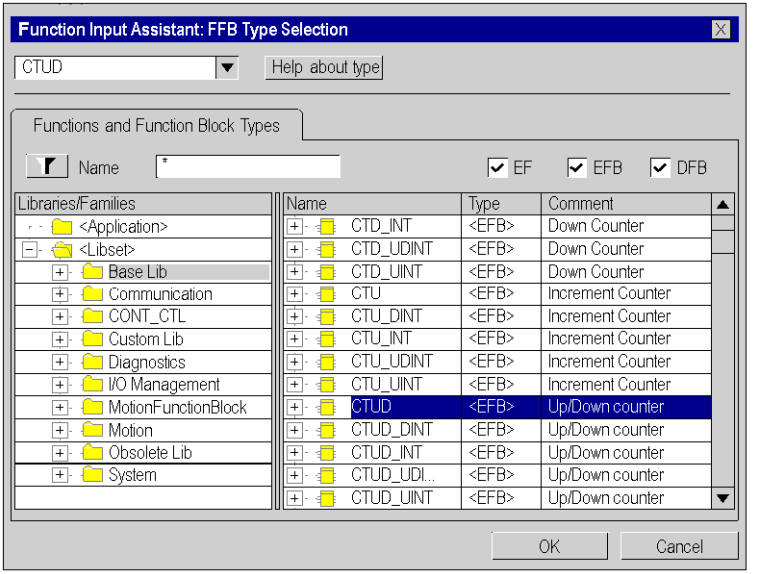
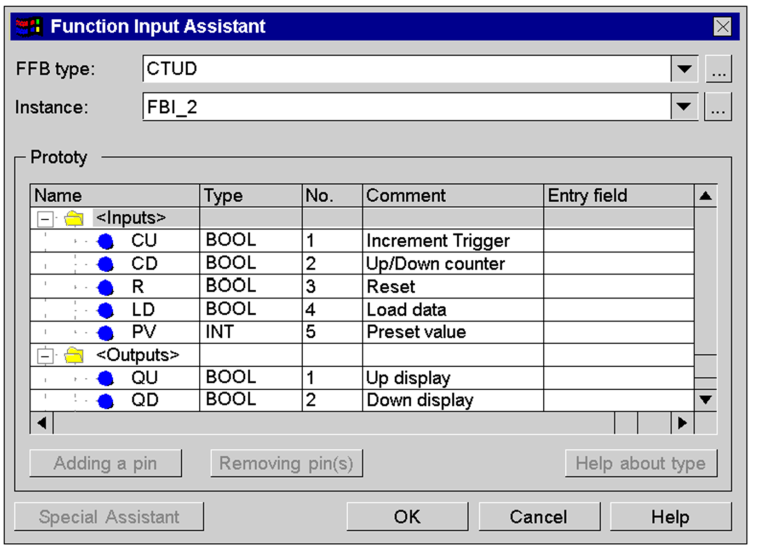
  or
- Press the key combination **Ctrl+I** (no object may be selected).

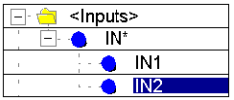
The active FFB placement mode is indicated by the  cursor symbol.

Selecting an FFB type via the FFB input assistant

Process the following steps to select an FFB type via the FFB input assistant:

Step	Action
1	From the **FFB input assistant....**from the shortcut menu (right-click) for the FFB input assistant, page 671section. The FFB input assistant, page 1199 is opened. 
2	You have the following options to enter the FFB type with the:**FFB Type** text box: • You can enter the FFB type in the textbox: In this case, continue with step 6 of the procedure. • You can select the FFB type from the list of recently used names using the ▼ symbol. In this case, continue with step 6 of the procedure. • Using the button **...** open the FFB selection dialog, page 1190.
3	Select the library from the **Libraries/Families** column which contains the desired function block. Tip: If you do not know where to find the FFB you are looking for select the **Libset**, entry to see a list of all the FFBs available independently of their libraries.
4	Select the desired function block from the **Name** column.

Step	Action
	Example: 
5	Confirm the selection with **OK**. **Result:** The dialog box is closed and FFB input assistant is activated. Example: 

Step	Action
6	You have the following options to enter the block instance for function blocks in the **instance** text box: • You can accept the name which is suggested. • You can change the name which is suggested.
7	The number of inputs, page 692can be increased with some elementary functions. Expandable pins can be recognized from the following display:  To add further pins select the last pin in the structure and press **Add pin**.
8	To assign the current parameter double-click the **Entry field** cell of the first formal parameter and enter the parameter to be used. You have the following options to enter the variables/addresses: • You can enter the variable name and confirm using **Enter**. • You can select the variable/address from the list of recently used variables/addresses using the ▼ symbol. or • Using the button **...**to open a variable selection dialog box, page 1181. Assign an actual parameter to all formal parameters of the function block in this way.
9	Confirm the entries using the **OK** button. **Result:** The FFB input assistant is closed and the FFB input mode is activated. The active FFB placement mode is indicated by the cursor symbol.
10	To place the FFB , and click the desired location in the FBD section. or Use the **arrow keys** to move the cursor to the target position, and press **Enter**. **Result:** The FFB is inserted and a syntax and semantics check, page 752 is performed.

Calling an FFB via the data selection

Set data selection to active

You have the following options to activate the data exchange:

- Use the **Edit > Data Selection... menu command**.

- Use the menu command**Data selection...**from the shortcut menu (right-click) for the section.
- Press the **Ctrl**+**D** key combination.

  or

- Select the  symbol.

  NOTE: When executing the commands no object can be selected in the section.

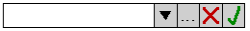
The active FFB placement mode is indicated by the  cursor symbol.

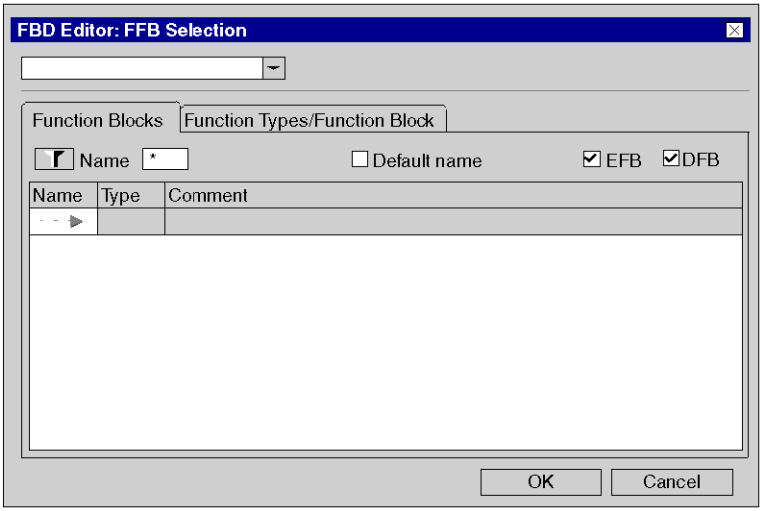
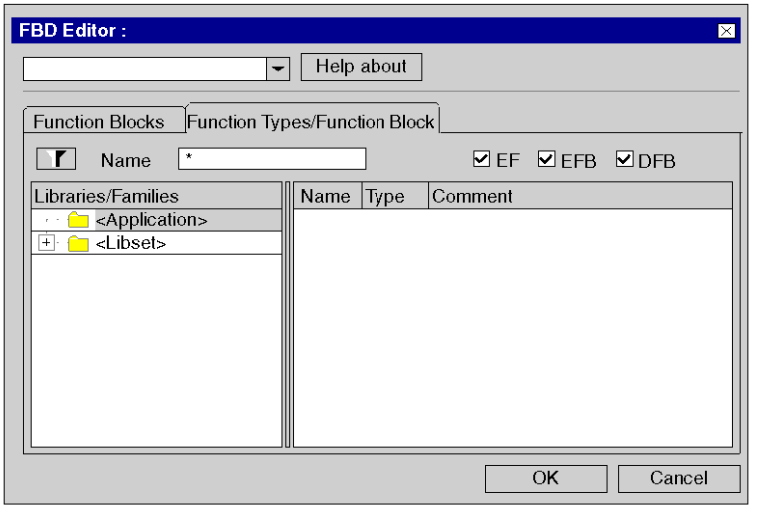
Note

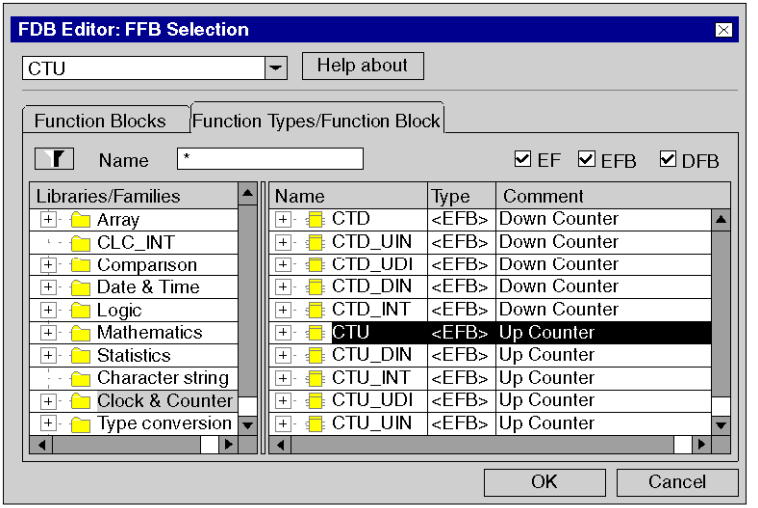
In the **Tools > Options** dialog box on the **Data and Languages** tab, if the **Automatically assign a variable to a new graphical object** checkbox is selected, then the corresponding input assistant is automatically opened when an FFB is placed. The procedures explained here are used when the checkbox is **cleared**.

Selecting an FFB type via the Data Selection

Process the following steps to select an FFB type via the Data selection:

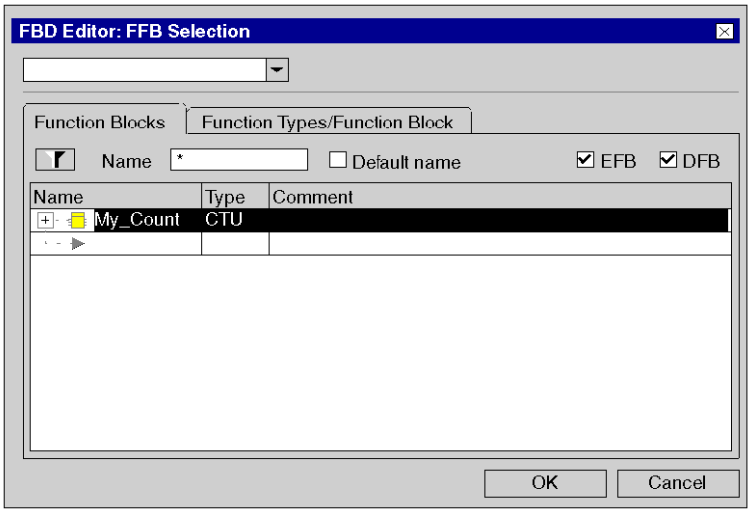
Step	Action
1	From the **Data selection....**from the shortcut menu (right-click) for the Data selection, page 674section. **Result:** A data selection box, page 1850 is opened. 
2	You have the following options to enter the FFB type: • You can now enter the FFB type and confirm using **Enter**or the ✓button. In this case, continue with step 7 of the procedure. • You can select the name from the list of recently used names using the FFB type▼ symbol and confirm using **Enter**or the button ✓. In this case, continue with step 7 of the procedure. • You can delete your entry by pressing **Esc**or clicking the ✗ button. • Using the button **...you can open** the dialog box for FFB selection, page 1186.

Step	Action
	Example of a FFB type selection, page 1188 dialog box: 
3	Select the **Function Types, Function Block Types** tab. **Result:** The tab used for selecting an FFB type is opened. 
4	Select the library and family from the **Libraries/Families** column which contains the desired function block.

Step	Action
	Tip: If you do not know where to find the FFB you are looking for select the **Libset**, entry to see a list of all the FFBs available independently of their libraries.
5	Select the desired function block from the **Name** column. Example: 
6	Confirm the selection with **OK**. **Result:** The dialog box is closed and FFB placement mode is activated.
7	To place the FFB , and click the desired location in the FBD section. or Use the **arrow keys** to move the cursor to the target position, and press **Enter**. **Result:** For the selected FFB type, an instance name is automatically generated (only valid for function blocks), the FFB is inserted and placement mode remains active so that further FFBs can be inserted. To end insert mode press **Esc**. **Note:** The automatically generated instance names can be changed to improve clarity, see also Managing of instances of data belonging to the family of function blocks (EF), page 380.

Selecting a block instance

Process the following steps to select a block instance that has already been declared:

Step	Action
1	Activate FFB placement mode. **Result:** A data selection box, page 1850 is opened. 
2	Press the button **...**, to open the FFB selection dialog, page 1186.
3	Select the **Function Blocks** tab (if it is not already active). **Result:** The tab used for selecting a block instance is opened and the desired selection of block instances, page 1187 is displayed. 
4	Select the desired block instance from the **Name** column.
5	Confirm the selection with **OK**. **Result:** The dialog box is closed and FFB placement mode is activated.
6	Click the target position in the FBD section. or Use the **arrow keys** to move the cursor to the target position, and press **Enter**. **Result:** The selected block instance is inserted and the placement mode remains active so that more block instances can be inserted. To end insert mode press **Esc**.

Assign actual parameters

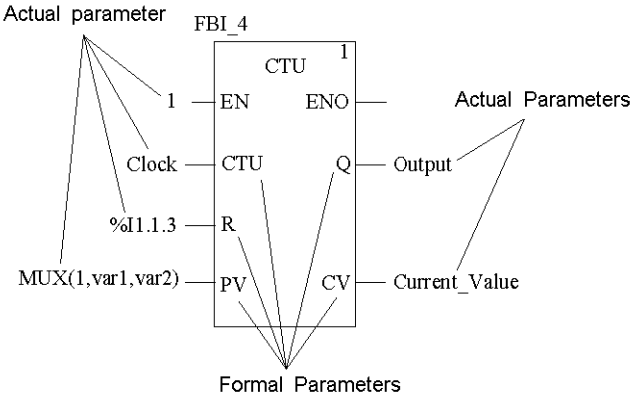
Introduction

Inputs and outputs are required to transfer values to or from an FFB. These are called formal parameters.

Objects are linked to formal parameters; these objects contain the current process states. They are called actual parameters.

The data type of the actual parameter must match the data type of the input/output (formal parameter). The only exceptions are generic inputs/outputs whose data type is determined by the actual parameter. If all actual parameters consist of literals, a suitable data type is selected for the function block.

Formal and Actual Parameters:



Permitted actual parameters are:

- Variables
  - Enter already declared variables, page 680
  - Enter undeclared variables, page 682
- Formal parameters of other function blocks, page 684
- Literals, page 686
- ST Expression, page 687

  ST expressions as formal parameters on FBB inputs are an extension of IEC 61131-3 and must be explicitly enabled via the **Tools > Project Settings** dialog box, in the **Language extensions** tab by selecting the **Usage of ST expressions** checkbox.
- Links, page 723 to other FBD objects

NOTE: Please take note of the section Information about Programming (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.

You can make actual parameter assignments for

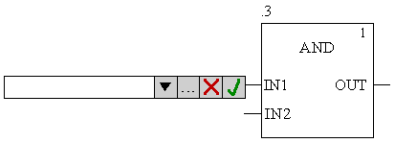
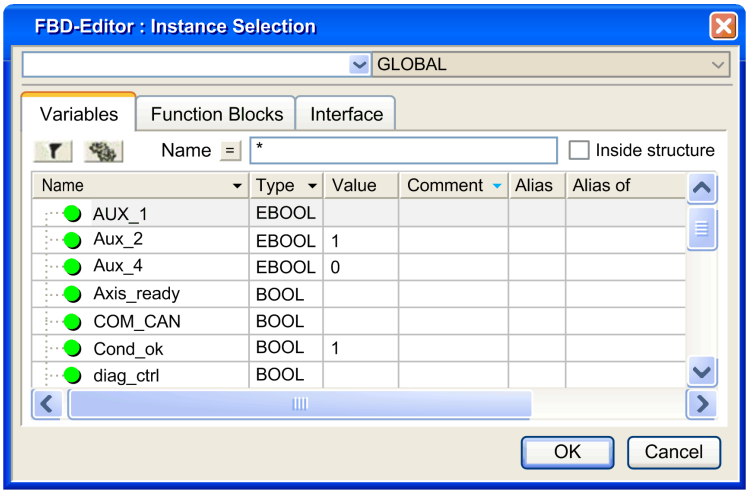
- each pin individually

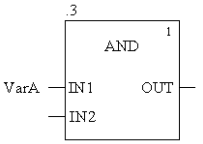
  or

- with the function input assistant, page 689 for all pins

Assign declared variables to a pin

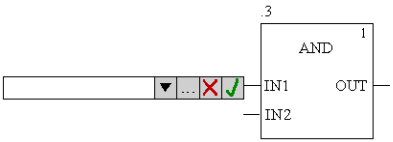
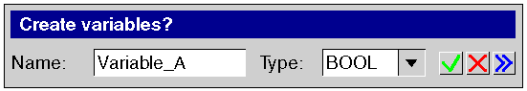
Process the following steps to assign undeclared variables to a pin. (See also Creation of EDT instances, page 390.)

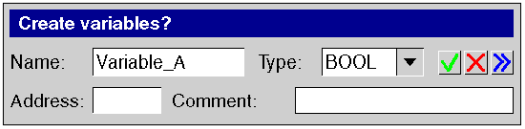
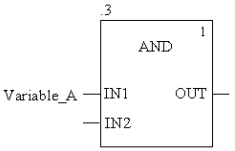
Step	Action
1	Open a data selection box for the desired FFB pin by: • double-clicking the pin. • selecting the pin and clicking the **Edit > Data Selection...** menu command. • selecting the pin and clicking the **Data Selection** menu command on the shortcut menu. • selecting the pin and pressing the **Ctrl+D** key combination. or • selecting the pin and clicking the  symbol. **Result:** The data selection box, page 1850 opens. 
2	You have the following options to enter variables: • You can now enter the variable name and confirm using **Enter**or the ✓button. • You can select the variable names from the list of recently used names using the ▼ symbol and confirm using **Enter**or the ✓button. • You can delete your entry by pressing **Esc**or clicking the ✕ button. • Using the button **...** open an **Instance selection** dialog box and confirm the selected variables there with **OK**. Example of a instance selection dialog box:  **Result:** The selected variable is inserted and a syntax and semantics check, page 654 is performed.

Step	Action
	Example: 

Assign undeclared variables to a pin

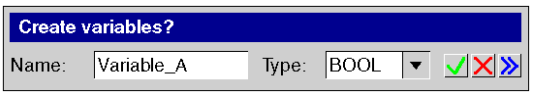
Process the following steps to assign undeclared variables to a pin:

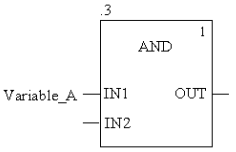
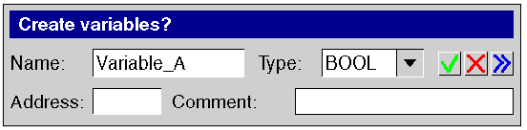
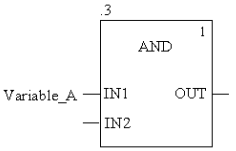
Step	Action
1	Open a data selection box for the desired FFB pin by: • double-clicking the pin. • selecting the pin and clicking the **Edit > Data Selection...** menu command. • selecting the pin and clicking the **Data Selection** menu command on the shortcut menu. • selecting the pin and pressing the **Ctrl+D** key combination. or • selecting the pin and clicking the  symbol. **Result:** A data selection box, page 1850 opens. 
2	Enter the variable name and confirm using **Enter**or the  symbol. **Result:** The variable declaration dialog box opens. 
3	Change the default data type, if desired. **Note:** If you wish to declare the variable later, close the dialog box using the  symbol. In this case, the variable name is enabled on the pin, but not declared.

Step	Action
4	If you do want to assign the variable an address and/or comment, use the symbol for obtaining an advanced dialog and enter the address and/or comment. 
5	Confirm with **Enter** or the symbol. **Result:** The variable is declared and enabled on the selected pin. 
6	Enter all actual parameters in this way.

Declaring variables on a pin

Process the following steps to declare variables on a pin:

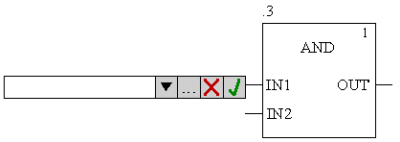
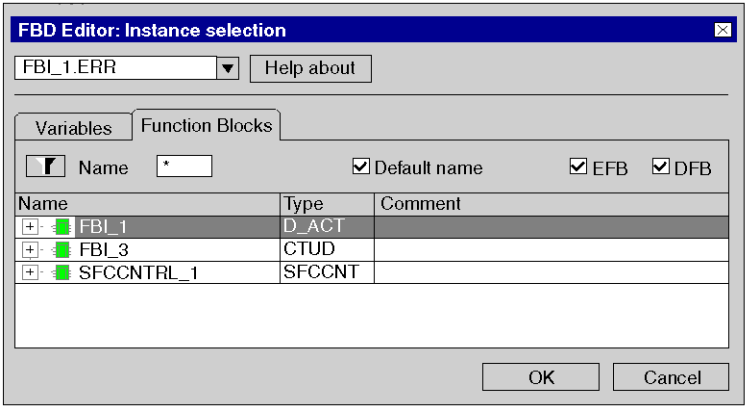
Step	Action
1	Open the variable declaration dialog box for the desired variable by. • Select an undeclared variable (red wavy line under the variable name) and the **Create Variable** command from the shortcut menu. • Select an undeclared variable (red wavy line under the variable name) and the **Shift+Enter** key combination. **Result:** The variable declaration dialog box opens. 
2	If you do not want to assign an address or comment, confirm using **Enter** or the symbol. **Result:** The variable is declared and the red wavy line under the variable name disappears.

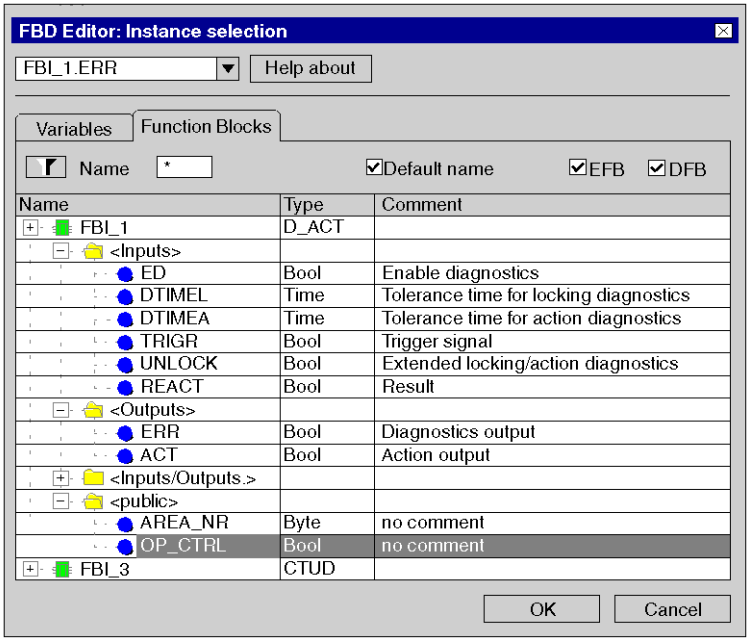
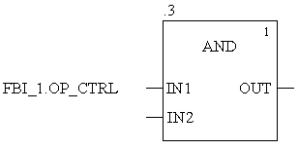
Step	Action
	
3	If you do want to assign the variable an address and/or comment, use the » symbol for obtaining an advanced dialog and enter the address and/or comment. 
4	Confirm with **Enter** or the ✓ symbol. **Result:** The variable is declared and the red wavy line under the variable name disappears. 

Assigning formal parameters to a pin

Process the following steps to assign formal parameters to a pin:

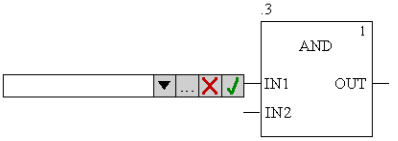
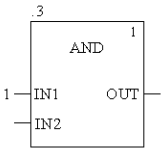
Step	Action
1	Select the desired FFB pin.
2	Open a data selection box by: • double-clicking the pin. • selecting the pin and clicking the **Edit > Data Selection** menu command. • selecting the pin and clicking the **Data Selection** menu command on the shortcut menu. • selecting the pin and pressing the **Ctrl+D** key combination. or • selecting the pin and clicking the symbol. **Result:**

Step	Action
	The data selection box, page 1850 opens. 
3	You have the following options to enter the formal parameter: • You can enter the formal parameter name and confirm using **Enter** or the ✓ button. • You can select the formal parameter name from the list of recently used names using the ▼ symbol and confirm using **Enter**or the ✓button. • You can delete your entry by pressing **Esc**or clicking the ✗ button. • Using the button **...** to open an **Instance selection** dialog box.
4	Select the **Function Block** tab. **Result:** The function block instances are displayed. 
5	Expand the display of the inputs, outputs and public variables by clicking their **+** symbols. **Result:**

Step	Action
	You can now see all available inputs, outputs and public variables of the function block. 
6	Select the desired formal parameter and confirm with **OK**.
7	**Result:** The formal parameter selected is inserted and a syntax and semantics check, page 654 is performed. Example: 

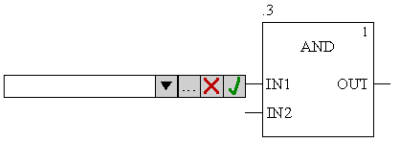
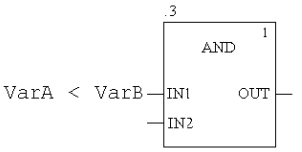
Assigning literals to a pin

Process the following steps to assign literals to a pin:

Step	Action
1	Select the desired FFB pin.
2	Open a data selection box by: • double-clicking the pin. • selecting the pin and clicking the **Edit > Data Selection** menu command. • selecting the pin and clicking the **Data Selection** menu command on the shortcut menu. • selecting the pin and pressing the **Ctrl+D** key combination. or • selecting the pin and clicking the  symbol. **Result:** The data selection box, page 1850 opens. 
3	Enter the literal (e.g. `0`, `1`, `TRUE`, `FALSE`, `1.0`, `3.5`, `t#2ms`) and confirm using the **Enter**button. **Tips:**When in the **Tools > Project Settings** dialog box, on the **Language extensions** tab, the **Allow leading digits** checkbox is cleared, the following simplifications for entering literals are possible: • When assigning `TIME` literals to a pin of the `TIME` data type, it suffices to enter the numerical value and the unit (such as `h`, `m`, `s`). The prefix (`t#`) is automatically added. • When assigning `REAL` literals to a pin of the `REAL` data type, for integers it suffices to enter the numerical value. The decimal marker (`t#`) is automatically added. **Result:** The literal is inserted and a syntax and semantics check, page 654 is performed. Example: 

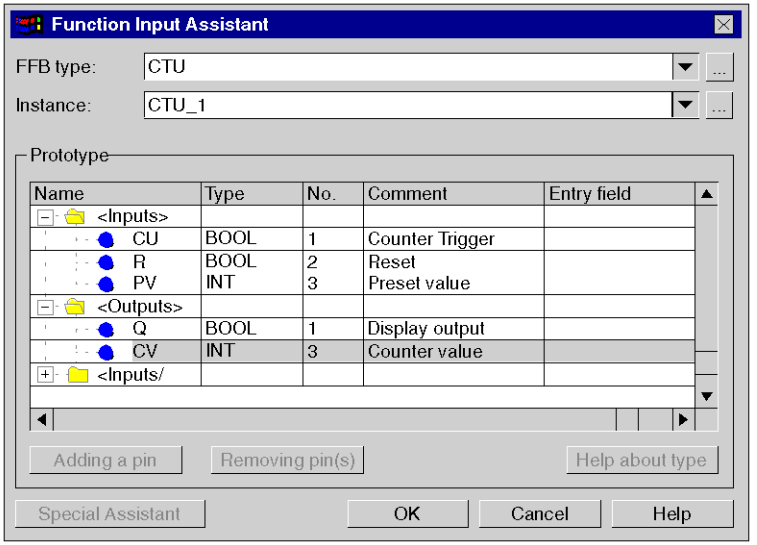
Assigning ST expressions to a pin

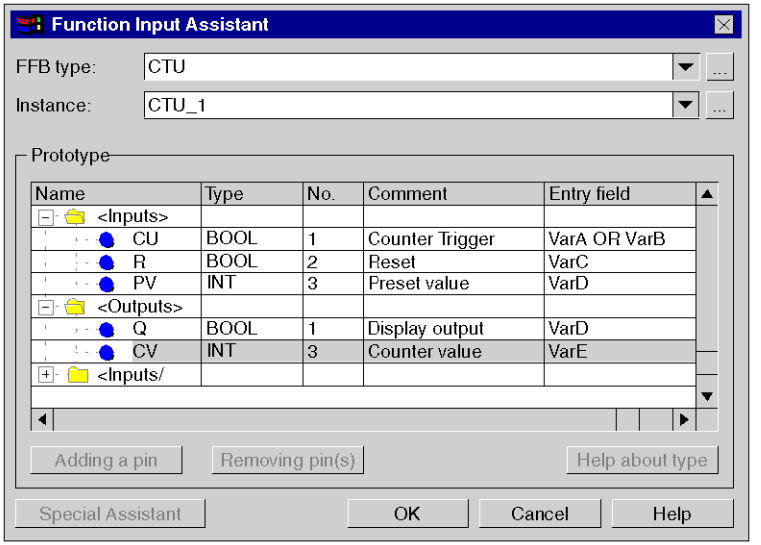
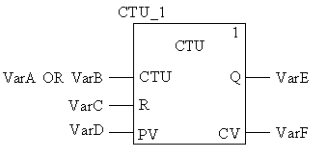
Process the following steps to assign ST expressions to a pin:

Step	Action
1	Make sure that the **Usage of ST expressions** checkbox is activated in the **Tools > Project Settings** dialog box.
2	Select the desired FFB pin.
3	Open a data selection box by: • double-clicking the pin. • selecting the pin and clicking the **Edit > Data Selection** menu command. • selecting the pin and clicking the **Data Selection** menu command on the shortcut menu. • selecting the pin and pressing the **Ctrl+D** key combination. or • selecting the pin and clicking the  symbol. **Result:** The data selection box, page 1850 opens. 
4	Enter the ST expression (e.g. `MUX(1,var1,var2)`, `VarA * VarB`, `VarA < VarB`) and confirm pressing **Enter**. **Result:** The ST expression selected is inserted and a syntax and semantics check, page 654 is performed. Example:  If the variables have already been declared (see also Creation of EDT instances, page 390), the procedure ends here. If the variables have not yet been declared, continue with step 5.
5	Declare all variables used, either: • using the data editor, page 390 or • using **Create variable** in the context menu (possible when only one variable of the ST statement is undeclared).

Actual parameter assignment using the function input assistant

Process the following steps to change actual parameter assignment with the function input assistant:

Step	Action
1	Select the desired FFB.
2	Open the function input assistant, page 1199: • The **Edit > FFB Input Assistant...** menu command. • The **FFB Input Assistant...** menu command from the shortcut menu. or • The **Ctrl+I** key combination. **Result:** The function input assistant will be opened. 

Step	Action
3	Double click the **Entry Field** cell of the first formal parameter and enter the actual parameter to be used. You have the following options to enter the variables/addresses: • You can enter the variable name and confirm using **Enter**. • You can select the variable/address from the list of recently used variables/addresses using the ▼ symbol. or • Using the buton **...** to open an **Instance selection** dialog box. Assign an actual parameter to all formal parameters of the function block in this way. Example: 
4	Confirm the entries using the **OK** button. **Result:** The actual parameter is inserted and a syntax and semantics check, page 654 is performed. Example: 

Using Public Variables

Introduction

In addition to inputs/outputs, some function blocks also provide public variables.

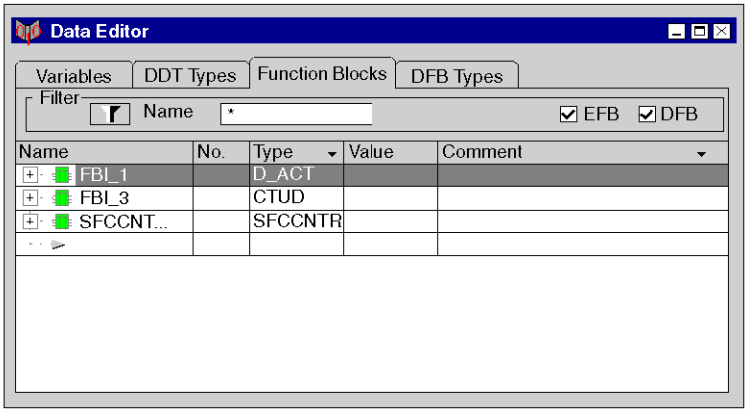
These variables transfer statistical values (values that are not influenced by the process) to the function block. They are used for setting parameters for the function block.

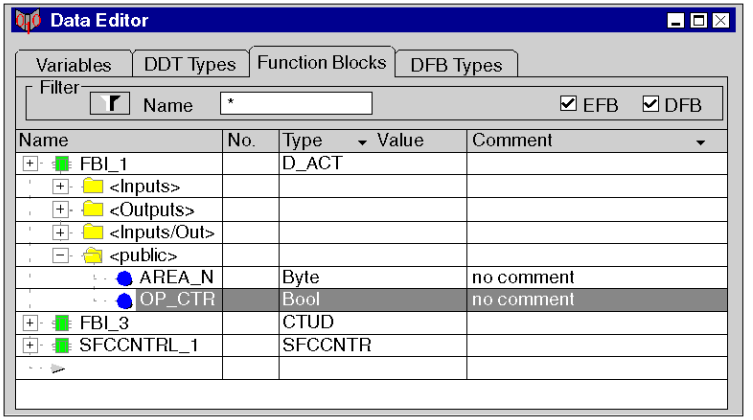
The assignment of values to public variables is made via their initial values.

Public variables are read via the instance name of the function block and the names of the public variables.

Assigning Values to Public Variables

Carry out the following steps to assign a value to a public variable:

Step	Action
1	Place a function block in the section, see also Calling an FFB via the data selection, page 674.
2	Open the data editor, see also Data Editor Access, page 349.
3	Select the **Function Block** tab. **Result:** The function block instances are displayed. 

Step	Action
4	Expand the display of the public variables by clicking the respective **+** symbols. **Result:** You can now see all available public variables of the function block. 
5	Enter the desired value for the public variable in the **Value** box.

Reading public variables

See Assigning formal parameters to a pin, page 684

Expanding Functions

Introduction

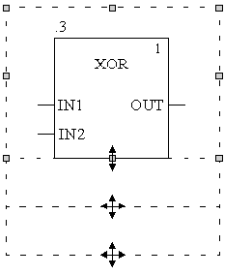
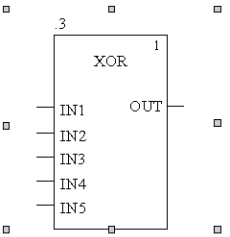
The number of inputs can be increased with some elementary functions.

You find out which functions can be expanded by referring to the descriptions for the individual functions.

NOTE: Only expand the function by the number of inputs actually required as non-occupied inputs are assigned a 0 as standard.

Expanding a function

Carry out the following steps to expanding a function:

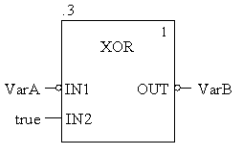
Step	Action
1	Select the function.
2	Position the mouse pointer on the bottom point to be expanded.
3	Press the left mouse button and drag the function to the required size. Example: 
4	Release the mouse key. **Result:** The desired number of inputs/outputs have been added. Example: 

Negating FFB Pins

Introduction

FFB pins (inputs and outputs) of the data type `BOOL` can be negated.

Negated pins are identified with a circle symbol:



NOTE: Negated pins must be connected with a graphical link or a variable. Open negated pins are not allowed.

Negating an FFB pin

FFB pins can be negated in the following ways:

- Select the pin to be negated and execute the menu command **Negate Pin** in the shortcut menu (right-click).

  or

- Launch the Inversion tool using
  - the menu command **New > Inversion Tool**,
  - the menu command **Inversion Tool** in the shortcut menu (right-click)

    or

  - the icon 

  and click on the pin to be negated.

EN and ENO Show and Hide

Introduction

With all FFBs, one EN input and one ENO output can be configured, see also EN and ENO (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.

EN and ENO show or hide

To show or hide EN and ENO, carry out the following steps:

Step	Action
1	Select the FFB.
2	Open the properties dialog box, page 666 for the FFB.
3	Check the **FFB Properties** check box **Show EN/ENO** to show EN/ENO, or uncheck it to hide EN/ENO. **Note:** If EN or ENO is occupied by an actual parameter or connected with a link, EN/ENO cannot be hidden.

Modifying the execution order

Introduction

The execution order is determined by the position of the FFBs in the section (executed from left to right and from top to bottom) and is displayed with execution numbers. If the FFBs are connected to a network using graphical links, the execution order is determined by the signal flow, see also section Execution order (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.

You can already influence the execution order when creating the program.

- The use of links in place of actual parameters (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual)
- The position of networks (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual)

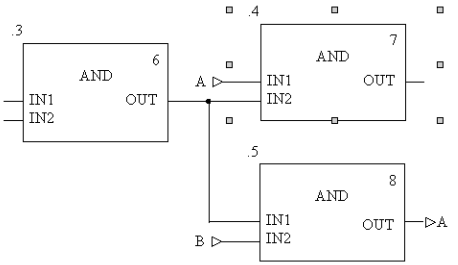
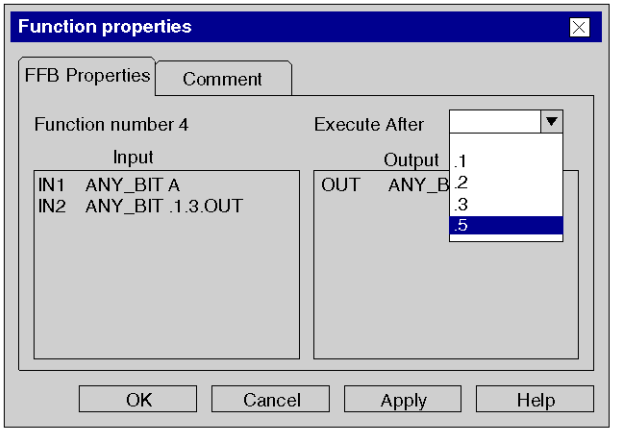
In some cases it might be necessary to explicitly modify the execution order.

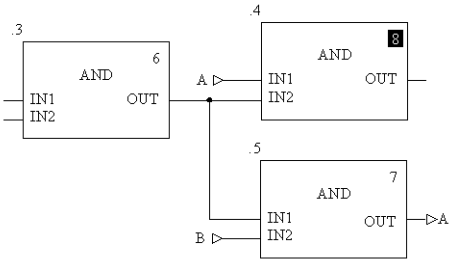
By using **Execute After** in an FFB properties dialog the execution order of two FFBs can be defined.

FFBs whose execution order has been explicitly modified receive the instance name / function number of the previously executed FFB as an additional entry in front of the execution number.

Modifying the execution order

In the example, the execution order of FFB 1.4 and FFB 1.5 has to be switched. Process the following steps to modify the execution order using an FFB properties dialog box.

Step	Action
1	Select the FFB to be executed second. Example: 
2	Open the properties dialog box, page 666 for the FFB.
3	In the **Execute After** list box, select the function number / instance name of the FFB to be executed before the selected FFB. Example:  **Result:**The execution sequence of both FFBs are exchanged. To indicate that the execution order was changed, the execution number is shown in a black field. Example:

Step	Action
	 **Note:**Only one reference of an instance is allowed, e.g. the instance ".6" may only be referenced once.

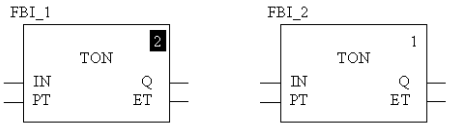
Notes

When using **Execute after** please note the following:

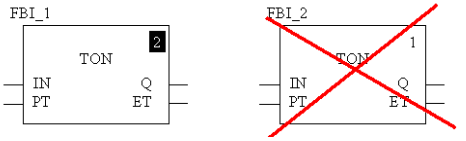
- If an FFB that is referenced by another FFB is deleted, the execution order attribute of the referencing FFB is reset.

  Example:

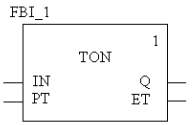
1. FBI_1 has an "Execute After" reference to FBI_2.



2. FBI_2 is deleted.



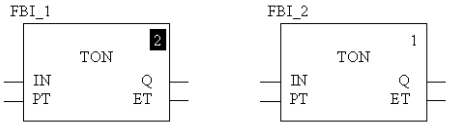
3. For FBI_1 the attribute for determining the execution sequence is deleted.



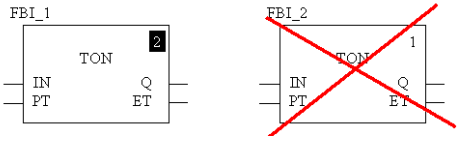
- If an FFB deletion operation carried out on a referenced FFB is undone using**Undo** it is reset to its former state.

  Example:

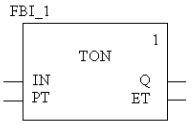
1. FBI_1 has an "Execute After" reference to FBI_2.



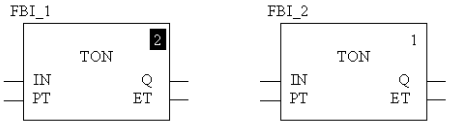
2. FBI_2 is deleted.



3. For FBI_1 the attribute for determining the execution sequence is deleted.



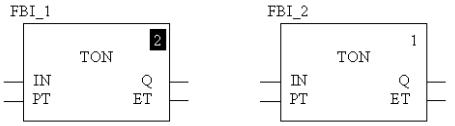
4. Execute from **Edit > Undo**.
5. FBI_2 is inserted again and FBI_1 and FBI_2 has an "Execute After" reference.



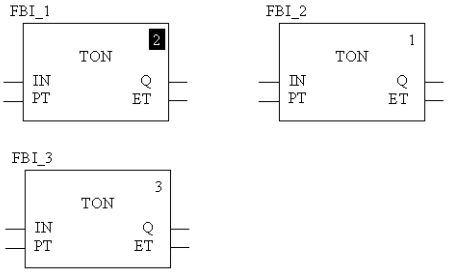
- If an FFB that has an "Execute After" reference pointing to another FFB is copied, the attribute which defines the execution order attribute of the FFB is reset during insertion.

  Example:

1. FBI_1 has an "Execute After" reference to FBI_2.



2. FBI_1 is copied and the copy (FBI_3) inserted in the section.
3. For FBI_3 the attribute for determining the execution sequence is deleted.



Replacing an FFB

Access

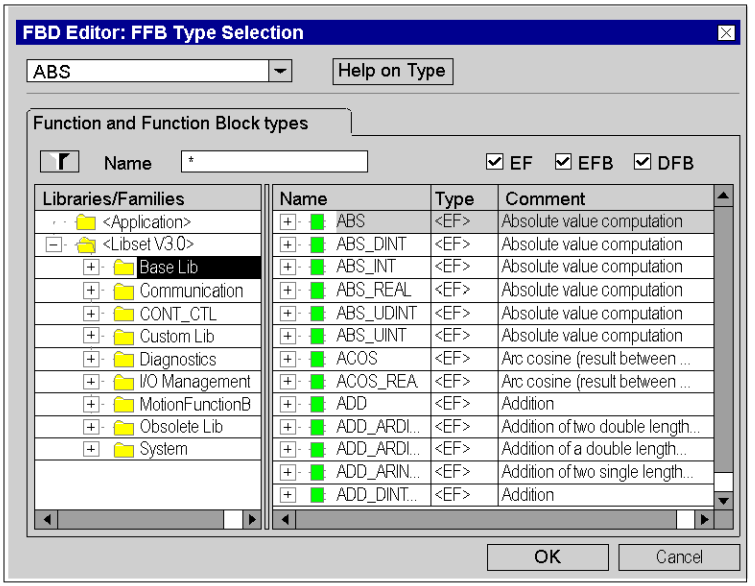
This is only available in FBD sections.

You have the following options to access the **Replace FFB** feature:

- Select the **Edit > Replace FFB** menu command.
- Select the **Replace FFB** command in the context menu (right-click on one or more FFBs in the section).

Replace FFB

Selecting the **Replace FFB** feature opens the **FBD-Editor: FFB Type Selection** dialog box.



For more information about this dialog box see FFB Selection Dialog box, page 1186.

Select the desired type and confirm with **OK**. The FFBs selected in the FBD section are replaced with the new type.

Links and variables connected will be retained if possible (if a pin on the same position of the new type exists).

Type Changing

There are different cases of type changing:

Type Changing	Description
EF -> EF	changes the type for the existing EF only
EF -> FB	creates a new instance of the FB type
FB -> EF	creates the new EF template, does not delete the old FB instance in the Data Editor (even if it is the only instance)
FB -> FB	creates the new FB instance, does not delete the old FB instance in the Data Editor (even if it is the only instance)

Undo/Redo

Replace FFB supports Undo/Redo, which means that the original state will be recreated on Undo (of course without deleting the new FFBs from Data Editor) and the state after replacement on Redo.

Online Mode

Replace FFB is supported in online mode.

Properties dialog box for elementary functions, procedures and sub-routine blocks.

Calling the properties dialog box

see Displaying the properties, page 666

Structure of the properties dialog box

The properties dialog box consists of two tabs:

- **FFB Properties**

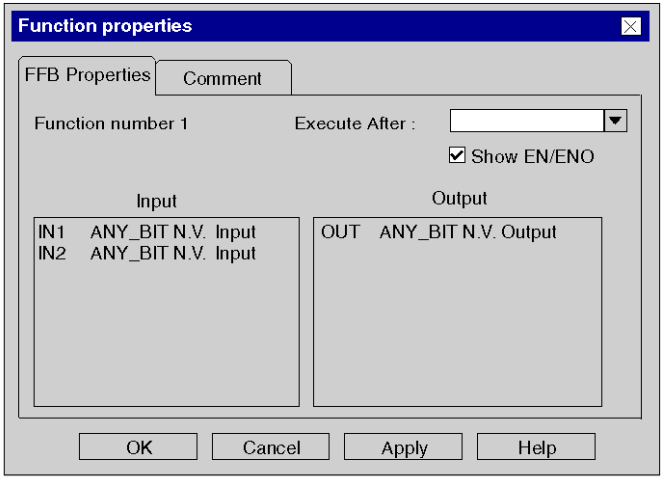
  General information about functions/procedures/sub-routine calls is displayed in this tab and you can change their order of execution.

- **Comment**

  You can enter a comment about the functions/procedures/sub-routine calls in this tab.

FFB Properties tab

Representation of the **FFB Properties** tab:

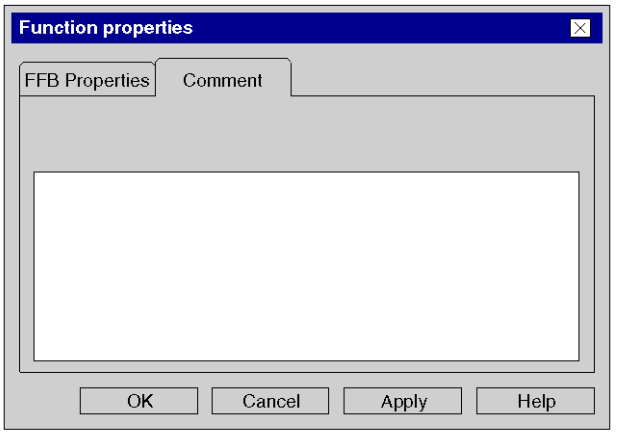


Elements of the **FFB Properties** tab:

Element	Description
Function number	Specifies the name of the function number and the current execution order. This name cannot be modified.
Execute After	You can define the FFB execution order in this list. Select the FFB from this list after which the current FFB should be executed, see also Modifying the execution order, page 695.
Show EN/ENO	When you activate this checkbox the pins EN/ENO are displayed. When you uncheck this checkbox the pins EN/ENO are no longer displayed. **Note:** If EN or ENO is occupied by an actual parameter or connected with a link, the pins EN/ENO cannot be hidden.
Input	Displays the formal parameters, the data types and the actual parameter of the inputs. If the input is not yet linked, the entry **N.L. appears**(not linked).
Output	Displays the formal parameters, the data types and the actual parameter of the outputs. If the output is not yet linked, the entry **N.L. appears**(not linked).
OK	Use this button to accept all entries and close the dialog box.
Apply	Use this button to accept all entries without closing the properties dialog box.

Comment tab

Representation of the**Comment**:



Elements of the **Comment** tab:

Element	Description
Text box	Enter a comment about the FFB. This comment is displayed in the tooltip when the cursor is placed over the function/ procedure.
OK	Use this button to accept all entries and close the dialog box.
Apply	Use this button to accept all entries without closing the properties dialog box.

Properties dialog box for elementary and derived function blocks (FBs)

Calling the properties dialog box

see Displaying the properties, page 666

Structure of the properties dialog box

The properties dialog box consists of two tabs:

- **FFB Properties**

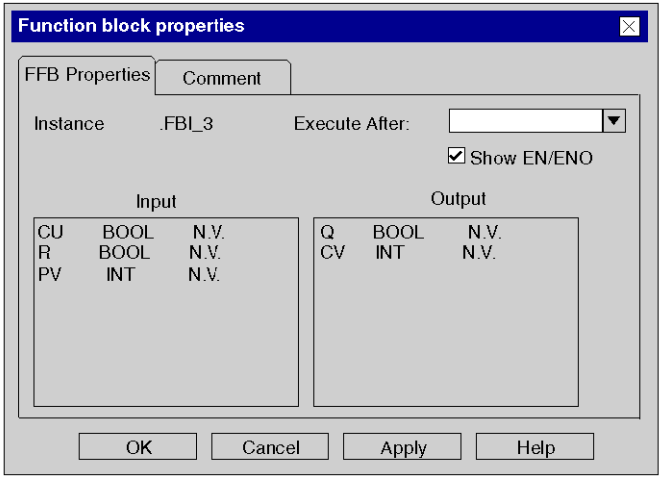
  General information about FBs is displayed in this tab and you can change their execution order.

- **Comment**

  A comment about the FB can be entered in this tab.

FFB Properties tab

Representation of the **FFB Properties** tab:



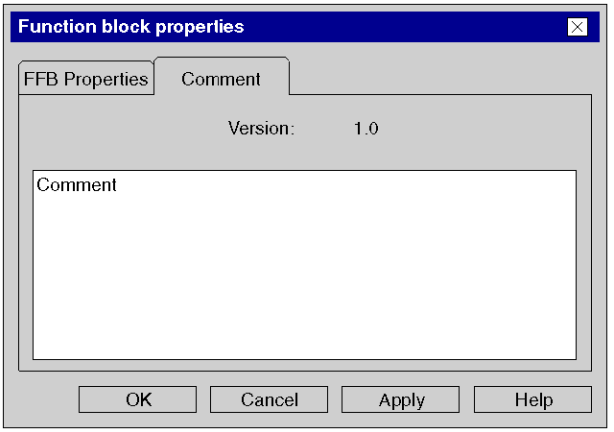
Elements of the **FFB Properties** tab:

Element	Description
Instance Name	Specifies the name of the block instance. You can change this name in the data editor, see also Managing of instances of data belonging to the family of function blocks (EF), page 380.
Execute After	You can define the FFB execution order in this list. Select the FFB from this list after which the current FFB should be executed, see also Modifying the execution order, page 695.
Show EN/ENO	When you activate this checkbox the pins EN/ENO are displayed. When you uncheck this checkbox the pins EN/ENO are no longer displayed. **Note:** If EN or ENO is occupied by an actual parameter or connected with a link, the pins EN/ENO cannot be hidden.
Input	Displays the formal parameters, the data types and the actual parameter of the FB inputs.

Element	Description
	If the input is not yet linked, the entry **N.L. appears**(not linked).
Output	Displays the formal parameters, the data types and the actual parameter of the FB outputs. If the output is not yet linked, the entry **N.L. appears**(not linked).
OK	Use this button to accept all entries and close the dialog box.
Apply	Use this button to accept all entries without closing the properties dialog box.

Comment tab

Representation of the tab **Comment**:



Elements of the **Comment** tab:

Element	Description
Version:	Specifies version of the FB code.
Text box	Enter a comment about the FB. This comment is displayed in the tooltip when the cursor is placed over the FB.
OK	Use this button to accept all entries and close the dialog box.
Apply	Use this button to accept all entries without closing the properties dialog box.

Returning from a Subroutine or DFB

Overview

This section describes how to return from a subroutine or DFB to the FBD programming language.

Returning from a Subroutine or DFB

Introduction

Every subroutine and every DFB (derived function block) is exited after being processed, i.e. you return to the main program called.

If the subroutine/DFB is left prematurely, the return to the main program can be forced by the return object.

If the status of the left link is 1, a return from the subroutine or DFB to the main program is performed.

Return objects can only be used in subroutines or DFBs. They cannot be used in main programs.

To generate a conditional return, a return object is linked to a Boolean FFB output.

Selecting the return object

You have the following options to select return objects:

- Use the **Edit > New > Return** menu command.

  or

- Choose the  symbol.

The active placement mode for return objects is indicated by the  cursor symbol.

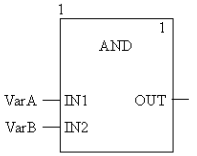
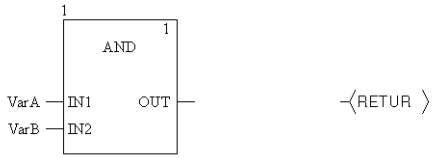
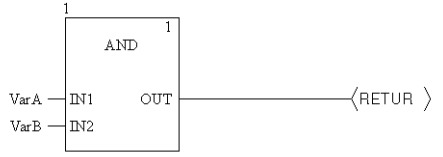
Note

In the **Tools > Options** dialog box, in the **Data and Languages** tab, if the **Automatically affect a variable on a new graphical object** check box is activated, its properties dialog

box is automatically opened when an object is placed. The procedure explained here is valid for **deactivated** check boxes.

Executing a return

Process the following steps to execute a return:

Step	Action
1	Create a subroutine, page 522 or a DFB, page 1256 in the FBD programming language.
2	Create the logic for the subroutine/DFB.
3	Create the logic for return. Example: 
4	Select the return object.
5	Click the target position in the FBD section. or Use the **arrow keys** to move the cursor to the target position, and press **Enter**. **Result:**The return object is inserted. Example: 
6	Create a connection between the return logic and return object using a graphical link, page 724. Example: 

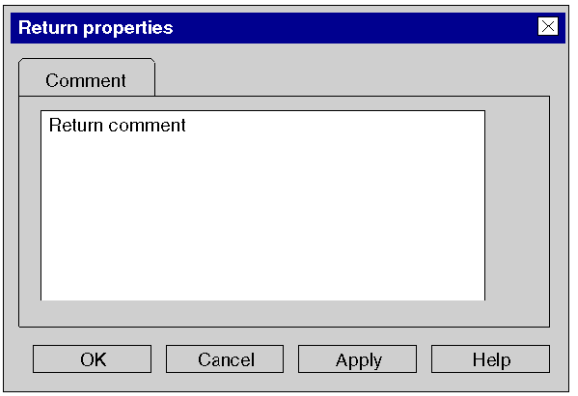
Return object properties dialog box

Calling the properties dialog box

see Displaying the properties, page 666

Structure of the properties dialog box

Representation of the properties dialog box:



Elements of the properties dialog box:

Element	Description
Text box	Enter a comment about the return object.
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Calling a Subroutine

Overview

This section describes how to call a subroutine in the FBD programming language.

Calling a subroutine

Introduction

In FBD, subroutines are called using a special subroutine block, see also section Subroutine call (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.

The subroutine to be called must be located in the same task as the FBD section calling.

For an unconditional subroutine call, page 710, EN and ENO of the subroutine block are hidden, page 694 or the input EN is assigned the value 1.

For a conditional subroutine call, page 711, EN and ENO of the subroutine block are shown, page 694 and the input EN is linked with a Boolean FFB output.

For a parallel call of several subroutines, page 712, EN and ENO of the subroutine blocks are shown, page 694 and the output ENO is connected to the EN input of the next subroutine block.

Subroutine calls are expansions to IEC 61131-3 and must be explicitly enabled via the **Tools > Project Settings** dialog box, in the **Language extensions** tab by activating the **Allow subroutines** check box.

Activating the placement mode for subroutine blocks

You have the following options to activate the placement mode for subroutine blocks:

- Use the **Edit > New > Subroutine** menu command.
- Use the **Subroutine** menu command from the shortcut menu.

  or

- Select the  symbol.

The active placement mode for subroutine blocks is indicated by the  cursor symbol.

Note

In the **Tools > Options** dialog box, in the **Data and Languages** tab, if the **Automatically affect a variable on a new graphical object** check box is activated, its properties dialog box is automatically opened when an object is placed. The procedures explained here are used when the check box is **cleared**.

Placing subroutine blocks:

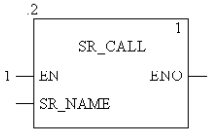
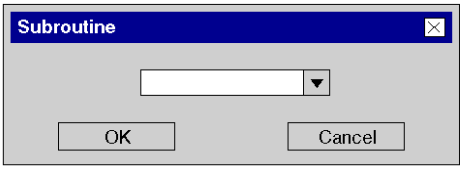
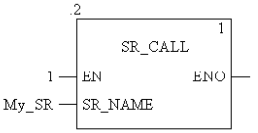
Placing subroutine blocks:

Step	Action
1	Activate the placement mode for subroutine blocks, see also Activating the placement mode for subroutine blocks, page 709.
2	Click the target cell in the FBD section. or Use the **arrow keys** to move the cursor to the target position, and press **Enter**. **Result:** The selected subroutine block is inserted and the selection mode is active again.
3	• Placing other subroutine blocks: Click the target cell in the FBD section. or Use the **arrow keys** to move the cursor to the target position, and press **Enter**. • To return to selection mode: Press the **Esc** key. • To insert other objects: Select the object you wish to insert.

Unconditional subroutine call

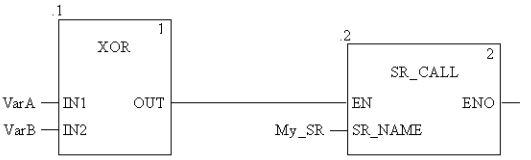
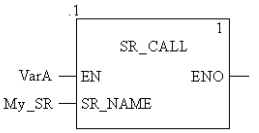
Unconditional subroutine call:

Step	Action
1	Place a subroutine block in the section, see also Placing subroutine blocks:, page 710.
2	Place the mouse pointer on the subroutine block. **Result:** The mouse pointer changes its symbol to  .
3	Double-click the `EN` input of the subroutine block. **Result**: A data selection box is opened.
4	Enter `1` or `TRUE` for the unconditional subroutine call.

Step	Action
	Example: 
5	Double-click the `SR_NAME` input of the subroutine block. **Result**: A selection dialog box for selecting the subroutine to be called is opened. Example: 
6	Enter the name of the subroutine, or click the ▾ symbol and select the subroutine to be called from the list. Confirm the entry with **Enter**. **Result:** The subroutine call is inserted into the section and a syntax and semantics check, page 752 is performed. Example:  If the subroutine has already been created (see also Create a Sub-Program Section (SR), page 522), the procedure ends here. If the subroutine has not yet been created, continue with step 7.
7	Create the subroutine, see alsoCreate a Sub-Program Section (SR), page 522.

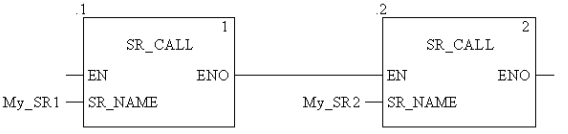
Conditional subroutine call

Conditional subroutine call:

Step	Action
1	Execute the Unconditional subroutine call, page 710 procedure, but do not assign a value to the EN input.
2	Create a Boolean logic for the subroutine call.
3	Connect the logic's Boolean output to the EN input of the subroutine block. Example:  **Note**: To execute the conditional subroutine call you can also allocate any Boolean variable to the EN input. Example: 

Parallel call of several subroutines

Calling several subroutines:

Step	Action
1	Execute the Unconditional subroutine call, page 710 procedure for all subroutines to be called in parallel.
2	Connect the ENO outputs and the EN inputs of the subroutine block. Example: 
3	Possibly create a Boolean logic for the first subroutine block.

Jumps within the Current Section

Overview

This section describes jumps within the current FBD section.

Jump within the current section

Introduction

Jump objects execute jumps to a jump target, page 718 within the current section.

When the status of the left link is 1, a jump is made to a label (in the current section).

To generate a conditional jump, a jump object is linked to a Boolean FFB output.

To generate an unconditional jump, the jump object is assigned the value 1 using the `AND` function for example.

Selecting the jump object

You have the following options to select jump objects:

- Use the **Edit > New > Jump** menu command.

  or

- Choose the  symbol.

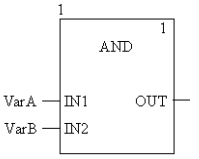
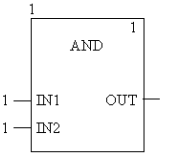
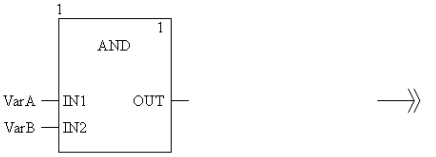
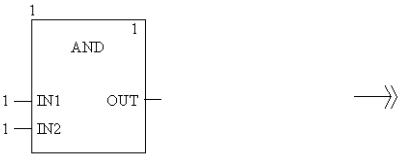
The active placement mode for jump objects is indicated by the  cursor symbol.

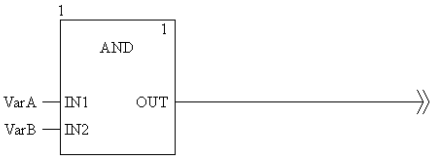
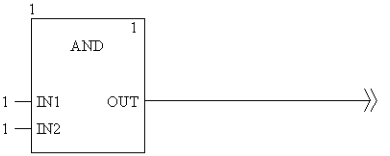
Note

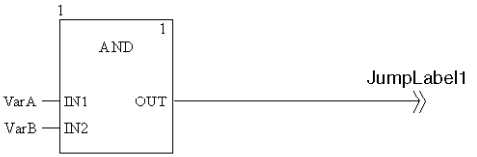
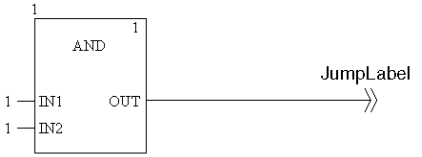
In the **Tools > Options** dialog box, in the **Data and Languages** tab, if the **Automatically affect a variable on a new graphical object** check box is activated, its properties dialog box is automatically opened when an object is placed. The procedure explained here is valid for **deactivated** check boxes.

Executing a jump

Process the following steps to execute a jump:

Step	Action
1	Create the logic for the jump. Example: Logic for conditional jump:  Example: Logic for unconditional jump: 
2	Select the jump object.
3	Click the target position in the FBD section. or Use the **arrow keys** to move the cursor to the target position, and press **Enter**. **Result:**The jump object is inserted. Example: Logic for conditional jump:  Example: Logic for unconditional jump: 
4	Create a link between the jump logic and jump object using a graphical link, page 724.

Step	Action
	Example: Logic for conditional return:  Example: Logic for unconditional return: 
5	To enter the jump target, open the properties dialog box, page 666 of the jump object. **Result**: The properties dialog box for the jump object is opened, see also Jump object properties dialog box, page 716.

Step	Action
6	You have the following options to enter the jump target: • You can now enter the name of the jump target and confirm using **Enter**. or • You can select the name of the jump target from the list of available jump targets using the ▼ symbol and confirm using **Enter**. **Result:** The jump target is inserted and a syntax and semantics check, page 654 is performed. Example of a conditional jump:  Example of an unconditional jump:  If the jump target has already been defined (see also Jump target, page 718), the procedure ends here. If the jump target has not yet been declared, continue with step 7.
7	Specify the jump target, see also Jump target, page 718.

Jump object properties dialog box

Calling the properties dialog box

see Displaying the properties, page 666

Structure of the properties dialog box

The properties dialog box consists of two tabs:

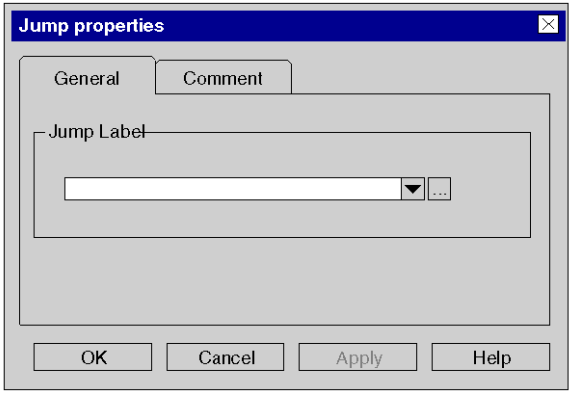
- **General**

  You can enter the jump target of the jump, page 713 in this tab.
- **Comment**

  A comment about the jump object can be entered in this tab.

General tab

Representation of the **General** tab:

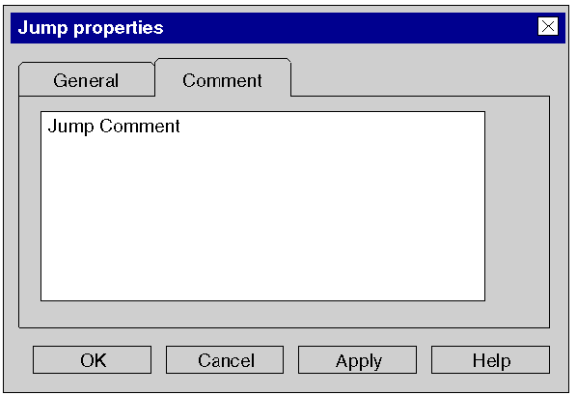


Elements of the **General** tab:

Element	Description
Jump Label	Enter the jump target in this text box, see also Definitions of Jump Labels, page 718.
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Comment tab

Representation of the **Comment** tab:



Elements of the **Comment** tab:

Element	Description
Text box	Enter a comment about the jump object.
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Definition of the Jump Target (Jump Labels)

Overview

This section describes how to define jump targets in the FBD programming language.

Definitions of Jump Labels

Introduction

Jump labels create a jump target for jumps, page 713 within the current section.

Jump labels are represented as text with a colon at the end.

The text is limited to 32 characters and must be unique across the entire section. The text must conform to general naming conventions.

Jump labels can only be placed between the first two grid points on the left edge of the section.

NOTE: Jump labels may not "dissect" networks, i.e. an assumed line from the jump label to the right edge of the section may not be crossed by any object. This also applies to the connections.

To find a specific jump label use the **Go to** dialog box.

Selecting the jump label

You have the following options to select jump labels:

- Use the **Edit > New > Jump Label** menu command.

  or

- Choose the JL symbol.

The active placement mode for jump labels is indicated by the JL cursor symbol.

Note

In the **Tools > Options** dialog box, in the **Data and Languages** tab, if the **Automatically affect a variable on a new graphical object** checkbox is activated, its properties dialog box is automatically opened when an object is placed. The procedure explained here is valid for **deactivated** checkboxes.

Placing a jump label

Placing a jump label:

Step	Action
1	Select the jump label.
2	Click with the mouse on the target position between the first two grid points on the left edge of the FBD section. or Use the **arrow keys** to move the cursor to the target positions between the first two grid points on the left edge of the FBD section, and press **Enter**. **Result:** The jump label is inserted and the selection mode is active again. Example: `???:`
3	To insert further jump objects: • Click the target cell in the FBD section. or Use the **arrow keys** to move the cursor to the target position, and press **Enter**. • To return to selection mode: Press the **Esc** key. • To insert other objects: Select the object you wish to insert.

Definition of the Name

Definition of the Name:

Step	Action
1	To enter the name, open the properties dialog box, page 666 for the jump label. **Result**: The jump label properties dialog box is opened, see also Jump label objects properties dialog box, page 721.
2	You have the following options to enter the name of the jump label: • You can now enter the name and confirm using **Enter**. or • You can select the name of the jump from the list of available jumps using the ▾ symbol and confirm using **Enter**. **Result:** The name is inserted and a syntax and semantics check, page 654 is performed. Example: `JumpLabel1:`

Jump label objects properties dialog box

Calling the properties dialog box

see Displaying the properties, page 666

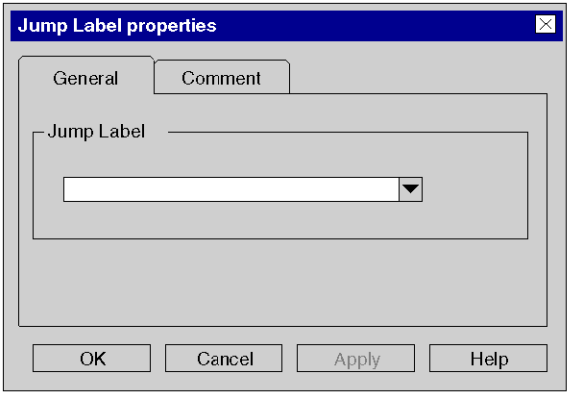
Structure of the properties dialog box

The properties dialog box consists of two tabs:

- **General**
  You can enter the name of the jump label, page 719 in this tab.
- **Comment**
  A comment about the jump label can be entered in this tab.

General tab

Representation of the **General** tab:

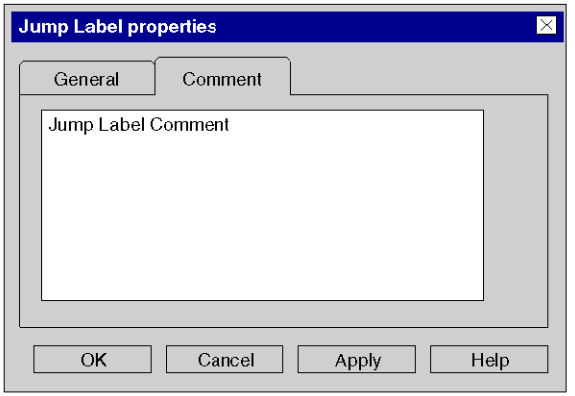


Elements of the **General** tab:

Element	Description
Jump Label	You can enter the name of the jump label in this list box. The text is limited to 32 characters and must be unique across the entire section. The text must conform to general naming conventions.
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Comment tab

Representation of the **Comment** tab:



Elements of the **Comment** tab:

Element	Description
Text box	Enter a comment about the jump label object.
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Editing Links

Overview

Editing links in the FBD programming language.

Links

Introduction

Links are vertical and horizontal connections between FFBs.

Attention should be paid to the following programming information:

- Links can be used for every data type.
- The data types of the inputs/outputs to be linked must be the same.
- Several links can be connected with one FFB output. However, only one may have an FFB output.
- Only inputs and outputs can be linked together. A link between several outputs cannot be made. This means that OR links are not possible using links in FBD. An `OR` function should always be used.
- An overlap of links with other objects is permitted.
- No loop can be configured with links because in this case the execution sequence in the section cannot be clearly determined. Loops must be terminated with actual parameters (see Terminating loops (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual)).
- To avoid links crossing each other, links can also be represented in the form of connectors, page 627.

  NOTE: A detailed description about Links (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) can be found in the Reference manual.

Selecting links

You have the following options to activate the placement mode:

- Use the **Edit > New > Link** menu command.
- Press the **F6** key.

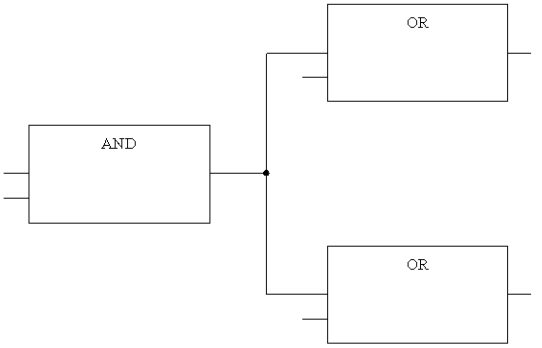
  or

- Select the  symbol.

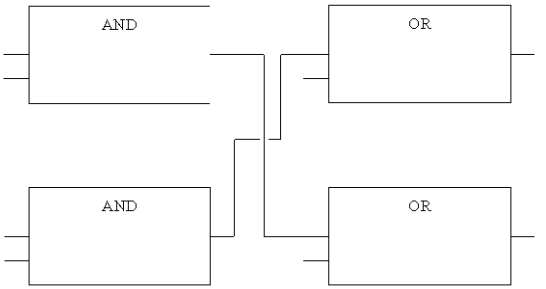
The + and  cursor symbols indicate which mode is active.

Representation

The link points are identified by a filled-in circle.



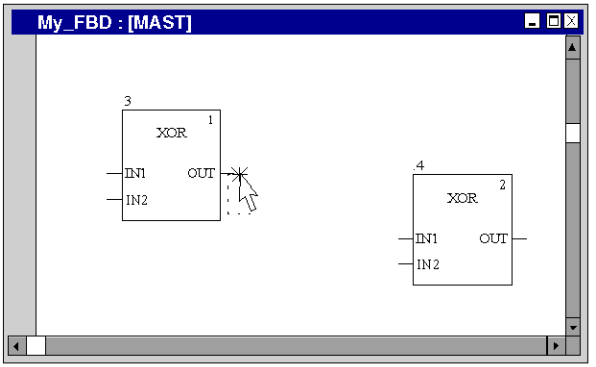
Crossed links are indicated by a "broken" link.

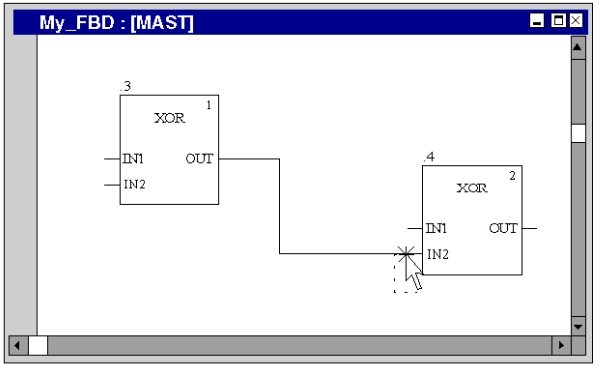
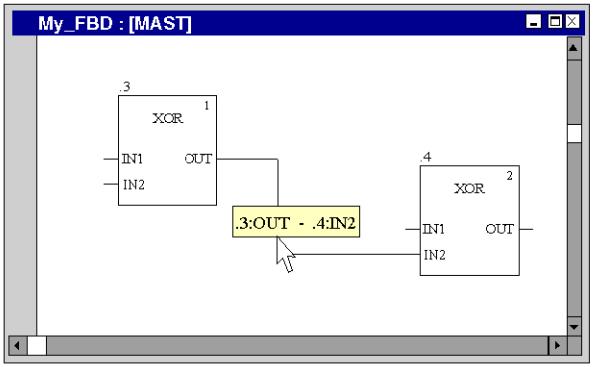


Placing a Link

Placing a link using the mouse

Carry out the following steps to place a link using the mouse:

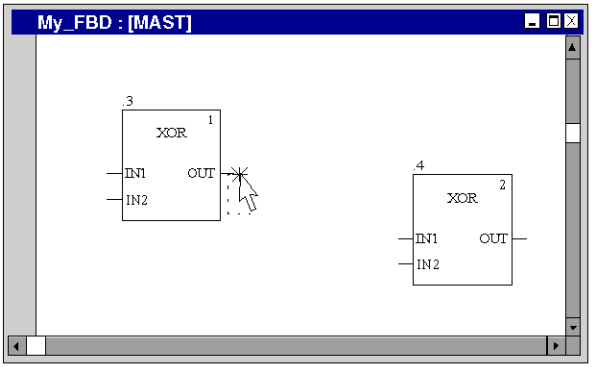
Step	Action
1	Activate the placement mode for links, see also Links, page 723.
2	Place the mouse pointer on the start point of the link. **Note:** The correct position is shown by the mouse pointer symbol. Example: 

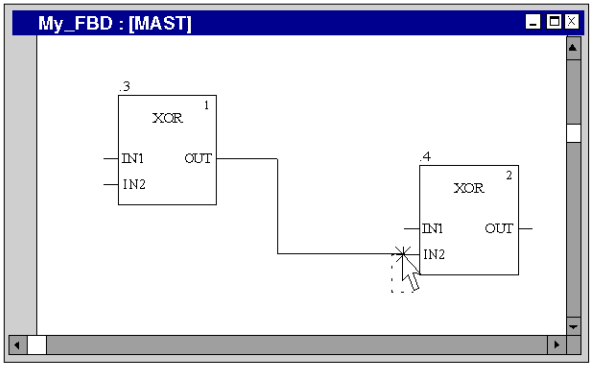
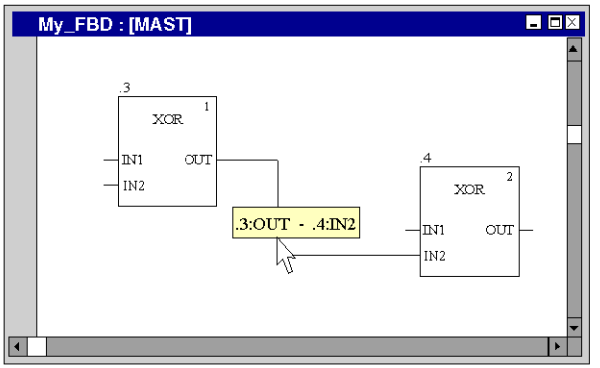
Step	Action
3	Click the left mouse button on the start point and drag mouse pointer to the target point of the link. Example:  **Tips:** • By clicking between the start point and target position you can place intermediate points; for example, to avoid crossing other objects. • You can leave this mode at any time by pressing **Esc**.
4	Click on the target position with the left mouse button. **Result:** The link is inserted. When the cursor is placed over a link, the Quickinfo (Tooltip) shows the source and destination of the link. Example:  See also Editing Links, page 729

NOTE: You can also place a link using the mouse in select mode, page 724.

Placing a link using the keyboard

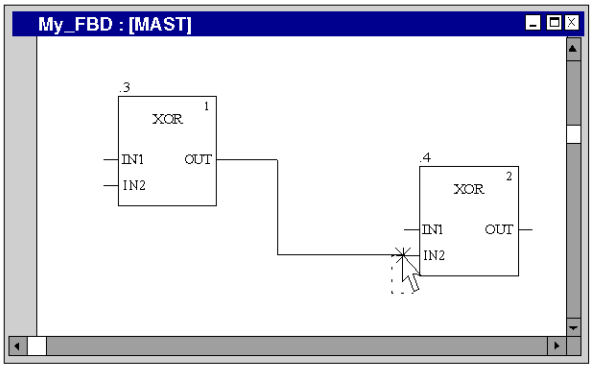
Carry out the following steps to place a link using the keyboard:

Step	Action
1	Activate the placement mode for links by using:, see also Links, page 723.
2	Use the **cursor keys** to move the cursor to the start point of the link, and press **Enter**. **Note:** The correct position is shown by the cursor symbol. Example: 

Step	Action
3	Use the **cursor keys** to move the cursor to the target position of the link. Example:  **Tips:** • By pressing the **Enter** key between the start point and target position you can place intermediate points; for example, to avoid crossing other objects. • You can leave this mode at any time by pressing **Esc**.
4	Press **Enter**. **Result:** The link is inserted. When the cursor is placed over a link, the Quickinfo (Tooltip) shows the source and destination of the link. Example:  See also Editing Links, page 729

Placing a link using the mouse in select mode

Carry out the following steps to place a link using the mouse in select mode:

Step	Action
1	Activate the select mode by using: • the **Edit > Select Mode** • the  symbol, or • the **Esc** key
2	Place the mouse pointer on the start point of the link. **Note:** The correct position is shown by the mouse pointer symbol. Click the left mouse button change the mode from select mode to placement mode.
3	Keeping the left mouse button pressed on the start point, drag mouse pointer to the target point of the link. Example:  **Result:** The link is inserted.

Editing Links

Editing links

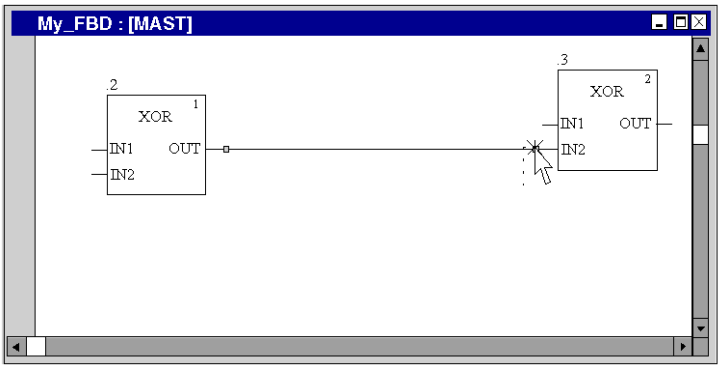
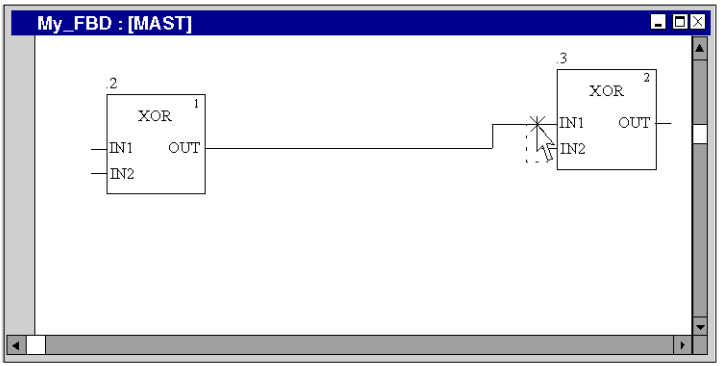
Links consist of one or more segments connected to one another.

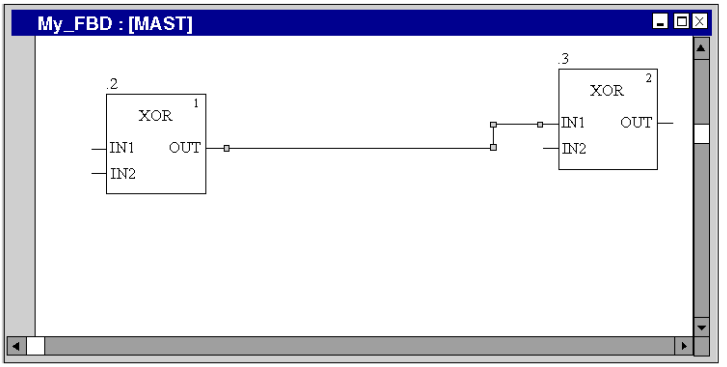
When cutting, page 659, deleting, page 658, copying, page 659 and moving, page 662 links, the action is executed for the entire link.

If a link is selected, the target points are displayed at their beginnings, ends and at every direction change

Modifying connections

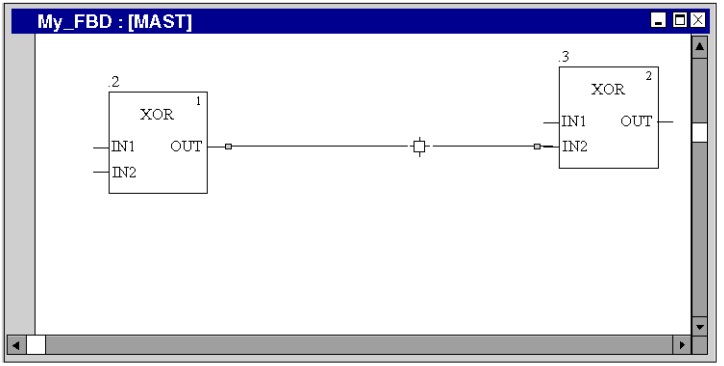
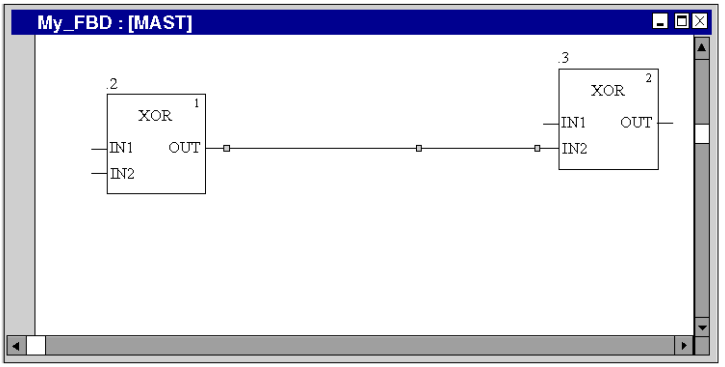
Modifying connections:

Step	Action
1	Select the link and place the mouse pointer on the connection to be modified. Example: 
2	Left-click the connection to be modified and drag the mouse pointer to the new connection. Example: 
3	Left-click the new target position. **Result:** The link route is modified. Example:

Step	Action
	

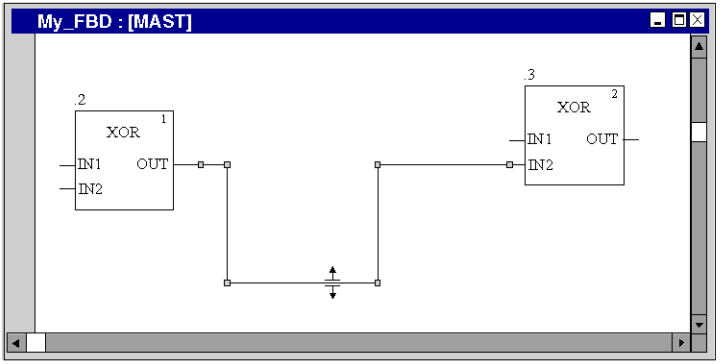
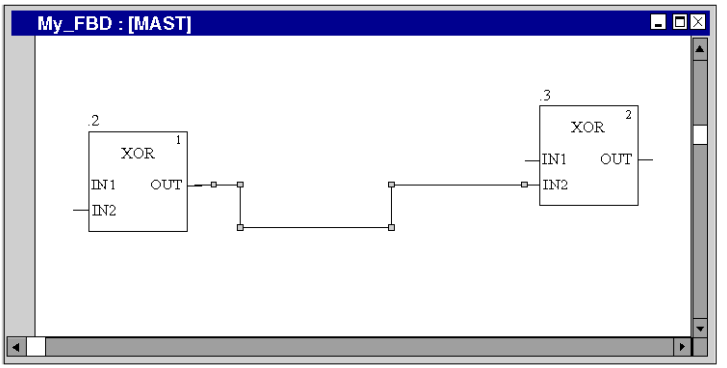
Creating sizing handles

The creation of angles in links can only be made using sizing handles. If no sizing handles are available they can be created. Process the following steps to create sizing handles in links:

Step	Action
1	Select the link.
2	Press the **Ctrl** key and place the mouse pointer at the target position. **Note:** The correct position is shown by the mouse pointer symbol. Example: 
3	Left-click the target position. **Result:** The sizing handle is created. 

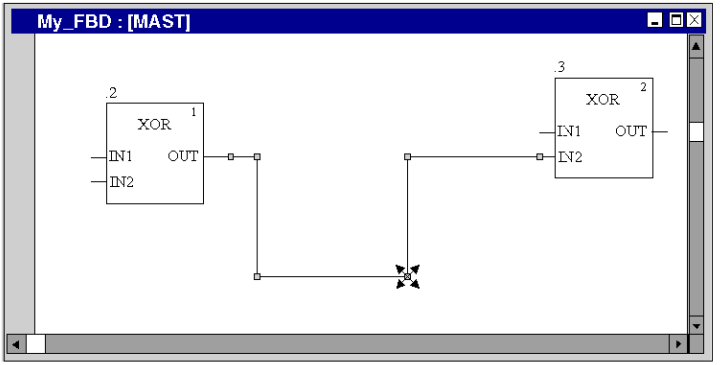
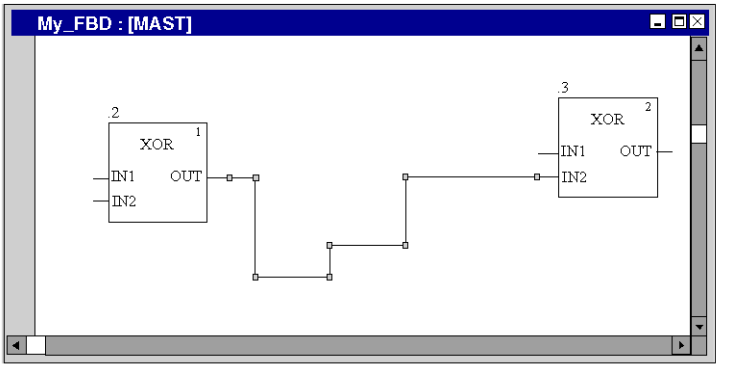
Moving links vertically/horizontally

Process the following procedures to move a link segment vertically or horizontally:

Step	Action
1	Select the link.
2	Place the mouse pointer on the item to be moved. **Note:** The correct position is shown by the mouse pointer symbol. Example: 
3	Click the left mouse button and drag the segment to the target position.
4	Release the mouse button. **Result:** The change is made. 

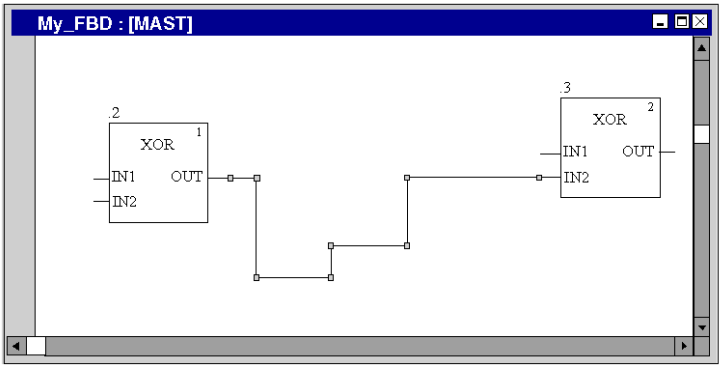
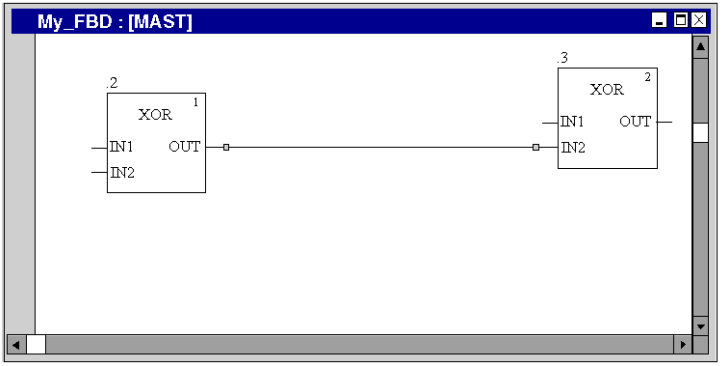
Creating angles in FFB links

Process the following steps to create angles in links:

Step	Action
1	Select the link.
2	Place the mouse pointer on a source position. **Note:** The correct position is shown by the mouse pointer symbol. Example: 
3	Click the left mouse button and drag the source point to the target position.
4	Release the mouse button. **Result:** The change is made. 

Determining the optimum connection route

Determining the optimum link route:

Step	Action
1	Select the link. Example: 
2	Use the **Edit > Link > Route** menu command. or Use the **Route** menu command from the shortcut menu. **Result:** The current link route is changed to the optimum route. Example: 

Entering Comments

Overview

This section describes how to enter comments in the FBD programming language.

Entering comments

Introduction

Comments can be placed in FBD in the form of text objects.

Text objects can overlap other objects.

The size of the object, depending on the size of the text, can be extended vertically and horizontally to fill additional grid units.

The same rules apply for entering text and navigating within text objects as for editing ASCII text in standard text editors. (To create a line break in text objects you must press the **Ctrl** +**Enter** key combination).

NOTE: Keep in mind that each change to a comment (e.g. change to a comment text, change to the size of the text object) makes it necessary to recreate the section involved (**Generation > Generate project**).

Displaying or hiding the comments

Use the  button in the toolbar to show or hide the comments.

The background color of the button in blue indicates that the comments are hidden.

NOTE: To select or place text objects, check that the  button in the toolbar is not activated.

Selecting the text object

You have the following options to select text objects:

- Use the **Edit > New > Comment** menu command.
- Use the **Comment** menu command from the shortcut menu.

- Press the **F8** key.

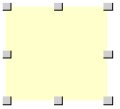
  or

- Select the  symbol.

The active placement mode for text objects is indicated by the  cursor symbol.

Placing text objects:

Placing text objects:

Step	Action
1	Select the text object.
2	Click the target position in the FBD section. or Use the **arrow keys** to move the cursor to the target position, and press **Enter**. **Result:** The selected text object is inserted and the selection mode is active again. Example: 
3	Enter the comment.
4	Confirm the text entered by: • Clicking with the mouse outside of the text object or • Pressing the **Enter** key.
5	To insert additional text objects: • Click the target position in the FBD section. or Use the **arrow keys** to move the cursor to the target position, and press **Enter**. • To return to selection mode: Press the **Esc** key. • To insert other objects: Select the object you wish to insert.

Online functions

Overview

This section describes the online functions of the FBD programming language.

Online functions

Online functions

The description of online functions is found in Debugging in Function Block Diagram (FBD) Language, page 1430.

FBD Reference Data Type

Reference Data Type in FBD

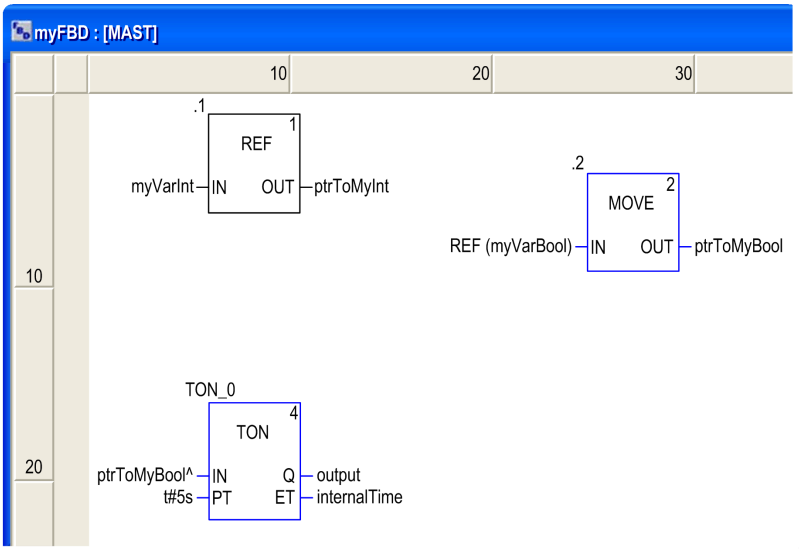
Introduction

References to variables can be assigned in FBD application programs.

A reference can be de-referenced in FBD application programs.

For more information, refer to Reference Data Type (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).

FBD Examples using the Reference Data Type:



The EF, `REF`, takes a variable as an input parameter (type *ANY*) and provides its address as output parameter (type *REF_ANY*).

The EF, `MOVE`, can assign a reference to a variable to another reference. If the input parameter is (*REF(MyVariable)*), `MOVE` has the same effect as the `REF` EF.

The EF, `TON`, takes *ptrToMyBool^* as an input parameter to dereference the reference ptrToMyBool.

Export/Import

Overview

This section describes the Import/Export functions of the FBD programming language.

Export/Import FBD Sections

Export/Import

The description for exporting/importing sections is found in the chapter Import / Export, page 1686.

Customize FBD Editor

Overview

This section describes how to customize the toolbar and colors of the FBD editor.

Customize FBD Editor Colors

Overview

To customize FBD editor colors, open the **Tools > Colors Settings...** dialog box.

The **Color Settings** dialog box allows to:

- Modify the color of the selected item in the list.
- Reset the colors with initial values.
- Import an *.ini file that defines the colors associated to the FBD editor.
- Export the definition colors associated to the FBD editor.

Color Setting File

Depending on the operating system of the PC, the color setting file FBDColors.ini is located in one of the following folders:

- `c:\Program Files\Schneider Electric\Control Expert ••.•\`
- `c:\Program Files (x86)\Schneider Electric\Control Expert ••.•\`

For more detail on destination folder when Control Expert is installed, refer to the *EcoStruxure™ Control Expert, Installation Manual*.

NOTE: Make a copy of this file before modifying it.

Description

This table describes the default color values for each parameter in the `FBDColors.ini` file:

Parameter	Default value (R, G, B)	Description
BkAnimation	`192,192,192`	Editor background color when animation is ON and you are connected in programming mode.
StatementError	`255,0,0`	Element in error is underlined with this color.
InspectBk	`0,255,255`	Inspect window background color
InspectText	`0,0,0`	Inspect window text color
InspectBkMin	`255,255,0`	Inspect window background color when the value is less than the minimum value defined in the inspect window settings.
InspectBkMax	`255,0,255`	Inspect window background color when the value is greater than the maximum value defined in the inspect window settings.
BkAnimMonitoring	`255,211,211`	Editor background color when animation is ON and you are connected in monitoring mode.
AnalyzeError	`0,0,255`	Drawing color when an element generates an analyze error.
GraphAnimBoolTrue	`0,150,0`	Color of true BOOL in animation mode.
GraphAnimBoolFalse	`255,0,0`	Color of false BOOL in animation mode.
BkComment	`255,255,204`	Background color of comment area
BkCommentEdit	`225,225,225`	Background color of comment area when you are editing it.
BkAnimNumeric	`255,255,0`	Background color of numeric variables
BlockBreakpoint	`128,0,0`	Bullet color in the element where the breakpoint is set.
BlockCurrentStep	`255,255,0`	Color of the current element in debug mode
AnchorLink	`128,0,0`	Color of anchor link
Background	`255,255,255`	Editor background color.
FBDFgExpression	`0,0,0`	Color for FBD text
FBDBgExpression	`255,255,255`	Background color for FBD text

FBD Favorite Bar

Overview

In the current FBD editor toolbar, the F button allows to manage the toolbar **FBD Favorite Bar** which can contain a maximum of 20 EF, EFB or DFB for direct call.

Showing and Hiding the Toolbar

You have two options to show the toolbar **FBD Favorite Bar** (to open it):

- Right-click in the common toolbar and select the toolbar from the shortcut menu.
- Open the **Tools > Customize** dialog box and select the toolbar,

You have three options to hide the toolbar **FBD Favorite Bar** (to close it):

- Right-click in the common toolbar and deselect the toolbar from the shortcut menu,
- Open the **Tools > Customize** dialog box and deselect the toolbar,
- With the close symbol (x) if the toolbar is non-anchored in the current FBD editor toolbar.

Managing the Toolbar

To manage the toolbar **FBD Favorite Bar**, opens the **Add/Delete Favorite** dialog box by clicking the F button.

The **Add/Delete Favorite** dialog box allows to:

- Add component via the **FFB Type Selection** dialog box.
- Delete component.
- Import an `*.ini` file that define the components for the **FBD Favorite Bar**.
- Export the components of the **FBD Favorite Bar**.

  NOTE: The components of the toolbar **FBD Favorite Bar** are saved in the `Favorite.ini` file.

  For example with Control Expert 14.0, the `Favorite.ini` file is located in `C:\ProgramData\Schneider Electric\Control Expert\14.0\Favorite\` folder.

LD Editor

What's in This Chapter

Creating a program with LD programming language 745
Editing Contacts 775
Editing Coils 783
Editing compare blocks 794
Editing operation blocks 800
Editing FFBs 806
Returning from a Subroutine or DFB 840
Jumps within the Current Section 843
Definition of the Jump Target (Jump Labels) 848
Editing Links 853
Entering Comments 880
Online functions 882
LD Reference Data Type 882
Export/Import 883
Customize LD Editor 884

Overview

This chapter describes the menus and dialog boxes of the LD editor.

For a description of the syntax for the LD programming language, please refer to Ladder Diagram LD (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).

Creating a program with LD programming language

Overview

This section describes the basics for creating a program in the LD programming language.

Structure of an LD program (ladder diagram)

Introduction

The structure of an LD program corresponds to a rung for relay switching.

The left power rail is located on the left side of the LD editor. This left power rail corresponds to the phase (L conductor) of a rung. With LD programming, in the same way as in a rung, only the LD objects which are linked to a power supply, that is to say connected with the left power rail, are "processed". The right power rail corresponds to the neutral conductor.

All contacts and FFB inputs must be directly or indirectly connected with the left power rail and all coils and FFB outputs must be directly or indirectly connected with the right power rail to allow current flow.

A group of connected objects that are not connected to other objects (except for the power rails), is called a network or rung.

Properties of an LD program

Properties of an LD program:

- An LD section contains 11-63 columns and 17-3998 lines.

  The default number of columns for a new section is set via the **Tools > Project Settings** dialog box, in the **Editors** tab in the **Number of columns** text box.

  The number of columns can be increased, page 774 or reduced, page 775 during the creation of section contents.

  The number of lines can be increased, page 773 or reduced, page 773 during the creation of section contents.
- The width of the cells can be managed, page 775 from one to three standard cell widths.
- LD programs are cell oriented, i.e. only one object can be placed in each cell.
- The processing sequence of the individual objects in an LD program is determined by the data flow within the section. Networks connected to the left power rail are processed from top to bottom (connection to the left power rail). Separate networks within the section are processed in the order in which they are placed (top to bottom). Refer also to section *Execution order and signal flow* (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual)).
- A syntax and semantics check, page 752 is performed immediately after the statement is entered. The result of this check is displayed in colored text and objects.
- Syntactically or semantically incorrect sections can be saved.

LD program objects

The objects of the LD programming language help to divide a section into a number of:

- Contacts, page 775,
- Coils, page 783,
- EFs and EFBs, page 806 (Elementary Functions and Elementary Function Blocks),
- DFBs, page 806 (derived function blocks)
- Procedures, page 806
- operate blocks, page 800,
- compare blocks, page 794,
- jumps, page 843 within the section and
- subroutine calls, page 790

These objects can be linked with each other through:

- Links, page 853 or
- actual parameters, page 817 (only FFBs).

Comments for the logic of the program can be added using text objects, page 880.

Edit and view functions

Edit and view functions of the LD editor:

- Selecting objects, page 760
- Deleting objects, page 763
- Cutting, page 763, copying, page 763 and pasting, page 764 objects
- Moving, page 766 objects (also between different LD/FBD sections)
- Replacing objects
- Undo, page 626 and Redo, page 627
- Using bookmarks, page 615
- Searching and replacing, page 1239 variables and function blocks and DFB instances

Input assistants

Input assistants for the LD editor:

- Mnemonics, page 769 for contacts, coils, blocks, links and jumps
- Function input assistants, page 828 for functions, function blocks and procedures

Online functions

Online functions of the LD editor:

- Display the current value, page 632.
- Setting breakpoints, page 1413
- Setting watchpoints, page 1419
- Step by Step, page 1416

Creating an LD program

Introduction

The LD editor window is made up of cells and 1 object can be placed in each one. The cells are separated visually using a grid, page 621 which can be switched off.

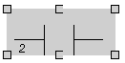
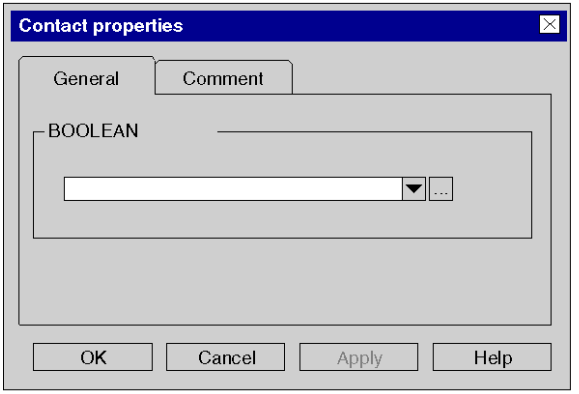
When creating a program with the keyboard, the selected (active) cell has a gray background, see also Programming with the keyboard, page 750.

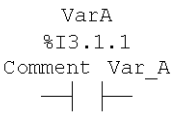
When creating a program with the mouse, this gray field is not relevant.

Creating an LD program with the mouse

Process the following steps to create an LD program with the mouse:

Step	Action
1	Create an LD section., page 502
2	Select the desired object (see also Pasting objects, page 853) using: • the menu commands in the **Objects > <Object>** menu or • the menu commands in the **Edit > New > <Object>** menu or • the symbols for the objects in the toolbar **Result:** The mouse pointer indicates the selected object (placement mode). **Note:** As soon as the text is entered several checks are made, such as syntax/semantic error checks, correct spelling of variable names etc. A detailed description of syntax is found in chapter Ladder Diagram (LD) (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual. The results of the checks are indicated by a colored folder, see also Syntax and semantics check during programming, page 752.

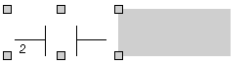
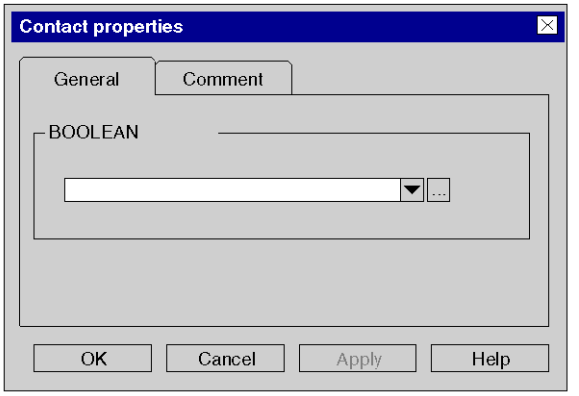
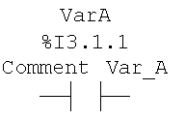
Step	Action
3	Click the target cells in the LD section. **Result:** The selected object is inserted and the selection mode is active again. Example:  **Note:** If you want to insert several objects of the same type, press and hold the **Ctrl** key and insert any number of selected objects by clicking with the left mouse button. Return to selection mode by clicking the symbol or pressing the **Esc** key.
4	To enter the respective actual parameters (variable/address), double-click the object. **Result:** The properties dialog box for the object is opened. Example: 

Step	Action
5	Enter the desired actual parameter and a comment (if required) and confirm the entries with **OK**. **Result:** In entry mode, the names of the associated variables are displayed above the object. In mixed display mode, the addresses and comments are displayed in addition to the names of the variables (if available), see also Mixed display mode, page 753. Example (entry mode):  Example (mixed display mode): 
6	Repeat these steps until all objects are entered.

Creating an LD program with the keyboard

Process the following steps to create an LD program with the keyboard:

Step	Action
1	Create an LD section., page 502
2	Place the field with the gray background on the cell where the object should be inserted, see also Navigating with the keyboard, page 758.
3	Select the desired object (see also Pasting objects, page 853) using the function keys for the objects. **Result:** The cursor symbol indicates the selected object. **Note:** As soon as the text is entered several checks are made, such as syntax/semantic error checks, correct spelling of variable names etc. A detailed description of syntax is found in chapter Ladder Diagram (LD) (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual. The results of the checks are indicated by a color folder, see also Representation, page 752.
4	Confirm the selection using the **Enter** key. **Result:** The selected object is inserted into the cell with the gray background and the gray field is automatically shifted to the next cell.

Step	Action
	Example:  Note: If you want to insert several objects of the same type, press and hold the **Ctrl** key and insert any number of selected objects with the **Enter** key. To return to selection mode, press the **Esc** key.
5	Press the **Enter** key to open a shortcut menu for the selected object.
6	To enter the respective actual parameters (variable/address), use the **Properties** menu command from the shortcut menu. **Result:** The properties dialog box for the object is opened. Example: 
7	Enter the desired actual parameter and a comment (if required) and confirm the entries with **OK**. **Result:** In entry mode, the names of the associated variables are displayed above the object. In mixed display mode, the addresses and comments are displayed in addition to the names of the variables (if available), see also Mixed display mode, page 753. Example (entry mode):  Example (mixed display mode): 
8	Repeat these steps until all objects are entered.

Syntax and Semantics Check during programming

Introduction

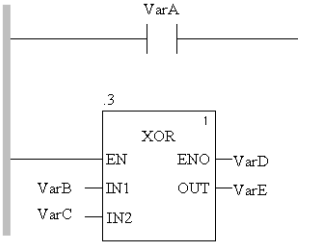
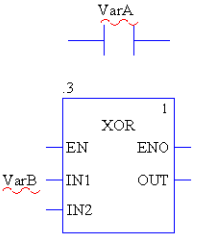
Syntax and semantics check is performed directly when creating the program.

The result of this check is displayed in three formats:

- directly in the program section with colored text
- as tooltip if the cursor is placed on incorrect text.
- In the output window, if **Generate > Analyze** is selected.

Representation

Meaning of colors and labels:

Color	Description	Example
Black	Syntax and semantics are correct	
Blue	Possible causes: • Associated variable is not declared • Data type of the variable does not match the data type of the pin • Input or output pin not connected	
Red wavy line	Erroneous text, e.g. variables which are not declared, variables with an incorrect data type, ...	

Tooltips

If the cursor is placed over incorrect text, the tooltip is displayed with a brief description of the cause of the error. It will also be displayed in the output window after the analysis.

Error message in the output window

Error message: `Accessed object is not a function block.`

Cause of error	Error Correction
Name of the function block was already used for a variable.	Rename the already used variable.

View for variables

Display modes for variables

The following view modes are available for variables:

- Input mode, page 754
- Mixed display mode, page 755

You can switch between the two modes using:

- The **View > Mixed Display Mode** menu command
- The **Ctrl**+**E** key combination.

A check-symbol displayed on the left of the **View > Mixed Display Mode** item indicates when the mixed display mode, page 755 is enabled.

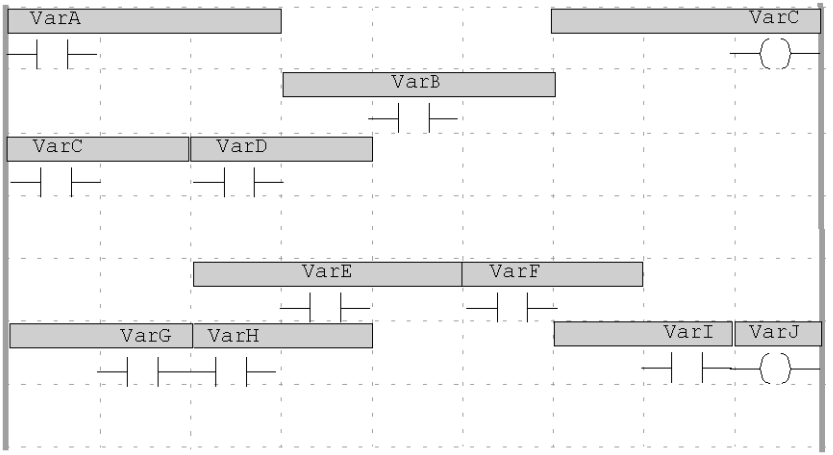
Variables display

Depending on the size set for the cells, it may be the case that variable names are not shown in full.

A maximum of 3 columns are available to display the variables. The space available depends on various factors, on both Input mode, page 754 & Mixed display mode, page 755

- On the position
  - If an element (e.g. a contact) was placed directly on the left power rail, columns 1-3 are available for the variable names.
  - If an element (e.g. a coil) was placed directly on the right hand power rail, columns 9-11 are available for the variable names.
  - If an element (e.g. a contact) was placed arbitrarily in the section, the column to the right and the column to the left of the element are available for the variable name.
- On the availability of the cells:
  - Only cells that are not occupied by other objects are available to display the variable name.
  - If an unused cell is between two objects, it is automatically reserved for the variable name of the left hand object and is not available to the right hand object (even if it is not used by the left hand object).

The gray fields indicate which area is available for the variable names of the individual objects.



Tip: There are two options for displaying the full variable names:

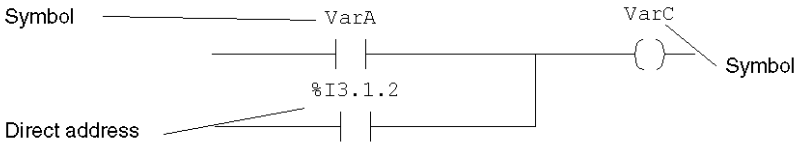
- Select the variable and the full variable name is shown in the status line.
- Place the cursor on the variable and a Tooltip will appear with type, name, address and comment of the variable.

Input mode

In input mode, only the symbolic name is displayed for variables and only the address for direct addresses:

- if you enter a symbol you will see in this mode a symbol even if you associate an address after in the data editor.
- if you enter a direct address you will see in this mode a direct address even if you associate a symbol after in the data editor.

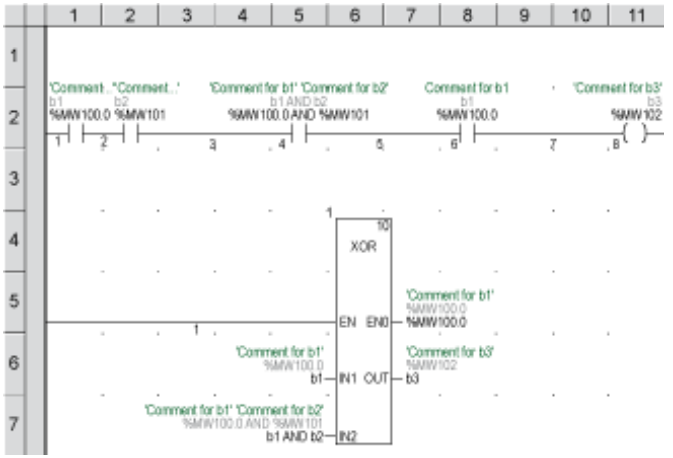
Example of a section in input mode:



Mixed display mode

In the mixed display mode, the comment, symbolic name and, if a unique association is possible, the address will be displayed for the variables.

Example of a section in Mixed Display Mode:



This functionality is available for variables for both LD objects (contact, coil, comparative module, operation module) and FFBs.

NOTE: When using structure elements, the entire comment (comment for root variables and comment(s) for the structure element(s)) or only the comment for the root variable may be displayed. To display the entire comment, select **Tools > Project Settings > Program > Common** and enable **Display complete comments for structure element**.

NOTE: To print one or several sections the option `mixed display` must be check in **Documentation configuration** of project.

For example, the following displays are possible:

Display	Description
'VariableA' VarA %I3.1.2	The located variable `VarA` with the comment `VariableA` has the address `%I3.1.2`. For this type of display the following requirements apply: • The display of comments is enabled. • The variable is a located variable. • A comment was defined for the variable. • An address is only associated with one symbolic name or the symbolic name was assigned to the actual parameter.
'Mounting.ManualStart' Motor.Start %I3.1.2	The structure element `Start` of the variable `Motor` is assigned to the address `%I3.1.2` assigned. The displayed comment consists of the comments for the root variable (`Motor`) and the comment(s) for the structure elements (`Start`). For this type of display the following requirements apply: • In **Tools > Project Settings > Program > Common** the option **Display complete comments for structure elements** is enabled. • The variable is a located variable. • A comment was defined for the variable. • An address is only associated with one symbolic name or the symbolic name was assigned to the actual parameter.
'Mounting' Motor.Start %I3.1.2	The element `Start` of the variable `Motor` is assigned to the address `%I3.1.2` assigned. The displayed comment is the comment for the root variable (`Motor`). For this type of display the following requirements apply: • In **Tools > Project Settings > Program > Common** the option **Display complete comments for structure elements** is disabled. • The variable is a located variable. • A comment was defined for the variable. • An address is only associated with one symbolic name or the symbolic name was assigned to the actual parameter.
'<n/a>' VarA %I3.1.2	The located variable `VarA` has the address `%I3.1.2`and no comment was defined for the variable. For this type of display the following requirements apply: • The variable is a located variable.

Display	Description
	• An address is only associated with one symbolic name or the symbolic name was assigned to the actual parameter.
'VariableA' VarA <n/a>	The variable VarA is an unlocated variable.
'<n/a' <...> %I3.1.2	An address with multiple assigned symbolic names was assigned to the object as an actual parameter. Since in this case, a unique assignment of symbolic name to address is not possible, no symbolic name is displayed.
'VariableB' <n/a> %I3.1.2	The address was a assigned a comment, but no symbolic name.
'<n/a>' VarA:VarB<VarC <###> or '<n/a>' <###> %I3.1.2:%I3.1.3<%I3.1.4	Syntax error. The expression entered is not correct. For this reason no addresses or symbols can be displayed.

NOTE: The **Comments line height** to display additional information can be specified under **Tools > Project Settings > Program > Languages > LD > Mixed display**.

The column width can be adjusted, page 775.

Adjusting variables font size

You can increase or decrease the size of the font for the variables using the following buttons on the toolbar  .

NOTE: The text inside the blocks and the instance name cannot be changed.

Navigating with the Keyboard

Using the keyboard

The following keys and key combinations are provided for navigation:

Key combinations	Motion
Cursor left	Moves the gray field in a cell to the left by one cell and selects the contents of that cell. If the first line is reached, the previous line is automatically selected.
Cursor right	Moves the gray field in a cell to the right by one cell and selects the contents of that cell. If the last line is reached, the next line is automatically selected.
Cursor up	Moves the gray field in a cell to the up by one cell and selects the contents of that cell
Cursor down	Moves the gray field in a cell to the down by one cell and selects the contents of that cell
Ctrl+Left arrow	Moves the cursor 1 pixel left and moves the gray field to the cursor position
Ctrl+Right arrow	Moves the cursor 1 pixel right and moves the gray field to the cursor position
Ctrl+Up arrow	Moves the cursor 1 pixel up and moves the gray field to the cursor position
Ctrl+Down arrow	Moves the cursor 1 pixel down and moves the gray field to the cursor position
Ctrl+Shift+Left arrow	Moves the cursor one cell left
Ctrl+Shift+Right arrow	Moves the cursor one cell right
Ctrl+Shift+Up arrow	Moves the cursor one cell up
Ctrl+Shift+Down arrow	Moves the cursor one cell down
Shift+Left arrow	Moves the selected object and the gray field to the left by one cell This also applies to several selected objects, the gray field must be behind one of the selected objects.
Shift+Right arrow	Moves the selected object and the gray field to the right by one cell. This also applies to several selected objects, the gray field must be behind one of the selected objects.
Shift+Up arrow	Moves the selected object and the gray field up by one cell. This also applies to several selected objects, the gray field must be behind one of the selected objects.

Key combinations	Motion
Shift+Down arrow	Moves the selected object and the gray field down one cell. This also applies to several selected objects, the gray field must be behind one of the selected objects.
Home	Positions the gray field in the first column and shows this new location.
End	Positions the gray field in the last column and shows this new location.
Ctrl+Home	Positions the gray field in the upper left hand cell in this section and shows this new location.
Ctrl+End	Positions the gray field in the upper right hand cell in this section and shows this new location.
Page Up	Scrolls the position of the gray field one page up and shows this new position.
Page Down	Scrolls the position of the gray field one page down and shows this new position.
Ctrl+Page Up	Scrolls the position of the gray field one page left and shows this new position.
Ctrl+Page Down	Scrolls the position of the gray field one page right and shows this new position.
Ctrl+Alt+Page Up	Displays the previous section (including via the **View > Previous Section**menu).
Ctrl+Alt+Page Down	Displays the next section (including via the **View > Next Section**menu).
Spacebar	Selects or deselects the object in the cells highlighted in gray. If several objects are in the gray highlighted field, the next object in the cell is selected every time you press the **space bar**. The position of the gray field is not affected by this function. See also Selecting objects, page 760.
Ctrl + space bar	If several objects are in the gray highlighted cells, every time you press the shortcut key combination **Ctrl+space bar** the next object is inserted in the currently selected cells (multiple selection). The position of the gray field is not affected by this function.
Shift + space bar	The position of the gray field is moved one cell to the right and all objects in the currently selected cells highlighted in gray (multiple selections) are inserted.
Enter	In Select mode: Opens the properties dialog box for the selected contact / coils. When you are finished press **Enter** to cancel press **Esc**.

Key combinations	Motion
	In Insert mode: Inserts the selected object into the currently selected cells highlighted in gray and moves the position of the gray field one cell to the right.
Alt+Enter	Opens the properties dialog box for the selected object/pin.
Tab	Select the next pin in an FFB if the FFB or an FFB pin is selected.
Esc	Activates Select mode.

Selecting objects

Select and placement mode

Objects are selected in select mode.

Select mode can be activated using:

- the **Objects > Select Mode** menu command
- the **Edit > Select Mode** menu command
- the  symbol,

  or
- the **Esc** key

The  cursor symbol indicates that select mode is active

Selecting an object

Using the mouse	Using the keyboard
Left-click the object you want to select.	Move the gray field to the cell with the object to be selected, see also Navigating with the keyboard, page 758. **Note:** If several objects are in the gray highlighted field, the next object in the cell is selected every time you press the **spacebar**.

Selecting several objects

Using the mouse	Using the keyboard
1. Click the left mouse button and keep it pressed. 2. Drag the mouse across the objects you want to select. or 1. Left-click the first object you want to select. 2. Press the **Ctrl** key and keep it pressed. 3. Left-click the next object you want to select. 4. Repeat these steps until all desired objects are selected.	Selecting several objects in **different** cells: 1. Move the gray field to the cell with the first object to be selected, see also Navigating with the keyboard, page 758. 2. Press the **Shift+spacebar** key combination. **Result:** The object is added to the current selection and the gray field is shifted one cell to the right. 3. Repeat these steps until all desired objects are selected. Selecting several objects in **one** cell: 1. Move the gray field to the cell with the objects to be selected, see also Navigating with the keyboard, page 758. 2. Press the **spacebar** until the first desired object is selected. 3. Press the **Ctrl+spacebar** key combination to select the next object in the cell. 4. Repeat these steps until all desired objects are selected.

Selecting by rows

Using the mouse	Using the keyboard
Selecting the contents of a row: 1. Left-click the number of the row you want to select in the vertical ruler. Selecting the contents of several lines: 1. Left-click the number of the first line you want to select in the vertical ruler. 2. Press the **Shift** key and keep it pressed. 3. Left-click the number of the last line you want to select in the vertical ruler. **Note:** Pressing the **Shift** key makes it possible to combine this procedure with the procedure for selecting columns.	-

Selecting by columns

Using the mouse	Using the keyboard
Selecting the contents of a column: 1. Left-click the number of the column you want to select in the vertical ruler. Selecting the contents of several columns: 1. Left-click the number of the first column you want to select in the vertical ruler. 2. Press the **Shift** key and keep it pressed. 3. Left-click the number of the last column you want to select in the vertical ruler. **Note:** Pressing the **Shift** key makes it possible to combine this procedure with the procedure for selecting lines.	-

Selecting all (entire contents of the section)

Using the mouse	Using the keyboard
Use the **Edit** > **Select All** menu command.	Press the **Ctrl**+**A** key combination.

Selecting all objects

Using the mouse	Using the keyboard
Left-click an empty space in the section.	1. Move the gray field to an empty space, see also Navigating with the keyboard, page 758. 2. Press the **spacebar**.

Deselecting objects

Using the mouse	Using the keyboard
Left-click an empty space in the section.	Move the gray field or press **the space bar**.

Deleting, cutting, copying, pasting, duplicating, and moving objects

Deleting objects

Deleting objects:

Using the mouse	Using the keyboard
1. Select, page 760 the object to be deleted. 2. Use the **Edit > Delete** menu command.	Deleting the selected object: 1. Select, page 760 the object to be deleted. 2. Press the **Delete** key. Deleting the object left of the gray field: 1. Press the **Backspace** key.
Result: The selected object is deleted. The actual parameters (and the links for FFBs) are deleted together with the object, even if they were not selected explicitly. Boolean links are only deleted if they are selected explicitly.	

Cutting objects

Cutting objects:

Using the mouse	Using the keyboard
1. Select, page 760 the object to be cut. 2. Use the **Edit > Cut** menu command. or Use the menu command **Cut** in the shortcut menu (right mouse button). or Click the  symbol.	1. Select, page 760 the object to be cut. 2. Press the **Ctrl+X** key combination.
Result: The selected object is cut from the section and copied to the clipboard. This is also the case for the actual parameters. FFB links are also cut. But they are only copied to the clipboard if they (and their partner objects) are selected explicitly.	

The cut object can be inserted, page 764 in any other position (also in another LD section).

Copying objects to the clipboard

Copying objects to the clipboard:

Using the mouse	Using the keyboard
1. Select, page 760 the object to be copied. 2. Use the **Edit > Copy** menu command. or Use the **Copy** menu command from the shortcut menu (right-click). or Click the  symbol.	1. Select, page 760 the object to be copied. 2. Press the **Ctrl+C** key combination.
Result: The selected object is copied to the clipboard. This is also the case for the actual parameters. FFB links are only copied to the clipboard if they (and their partner objects) are selected explicitly.	

The copied object can be inserted, page 764 in any other position (also in another LD section).

If the copied object is an EFB or a DFB instance, it can also be duplicated, page 765 in any other available position (also in another LD or FBD section).

Pasting objects from the clipboard

Pasting objects from the clipboard:

Using the mouse	Using the keyboard
1. Use the menu command **Edit > Paste**. or Use the menu command **Paste** in the shortcut menu (right mouse button). or Click the  symbol. 2. Left-click the target position.	1. Move the cursor to the target position using the arrow keys. 2. Press the **Ctrl+V** key combination. 3. Press **Enter**.
Note for pasting contacts and coils: If the target position already contains a contact or coil, the existing object is overwritten by the new object. **Note:** This presents the behavior of the **Copy**, **Cut**, **Paste** on a function block instance. It only applies to graphical languages as **FBD** and **LD**. • Using the **Paste** function after a **Copy** of an object: As a result of a **Copy**, a new function block instance (**FBI**) is used. The **Paste** function creates a new FBI whenever it is repeated. Accordingly, the FBI is incremented. • Using the **Paste** function after a **Cut** of an object: As a result of a **Cut**, the same instance of the function is used. The **Paste** function uses the same FBI whenever it is repeated. Accordingly, the FBI is identical. **NOTE:** Using same instance of a function block is not recommended because it can lead to incorrect execution of the function blocks. A function block instance uses contextual variables that are owned by each instance usage and cannot be applied to any other. For instance, each current value is specific to one timer instance and one timer usage (Current value of a timer named TON_1 cannot be shared with another timer named TON_2).	

Duplicating an EFB or a DFB instance from the clipboard

Duplicating an EF instance is not possible. If the clipboard contains several objects with different types (EF, EFB and DFB), the EF is copied while the EFB and DFB instances are duplicated.

Duplicating an EFB or a DFB instance from the clipboard:

Using the mouse	Using the keyboard
1. Use the menu command **Duplicate** in the shortcut menu (right mouse button). 2. Move the mouse to the desired target position. **Result:** A dotted frame appears and the mouse pointer changes its symbol to  for a valid position. 3. Left-click the target position.	1. Move the cursor to the target position using the arrow keys. 2. Press the **Ctrl+W** key combination. **Result:** The target position appears as a dotted frame with the symbol  in its center for a valid position. 3. Press **Enter**.

The duplicate operation is also possible across sections between different opened LD or FBD sections in the same or another Control Expert instance.

NOTE: If the EFB or DFB instances to be duplicated do not already exist in the other Control Expert instance, these instances are copied.

Moving objects

Moving objects:

Using the mouse	Using the keyboard
1. Select, page 760 the object to be moved. 2. Position the mouse pointer on the selected object. (on one of the selected objects if several are selected). **Result:** The mouse pointer changes its symbol to  . 3. Click the left mouse button and keep it pressed. **Result:** The mouse pointer changes its symbol to  . 4. Drag the object to the new position and release the mouse button.	1. Position, page 760 the gray field on the object to be moved. 2. Press the **spacebar**. 3. Press the **Shift** key and keep it pressed. 4. Move the object to the target position using the arrow keys.
Result: The selected object is moved from its original position to the target position. The object's actual parameters are moved together with the object. FFBs still have their FFB-links and they are adjusted to the new position of the FFB. Boolean links are only kept for horizontal movement operations. **Note:** A moving of FFBs to cells that are already occupied (e.g. by contact, coil, other FFB, boolean link), is not possible. **Note:** The move operation is also possible across sections between different opened LD sections. **Note:** When objects are moved beyond the visible area of the editor window, automatic scrolling of the window will not begin until you reach the edge of the window with the mouse pointer, not as soon as the objects are moved beyond the edge.	

Copying objects via drag & drop

Copying objects via drag & drop:

Using the mouse	Using the keyboard
1. Select, page 760 the object to be copied. 2. Position the mouse pointer on the selected object. **Result:** The mouse pointer changes its symbol to  . 3. Click the left mouse button and keep it pressed. 4. Press the **Ctrl** key and keep it pressed. **Result:** The mouse pointer changes its symbol to  . 5. Drag the object to the new position. **Note:** This is also possible across sections between different opened LD sections. 6. Release the mouse button. **Result:** A copy of the selected object is pasted at the target position. The object's actual parameters (variable/address) are copied together with the object. A new instance is automatically created for FFBs. FFB links are only copied if they (and their partner objects) are selected explicitly.	-

Copying, and pasting variables

Overview

You can copy / paste the variable of one contact, coil or pin of FBD block to another one.

Copying variables to the clipboard

Copying variables to the clipboard:

Using the mouse	Using the keyboard
1. Select, page 760 the object containing the variable to be copied. 2. Use the **Edit > Copy** menu command. or Use the **Copy** menu command from the shortcut menu (right-click). or Click the  symbol.	1. Select, page 760 the object to be copied. 2. Press the **Ctrl**+**C** key combination.
Result: The variable of the selected object is copied to the clipboard.	

Pasting variables from the clipboard

Pasting variables from the clipboard:

Using the mouse	Using the keyboard
1. Left-click the target object. 2. Use the menu command **Paste variable** in the shortcut menu (right mouse button).	1. Move the gray field to the cell with the target object using the arrow keys. 2. Press the **Ctrl**+**Shift**+**V** key combination. 3. Press **Enter**.
NOTE: If the target position already contains a variable, a confirmation message to replace the existing variable is returned.	

Copying variables via drag & drop

NOTE: This functionality is not available for copying variable of a pin.

Copying variables via drag & drop:

Using the mouse	Using the keyboard
1. Select, page 760 the contact or coil to be copied. 2. Position the mouse pointer on the selected object. **Result:** The mouse pointer changes its symbol to  . 3. Click the left mouse button and keep it pressed. 4. Press the **Alt** key and keep it pressed. **Result:** The mouse pointer changes its symbol to  . 5. Drag the object to the new position. **Note:** This is also possible across sections between different opened LD sections. 6. Release the mouse button. Then release the **Alt** key. **Result:** A copy of the selected object variable is pasted at the target position. If the target position already contains a variable, a confirmation message to replace the existing variable is returned.	-

Insert Mode

Overview

The insert mode allows to drop the following objects on an occupied cell:

- Contact
- Coil
- Operate block
- Compare block

Activating Insert Mode

Insert mode is activated using the Ins button.

The background color of the button in blue indicates that insert mode is active.

The object cursor symbol indicates that select mode is active for this object.

Inserting an Object

Dropping an object on an occupied cell makes an insert of the object if the following conditions are fulfilled:

- There is no derivation on the right side
- There is no EF, EFB, DFB on the right side.
- In addition to the preceding conditions:

for a contact or a coil:	At least one empty cell or one horizontal link is available on the right side.
for an operate block:	At least two empty cell or two horizontal links are available on the right side.
for a compare bloc:	At least four empty cell or four horizontal links are available on the right side.

NOTE: If insert is not possible, a confirmation message of replacement opens.

Using Mnemonics

Introduction

Instead of inserting LD objects via toolbar, menu command or drag-and-drop, you can use so called **Mnemonics**.

Mnemonics Option

Open the **Tools > Options...** dialog.

Under **Data and Languages > Languages > LD > Mnemonics**, the **Property label** column lists the frequently used objects available for LD sections.

In the **Property value** column you can enter a mnemonic for each listed element.

A mnemonic can consist of up to 4 alphanumeric signs.

Using Mnemonics

As you start typing text while the current cell marker is on an empty cell in the LD Editor, a small edit box opens in the editor.

Entering a mnemonic (first key typed is taken over into the edit box) and pressing the **Enter** key insert the corresponding LD objects at the current cell.

You can also insert EF, EFB, and DFB by typing the complete instance name (for example MOVE or ADD) and pressing the **Enter** key.

If the cells needed to insert the element are already occupied, a message box informs you and the element is not inserted.

Displaying the properties

Displaying the object properties

You have the following options to display the object properties dialog box:

- Double-click the object.
- Select, page 760 the object and use the **Edit > Properties... dialog box.**.
- Select, page 760 the object and use the **Properties...command**from the shortcut menu.
- Select, page 760 the object and then press the **Alt**+**Enter** key combination.

When the properties dialog box is called without an object being selected, the section properties dialog box , page 505 is opened.

If several objects are selected, the commands do not work.

View data properties

You have the following options to display the section properties dialog box, page 1202:

- using the data editor, page 348

1. Selecting one or several lines in a list editor
2. Using the shortcut menu select the **Properties**command.

- via the LD section

1. Selecting one or several element(s) in the LD section.
2. From the context menu select the command **Data properties** or press **Ctrl** + **Enter**.

Refining DFBs and subroutines

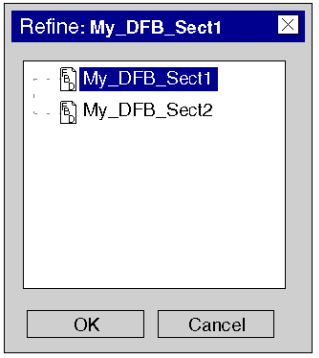
Introduction

This function makes it possible to "look into" a DFB or subroutine. This makes it possible to see the internal state of the DFB/subroutine when animation is used, or to make changes to the logic very quickly.

NOTE: The only limitation occurs when the displayed variable is in an ARRAY with at least 2 dimensions and when the index is not a constant. In this case an error message is displayed and the variable value can't be displayed.

Refining DFBs

Process the following steps to refine DFBs:

Step	Action
1	Select the DFB.
2	You have the following options to refine DFBs: • Use the **Services > Refine** menu command. • Use the **Refine** menu command from the shortcut menu. or • Press the **Ctrl+Q** key combination. **Result:** If the DFB contains only one section, this will be opened automatically. If the DFB contains several sections, a selection dialog is opened. Example: 
3	Select the desired DFB section. **Result:** The DFB section is opened. **Note:** With nested DFBs, several **refine** processes are possible, but do not open simultaneously the refine editors or it could lead to an incorrect data display.

Refining subroutines

Process the following steps to refine subroutines:

Step	Action
1	Select the subroutine block, page 709.
2	You have the following options to to refine subroutines: • Use the **Services > Refine** menu command. • Use the **Refine** menu command from the shortcut menu. or • Press the **Ctrl**+**Q** key combination. **Result:** The subroutine section is opened. **Note:** With nested subroutines several **refine** processes are possible.

Inserting, Deleting, and Copying Rows

Inserting Rows

The following rules apply when inserting rows:

- The same number of rows are inserted as are selected in the vertical ruler.
- The new rows are inserted above the first selected row.
- The first selected row may not contain any objects.

Step	Action
1	Select, page 761 the desired number of rows in the vertical ruler (click the row numbers).
2	Use the **Insert rows** menu command from the shortcut menu of the vertical ruler or use the **Edit > Insert row** menu command. **Result:** The selected number of rows is inserted above the first selected row.

Deleting Rows

The following rules apply when deleting rows:

- The rows to be deleted may not contain any objects.

Step	Action
1	Select, page 762 the desired number of rows to be deleted from the vertical ruler (click the row numbers).
2	Use the **Delete rows** menu command from the shortcut menu of the vertical ruler or use the **Edit > Delete row** menu command. **Result:** The selected rows are deleted.

Copying Rows

Step	Action
1	Select, page 761 the desired rows to copy in the vertical ruler (click the row numbers).
2	Use the **Copy** menu command from the shortcut menu of the vertical ruler or use the **Edit > Copy** menu command.
3	Either: • Select, page 761 the same desired number of rows in the vertical ruler (click the row numbers). Use the **Paste** menu command from the shortcut menu of the vertical ruler or use the **Edit > Paste** menu command. **Result:** The desired selected number of rows is copied. **NOTE:** If the target rows are not empty, the existing objects are overwritten by the new ones. • Use the **Insert Copy** menu command from the shortcut menu of the vertical ruler. **Result:** The desired selected number of rows is inserted above the first selected row.

Columns Management

Default Number of Columns

The default number of columns, when LD section is created, is specified in the **Tools > Project Settings** dialog box, on the **Program > Languages > LD** tab, in the **Number of columns** text box.

Inserting columns

The following rules apply when inserting columns:

- The same number of columns are inserted as are selected in the horizontal ruler.
- The new columns are inserted before the first selected column.

Step	Action
1	Select, page 761 the desired number of columns in the horizontal ruler (click the column numbers).
2	Use the **Insert Column** menu command from the shortcut menu of the horizontal ruler. **Result:** The selected number of columns is inserted before the first selected column and connections between links and blocks are kept.

Deleting Columns

Step	Action
1	Select, page 762 the desired number of columns to be deleted from the horizontal ruler (click the column numbers).
2	Use the **Delete Column** menu command from the shortcut menu of the horizontal ruler. **Result:** The selected columns are deleted.

Column Width

The column width can be managed using the following LD editor toolbar buttons |S| |M| |L|:

|S|	Small size
|M|	Medium size that is double small size
|L|	Large size that is triple small size

Editing Contacts

Overview

This section describes how to edit contacts in the LD programming language.

Select contacts

Introduction

A contact is an LD element that transfers a status on the horizontal link to its right side. This status comes from the Boolean AND link of the status of the horizontal link on the left side with the status of the relevant Boolean current parameter.

Selecting contacts

Selecting contacts:

Contact type	Description	Call via the menu command	Call via the icon	Call via a key	Cursor symbol
N.O. contact	For normally open contacts, the status of the left link is copied to the right link, if the status of the relevant Boolean actual parameter is 1. Otherwise, the status of the right link is 0.	**Objects > Normally open contact** or **Edit > New > Normally open contact** or **Contacts > Normally open contact** in the shortcut menu		**F3**	
N.C. contact	For normally closed contacts, the status of the left link is copied to the right link, if the status of the relevant Boolean actual parameter is 0. Otherwise, the status of the right link is 0.	**Objects > Normally closed contact** or **Edit > New > Normally closed contact** or **Contacts > Normally closed contact** in the shortcut menu		**Shift+F3**	

Contact type	Description	Call via the menu command	Call via the icon	Call via a key	Cursor symbol
Positive transition contact	With contacts for detection of positive transitions, the right link for a program cycle is 1 if a transfer of the relevant actual parameter goes from 0 to 1 and the status of the left link is 1 at the same time. Otherwise, the status of the right link is 0. Also see the Edge recognition (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the user guide.	**Objects > Positive transition-sensing contact** or **Edit > New > Positive transition-sensing contact** or **Contacts > Positive transition-sensing contact** in the shortcut menu		**Ctrl+F3**	
Negative transition contact	With contacts for detection of negative transitions, the right link for a program cycle is 1 if a transfer of the relevant actual parameter goes from 1 to 0 and the status of the left link is 1 at the same time. Otherwise, the status of the right link is 0. Also see the Edge recognition (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the user guide.	**Objects > Negative transition-sensing contact** or **Edit > New > Negative transition-sensing contact** or **Contacts > Negative transition-sensing contact** in the shortcut menu		**Ctrl** + **Shift** + **F3**	

Placing contacts

Introduction

Contacts can be placed in any free cell except cells directly on the right power rail.

If a contact is placed in a cell that was previously occupied by a Boolean link or a label, the cell contents are replaced by the new contact.

If a contact is placed in a cell that is already occupied by another contact or coil, a confirmation message is returned to replace the cell contents by the new contact.

If a contact is placed in a cell that is already occupied by another object and can not be replaced (e.g. FFBs), an error message is returned.

NOTE: The above behavior maybe different when Insert Mode, page 768 is activated.

A placed contact automatically creates a connection with its neighboring objects on the left and right if they are of the `Bool` data type and there are no free cells between them.

The following are permitted as actual parameters for contacts:

- Boolean variables
- Boolean literals (`0`, `1`, `FALSE`, `TRUE`)
- Boolean addresses (topological addresses or symbolic addresses)
- ST expressions that return a Boolean result (e.g. `VarA > VarB`)

  ST expressions as formal parameters on contacts are an extension of IEC 61131-3 and must be explicitly enabled via the **Tools > Project Settings** dialog box, in the **Program > Languages > Common** tab by selecting the **Usage of ST expressions (LD/FBD)** checkbox.

Note

In the **Tools > Options** dialog box on the **Data and Languages > Languages** tab, if the **Automatically assign a variable to a new graphical object** checkbox is selected, then the corresponding properties dialog box is automatically opened when an object is placed. The procedures explained here are used when the checkbox is **cleared**.

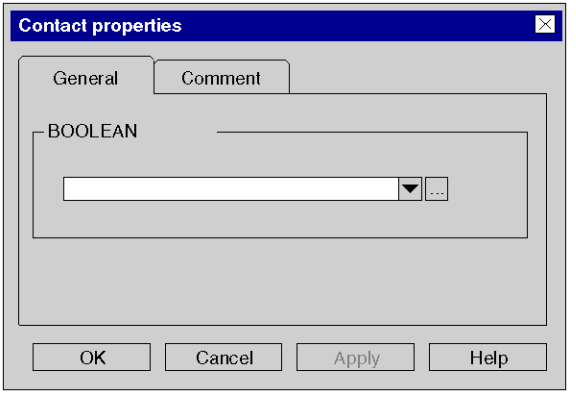
Placing contacts:

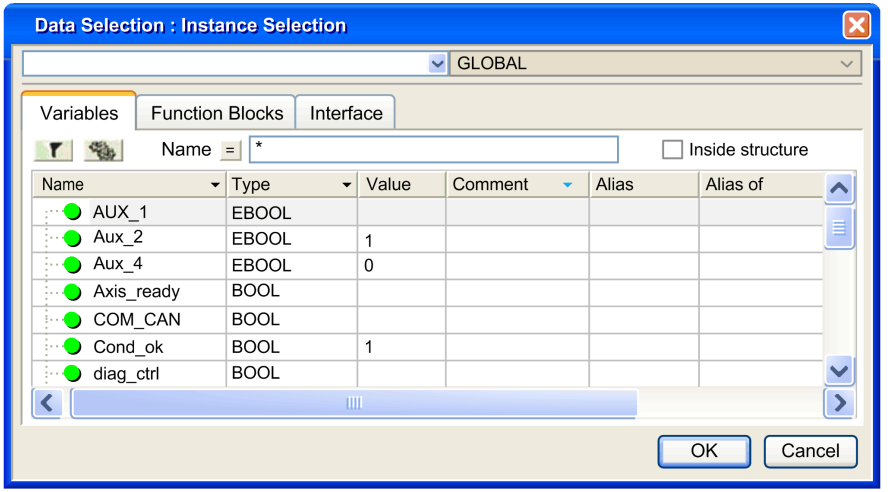
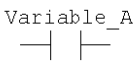
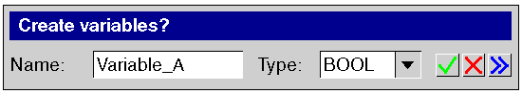
Placing contacts:

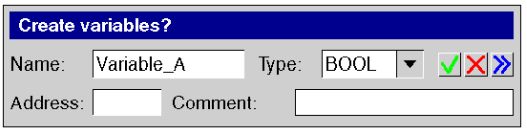
Step	Action
1	Select the desired contact.
2	Click the target cell in the LD section. or Use the **arrow keys** to move the gray field to the target position, and press **Enter**. **Result:** The selected contact is inserted.
3	To place further contacts of the same type: • Click the target cell in the LD section. or Use the **arrow keys** to move the gray field to the target position and press **Enter**. • To return to selection mode: Press the **Esc** key. • To insert other objects: Select the object you wish to insert.

Assigning actual parameters

Assigning actual parameters:

Step	Action
1	Open the properties dialog box, page 770 for the contact. **Result**: The contacts properties dialog box is opened. Example: 
2	You have the following options to enter the actual parameters: • You can now enter the actual parameter and confirm using **Enter**.

Step	Action
	• You can select the actual parameter from the list of recently used names using the ▼ symbol and confirm using **Enter**. or • Using the button **...** you can open an **Instance selection** dialog box and confirm the selected variables there with **OK**. Example of a variable selection dialog box: 
3	Confirm the selected variable with **OK**. **Result:** • If the selected variable is declared, it is enabled in the section. A syntax and semantics check, page 752 is performed and the procedure ends here.  • If the variable is not declared, a variable declaration dialog box opens. In this case, continue with the following steps of this procedure. Variable declaration dialog box  **Note:** If you wish to declare the variable later, close the dialog box using the ✕ symbol. In this case, the variable name is enabled in the section, but not declared.
4	If you do not want to assign the variable an address or comment, confirm using **Enter** or the ✓ symbol. **Result:**The variable is declared and enabled in the section. A syntax and semantics check, page 752 is performed and the procedure ends here.

Step	Action
	Example: Variable_A —| |—
5	If you do want to assign the variable an address and/or comment, use the » symbol for obtaining an advanced dialog and enter the address and/or comment. 
6	Confirm with **Enter** or the ✓ symbol. **Result:** The variable is declared and enabled in the section. A syntax and semantics check, page 752 is performed immediately after the statement is entered. Example: Variable_A —| |—

Usage of ST expressions

To enter ST expressions for contacts the same rules apply as for entering ST expressions for FFB pins., page 826

Property dialog box for contacts

Calling the properties dialog box

see Displaying the properties, page 770

Structure of the properties dialog box

The contact properties dialog box consists of two tabs:

- **General**

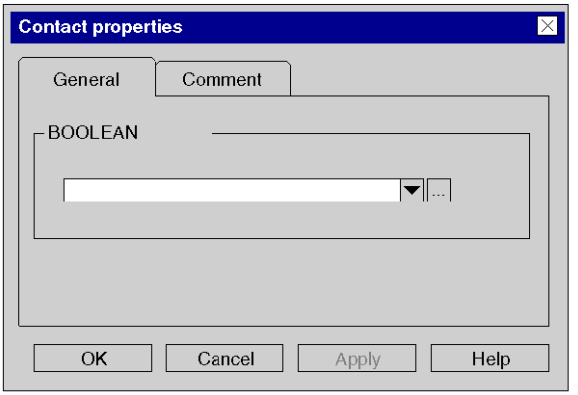
  In this tab, you can enter the actual parameter, page 779 of the contact.

- **Comment**

  In this tab, a comment about the contact can be entered.

General tab

Representation of the **General** tab:

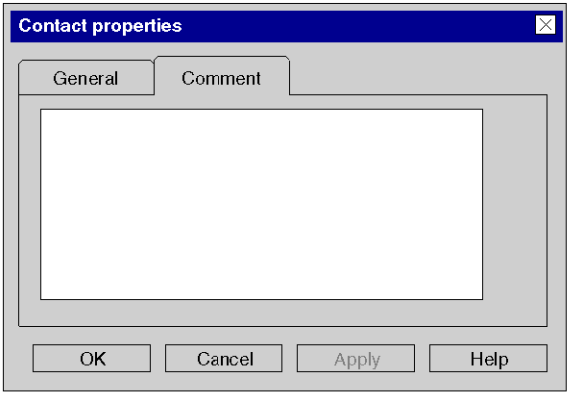


Elements of the **General** tab:

Element	Description
BOOLEAN Expression	Enter the name of the actual parameter in this text box. You have the following options: • You can directly enter the name of the variable/address or paste it from the clipboard. • You can select the name of the variable from the list of recently used names using the ▼ symbol. or • Use the command button **...**to open a variable selection dialog box, page 1181. The actual parameters which are permitted are: • Boolean variables • Boolean constants • Boolean addresses (topological addresses or symbolic addresses) • ST expression delivering a Boolean result (e.g. `BoolVar1 OR BoolVar1`) • Literal (`1` or `0` or. `TRUE` or `FALSE`)
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Comment tab

Representation of the **Comment** tab:



Elements of the **Comment** tab:

Element	Description
Text box	Enter a comment about the contact. This comment is displayed as a tooltip when the cursor is placed over the contact. It is independent from the variable comment that can be assigned to the variable in the data editor.
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Editing Coils

Overview

This section describes how to edit coils in the LD programming language.

Selection of coils

Introduction

A coil is an LD element which transfers the status of the horizontal link on the left side, unchanged, to the horizontal link on the right side. The status is saved in the respective Boolean actual parameter. Coils normally follow contacts or FFBs, (functions and function blocks), but they can also be followed by contacts.

Selecting coils

Selecting coils:

Coil type	Description	Call via menu command	Call via symbol	Call via keyboard	Cursor symbol
Coil	With coils, the status of the left link is copied to the relevant Boolean actual parameter and the right link.	**Objects > Coil** or **Edit > New > Coil** or **Coils > Coil** from the shortcut menu		**F5**	
Negated coils	With negated coils, the status of the left link is copied onto the right link. The inverted status of the left link is copied to the relevant Boolean actual parameter. If the left link is 0, the right link is also 0 and the respective Boolean actual parameter is 1.	**Objects > Negated Coil** or **Edit > New > Negated Coil** or **Coils > Negated Coil** from the shortcut menu		**Shift+F5**	
Positive transition-sensing coil	With coils for detection of positive transitions, the status of the left link is copied to the right link. The relevant Boolean actual parameter is 1 for a program cycle, if a transition from 0 to 1 is made on the left link. Also see the Edge recognition (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the user guide.	**Objects > Positive transition-sensing coil** or **Edit > New > Positive transition-sensing coil** or **Coils > Positive transition-sensing coil** from the shortcut menu		-	

Coil type	Description	Call via menu command	Call via symbol	Call via keyboard	Cursor symbol
Negative transition-sensing coil	With coils for detection of negative transitions, the status of the left link is copied to the right link. The relevant Boolean actual parameter is 1 for a program cycle, if a transition from 1 to 0 is made on the left link. Also see the Edge recognition (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the user guide.	**Objects > Negative transition-sensing coil** or **Edit > New > Negative transition-sensing coil** or **Coils > Negative transition-sensing coil** from the shortcut menu		-	
Set coil	With set coils, the status of the left link is copied onto the right link. The relevant Boolean actual parameter is set to 1 if the left link has the status 1, otherwise it remains unchanged. The relevant Boolean actual parameter can be reset through the reset coil.	**Objects > Set Coil** or **Edit > New > Set Coil** or **Coils > Set Coil** from the shortcut menu		**Alt+F5**	
Reset coil	With reset coils, the status of the left link is copied onto the right link. The relevant Boolean actual parameter is set to 0 if the left link has the status 1, otherwise it remains unchanged. The relevant Boolean actual parameter can be set through the set coil.	**Objects > Reset Coil** or **Edit > New > Reset Coil** or **Coils > Reset Coil** from the shortcut menu		**Shift** + **Alt** + **F5**	

Coil type	Description	Call via menu command	Call via symbol	Call via keyboard	Cursor symbol
Halt coil	With halt coils, if the status of the left link is 1, the program execution is stopped immediately. (With halt coils the status of the left link is not copied to the right link.)	**Objects > Halt coil** or **Edit > New > Halt coil** or **Coils > Halt coil** from the shortcut menu		-	
Call coil	With call coils, the status of the left link is copied to the right link. If the status of the left link is 1, the corresponding subroutine is called. Subroutine calls are expansions to IEC 61131-3 and must be explicitly enabled via the **Tools > Project Settings** dialog box, in the **Program > Languages > Common** tab by selecting the **Allow subroutines** check box.	**Objects > Call Coil** or **Edit > New > Call Coil** or **Coils > Call Coil** from the shortcut menu		**F4**	

Placing coils

Introduction

Coils can be placed in any free cell except cells directly on the left power rail.

If all coils are to be placed on the right power rail, select the **Tools > Project Settings** menu, and in the **Program > LD** tab, check the **Right-justify coils** checkbox.

If a coil is placed in a cell that is already occupied by a Boolean link or a label, the cell contents are replaced by the new coil.

If a coil is placed in a cell that is already occupied by a coil, or a contact, a confirmation message is returned to replace the cell contents by the new coil.

If a coil is placed in a cell that is already occupied by another object and can not be replaced (e.g. FFBs), an error message is returned

NOTE: The above behavior maybe different when Insert Mode, page 768 is activated.

A placed coil automatically creates a connection with its neighboring objects on the left and right if they are of the `BOOL` data type, even if free cells are between them.

The following are permitted as actual parameters for coils:

- Boolean variables
- Boolean addresses (topological addresses or symbolic addresses)

Note

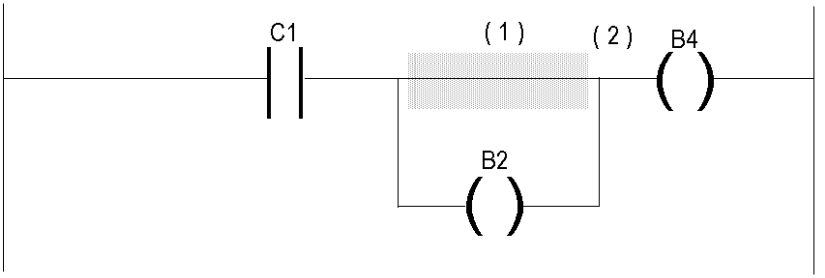
In the **Tools > Options** dialog box on the **Data and Languages > Languages** tab, if the **Automatically assign a variable to a new graphical object** checkbox is selected, then the corresponding properties dialog box is automatically opened when an object is placed. The procedures explained here are used when the checkbox is **cleared**.

Placing Coils

Placing Coils

Step	Action
1	Select the desired coil.
2	Click the target cell in the LD section. or Use the **arrow keys** to move the gray field to the target position, and press **Enter**. **Result:** The selected coil is inserted. In the **Tools > Project Settings** dialog box, on the **Program > LD** tab, if the **Right-justify coils** checkbox is selected and the last cell in the current row (cell directly on the right power rail) is not occupied, the coil is placed automatically on the right power rail. If this cell is already occupied, the coil is inserted into the current cell (cursor position gray field).
3	To place further coils of the same type: • Click the target cell in the LD section. or Use the **arrow keys** to move the gray field to the target position and press **Enter**. • To return to selection mode: Press the **Esc** key. • To insert other objects: Select the object you wish to insert.

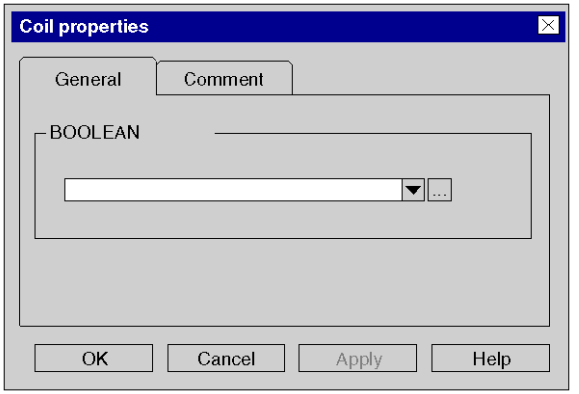
NOTE: When editing a program in Ladder, a short circuit on a coil may lead to a state always true. For a good logic behavior, the short-circuit must be deleted..

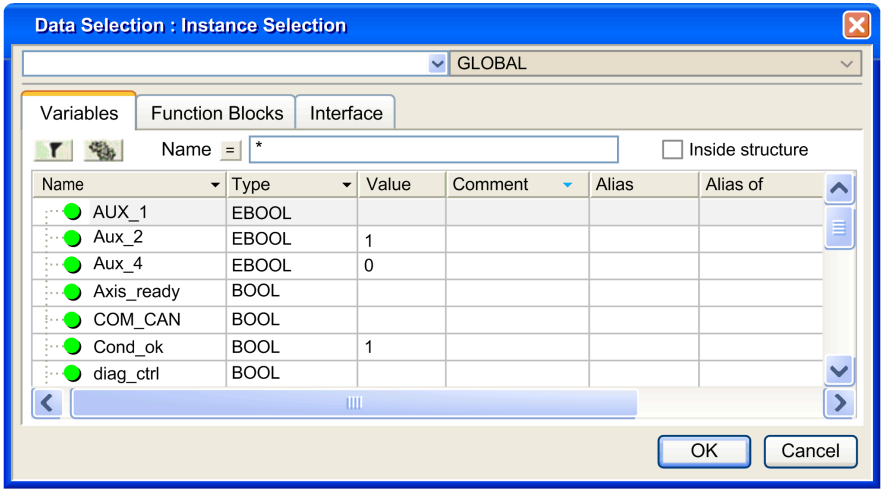
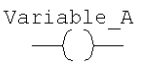
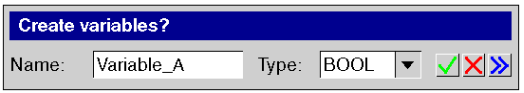


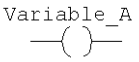
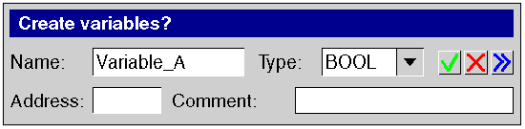
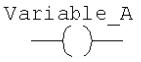
If the short-circuit (1) is located on top of the coil B2, this coil will remain in a true state (B2 = 1) regardless of the logic activating the coil C1. As a consequence, the active state (2) is propagated to the following logic (b4 = 1).

Assigning actual parameters

Assigning actual parameters:

Step	Action
1	Open the properties dialog box, page 770 for the coil. **Result**: The coils properties dialog box is opened. Example: 
2	You have the following options to enter the actual parameters: • You can now enter the variable/address name and confirm using **Enter**.

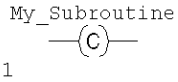
Step	Action
	• You can select the variable name from the list of recently used names using the ▼ symbol and confirm using **Enter**. or • Using the button **...** you can open an **Instance selection** dialog box and confirm the selected variables there with **OK**. Example of a variable selection dialog box: 
3	Confirm the selected variable with **OK**. **Result:** • If the selected variable is declared, it is enabled in the section. A syntax and semantics check, page 752 is performed and the procedure ends here.  • If the variable is not declared, a variable declaration dialog box opens. In this case, continue with the following steps of this procedure. Variable declaration dialog box  **Note:** If you wish to declare the variable later, close the dialog box using the ✕ symbol. In this case, the variable name is enabled in the section, but not declared.
4	If you do not want to assign the variable an address or comment, confirm using **Enter** or the ✓ symbol. **Result:**The variable is declared and enabled in the section. A syntax and semantics check, page 752 is performed and the procedure ends here.

Step	Action
	Example: 
5	If you do want to assign the variable an address and/or comment, use the  symbol for obtaining an advanced dialog and enter the address and/or comment. 
6	Confirm with **Enter** or the  symbol. **Result:**The variable is declared and enabled in the section. A syntax and semantics check, page 752 is performed. Example: 

Calling a subroutine

Calling a subroutine:

Step	Action
1	Place a call coil in the section. **Note:** Subroutine calls are expansions to IEC 61131-3 and must be explicitly enabled via the **Tools > Project Settings** dialog box, in the **Program > Languages > Common** tab by selecting the **Allow subroutines** checkbox.
2	Open the properties dialog box, page 770 for the coil. **Result**: The coils properties dialog box is opened.

Step	Action
3	Click the ▾ symbol and select the subroutine to call from the list and confirm using **Enter**. **Note:** The subroutine to be called must be located in the same task as the LD section called. **Result:** The subroutine call is inserted into the section and a syntax and semantics check, page 752 is performed. Example:  If the subroutine has already been created, the procedure ends here. If the subroutine has not yet been created, continue with step 4.
4	Create the subroutine.

Property dialog box for coils

Calling the properties dialog box

see Displaying the properties, page 770

Structure of the properties dialog box

The properties dialog box for **coils**, **negated coils**, **coils for sensing positive/negative transitions** and **set/reset coils** consists of two tabs:

- **General**

  In this tab, you can enter the actual parameter, page 788 of the coil.
- **Comment**

  In this tab, a comment about the coil can be entered.

The properties dialog box for **halt coils** consists of one tab:

- **Comment**

  In this tab, a comment about the coil can be entered.

The properties dialog box for **call coils** consists of two tabs:

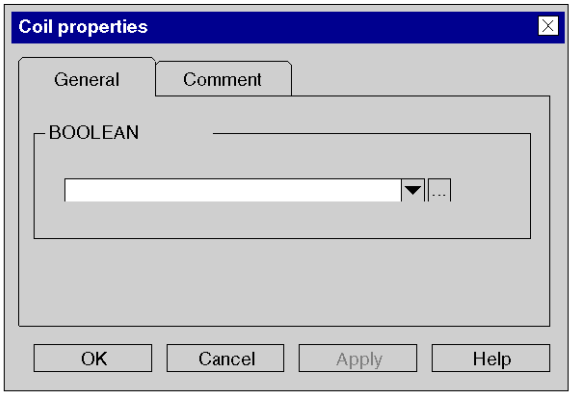
- **General**

  In this tab, you can enter the name of the subroutine to be called, page 790.

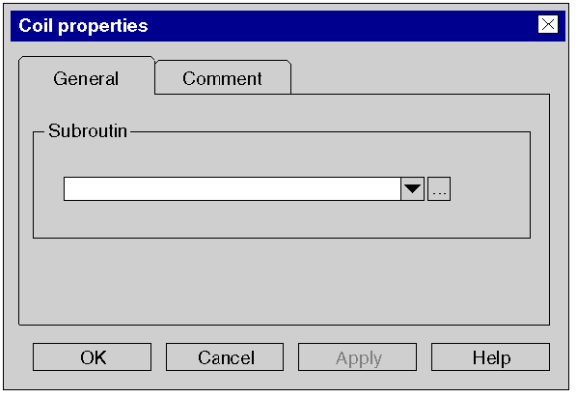
- **Comment**
  In this tab, a comment about the coil/subroutine call can be entered.

General tab

Representation of the **General** tab:



Representation of the **General** tab for call coils:

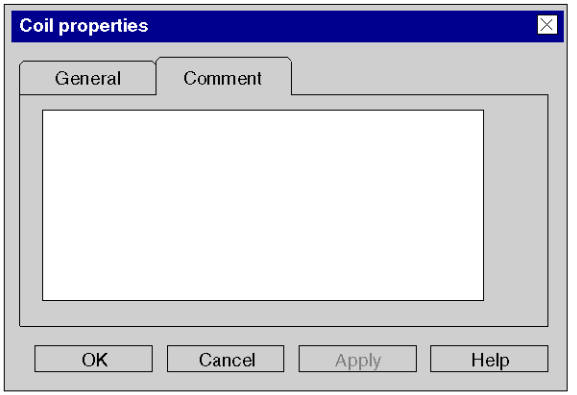


Elements of the **General** tab:

Element	Description
BOOLEAN Expression	This text box is only available for **coils**, **negated coils**, **coils for sensing positive/ negative transitions** and **set/reset coils**. Enter the name of the actual parameter in this text box. You have the following options: • You can directly enter the name of the variable/address or paste it from the clipboard. • You can select the name of the variable from the list of recently used names using the ▼ symbol. or • Use the command button **...**to open a variable selection dialog box, page 1181. The actual parameters which are permitted are: • Boolean variables • Boolean addresses (topological addresses or symbolic addresses)
Subroutine	This text box is only available for **call coils**. You can enter the name of the subroutine to be called in this text box. You have the following options: • You can directly enter the name of the subroutine or paste it from the clipboard. or • You can select the name of the subroutine from the list of subroutines using the ▼ symbol.
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Comment tab

Representation of the **Comment** tab:



Elements of the **Comment** tab:

Element	Description
Text box	Enter a comment about the coil. This comment is displayed as a tooltip when the cursor is placed over the coil. It is independent from the variable comment that can be assigned to the variable in the data editor.
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Editing compare blocks

Overview

This section describes how to edit compare blocks in the LD programming language.

Placing compare blocks

Introduction

Compare blocks are only available in the LD programming language. They are used to execute a compare expression (<, >, <=, >=, =, <>) in the ST programming language.

NOTE: The same functionality is also possible using ST expressions, page 826.

If the state of the left link is 1 and the result of the comparison is 1, the state of the right link is 1.

Compare blocks can be placed in any free cell except the last cell directly on the right power rail.

A compare block requires 1 line and 2 columns.

If a compare block is placed in a cell that is occupied by a Boolean link, and the following cell is either empty or also occupied by a Boolean link, then these cells will be occupied by the placed compare block.

If a compare block is placed in a cell, and this cell or the following cell is already occupied by another object, an error message is returned.

NOTE: The above behavior maybe different when Insert Mode, page 768 is activated.

A placed compare block automatically creates a connection with its neighboring objects on the left and right if they are of the `BOOL` data type and there are no free cells between them.

Activating compare block placement mode

You have the following options to activate the compare block placement mode:

- Use the **Objects > Compare Block** menu command.
- Use the **Edit > New > Compare Block** menu command.
- Use the **Common Objects > Compare Block** menu command from the shortcut menu.
- Press the **Ctrl**+**F7** key combination.

  or

- Choose the 

symbol.

The active compare block placement mode is indicated by the 

cursor symbol.

Note

In the **Tools > Options** dialog box on the **Data and Languages > Languages** tab, if the **Automatically assign a variable to a new graphical object** checkbox is selected, then the corresponding properties dialog box is automatically opened when an object is placed. The procedures explained here are used when the checkbox is **cleared**.

Placing a compare block

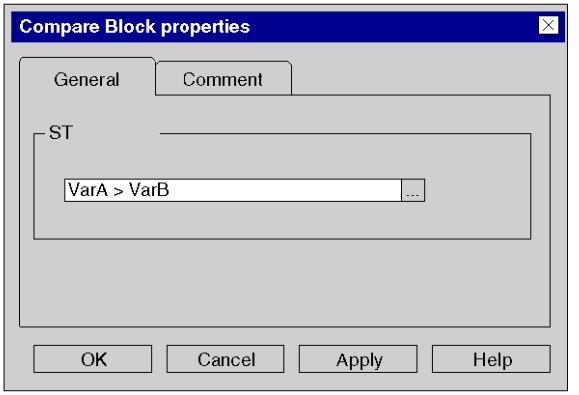
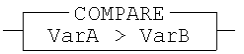
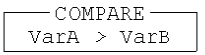
Placing a compare block:

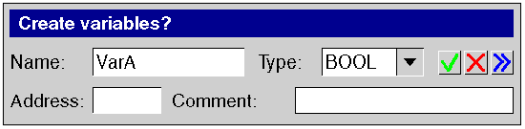
Step	Action
1	Activate the compare block placement mode.
2	Click the target cell in the LD section. or Use the **arrow keys** to move the gray field to the target position, and press **Enter**. **Result:**The compare blocks is placed.
3	Placing other compare blocks: • Click the target cell in the LD section. or Use the **arrow keys** to move the gray field to the target position and press **Enter**. • To return to selection mode: Press the **Esc** key. • To insert other objects: Select the object you wish to insert.

Defining the compare operation

Defining the compare operation:

Step	Action
1	Open the properties dialog box, page 770 for the compare block. **Result**: The properties dialog box for the compare block is opened.
2	Enter the desired compare operation in the text box, see also **General** tab, page 799.

Step	Action
	Example: 
3	Confirm the compare operation with **OK**. **Result:** • If the selected variables are declared, they are enabled in the section. A syntax and semantics check, page 752 is performed and the procedure ends here.  • If the more than one of the selected variables are not declared, they are enabled in the section but not declared. A syntax and semantics check, page 752 is performed. Declare them at this time, using the data editor, page 390. • If the **one** of the selected variables is not declared, a variable declaration dialog box opens. In this case, continue with the following steps of this procedure. Variable declaration dialog box  **Note:** If you wish to declare the variable later, close the dialog box using the symbol. In this case, the variable name is enabled in the section, but not declared.
4	If you do not want to assign the variable an address or comment, confirm using **Enter** or the symbol. **Result:** The variable is declared and enabled in the section. A syntax and semantics check, page 752 is performed and the procedure ends here. Example: 

Step	Action
5	If you do want to assign the variable an address and/or comment, use the » symbol for obtaining an advanced dialog and enter the address and/or comment. 
6	Confirm with **Enter** or the ✓ symbol. **Result:** The variable is declared and enabled in the section. A syntax and semantics check, page 752 is performed. Example: 

Compare block properties dialog box

Calling the properties dialog box

see Displaying the properties, page 770

Structure of the properties dialog box

The properties dialog box consists of two tabs:

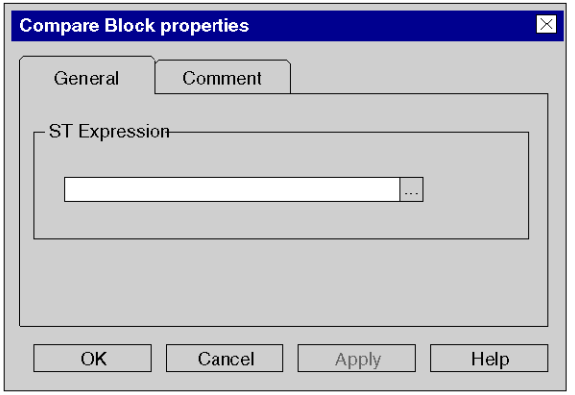
- **General**

  In this tab, you can enter the compare operation, page 796.
- **Comment**

  In this tab, a comment about the compare block can be entered.

General tab

Representation of the **General** tab:

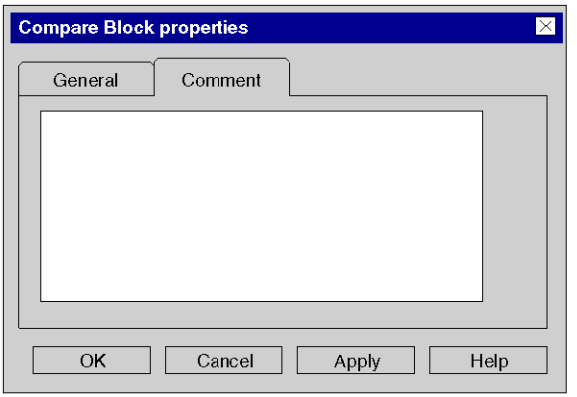


Elements of the **General** tab:

Element	Description
ST Expression	Enter the compare operation in this text box, see also Placing compare blocks, page 795. You have the following options to variables for the comparison operation: • You can enter the name of the variable/address directly or copy it from the clipboard. • Using the command button **...**to open a variable selection dialog box, page 1181. A compare block can contain up to 4,096 characters. If not all characters can be displayed, the beginning of the string sequence is followed by an ellipsis (...).
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Comment tab

Representation of the**Comment**tab:



Elements of the **Comment** tab:

Element	Description
Text box	Enter a comment about the compare block.
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Editing operation blocks

Overview

This section describes how to edit operate blocks in the LD programming language.

Placing Operate Blocks

Introduction

Operate blocks are only available in the LD programming language. They are used to execute an ST statement.

When the status of the left link is 1, the ST statement in the block is executed.

All ST statements are allowed **except** the control statements:

- (`RETURN`,
- `JUMP`,
- `IF`,
- `CASE`,
- `FOR`
- etc.)

For operate blocks, the state of the left link is passed to the right link (regardless of the result of the ST statement).

Operate blocks can be placed in any free cell.

A operate block requires 1 line and 4 columns.

If an operate block is placed in a cell that is occupied by a Boolean link, and the following cells are either empty or also occupied by Boolean links, then these cells will be occupied by the placed operate block.

If an operate block is placed in a cell, and this cell or the following 3 cells are already occupied by another object, an error message is returned.

NOTE: The above behavior maybe different when Insert Mode, page 768 is activated.

A placed operate block automatically creates a connection with its neighboring objects on the left if they are of the `BOOL` data type and there are no free cells between them.

Activating operate block placement mode

You have the following options to activate the operate block placement mode:

- Use the **Objects > Operate Block** menu command.
- Use the **Edit > New > Operate Block** menu command.
- Use the **Common Objects > Operate Block** menu command from the shortcut menu.
- Press the key combination **Alt**+**F7**.

  or

- Choose the 

symbol.

The active operate block placement mode is indicated by the 

cursor symbol.

Note

In the **Tools > Options** dialog box on the **Data and Languages > Languages** tab, if the **Automatically assign a variable to a new graphical object** checkbox is selected, then the corresponding properties dialog box is automatically opened when an object is placed. The procedures explained here are used when the checkbox is **cleared**.

Placing an operate block

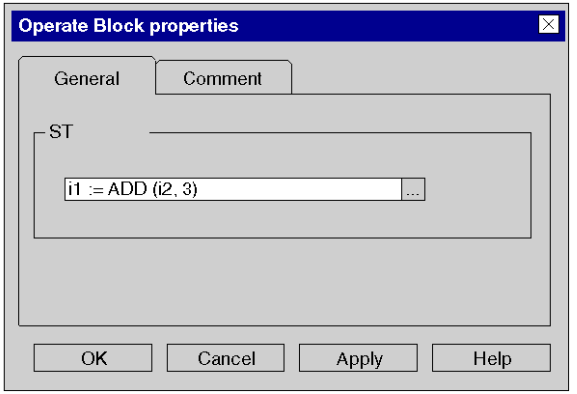
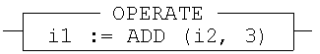
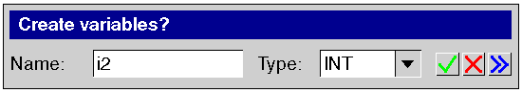
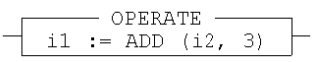
Placing an operate block:

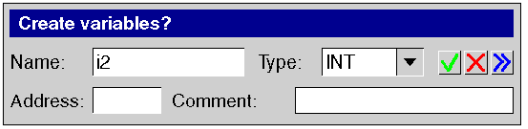
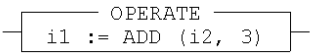
Step	Action
1	Activate operate block placement mode.
2	Click the target cell in the LD section. or Use the **arrow keys** to move the gray field to the target position and press **Enter**. **Result:** The operation block is placed.
3	Placing other operate blocks: • Click the target cell in the LD section. or Use the **arrow keys** to move the gray field to the target position and press **Enter**. • To return to selection mode: Press the **Esc** key. • To insert other objects: Select the object you wish to insert.

Defining the operation

Defining the operation:

Step	Action
1	Open the properties dialog box, page 770 for the operate block. **Result**: The operate block properties dialog box is opened.
2	Enter the desired ST statement in the text box.

Step	Action
	Example: 
3	Confirm the ST statement with **OK**. **Result:** • If the selected variables are declared, they are enabled in the section. A syntax and semantics check, page 752 is performed and the procedure ends here.  • If the more than one of the selected variables are not declared, they are enabled in the section but not declared. A syntax and semantics check, page 752 is performed. Declare them at this time, using the data editor, page 390. • If the **one** of the selected variables is not declared, a variable declaration dialog box opens. In this case, continue with the following steps of this procedure. Variable declaration dialog box  **Note:** If you wish to declare the variable later, close the dialog box using the symbol. In this case, the variable name is enabled in the section, but not declared.
4	If you do not want to assign the variable an address or comment, confirm using **Enter** or the symbol. **Result:** The variable is declared and the ST statement is enabled in the section. A syntax and semantics check, page 752 is performed and the procedure ends here. Example: 

Step	Action
5	If you do want to assign the variable an address and/or comment, use the  symbol for obtaining an advanced dialog and enter the address and/or comment. 
6	Confirm with **Enter** or the  symbol. **Result:** The variable is declared and the ST statement is enabled in the section. A syntax and semantics check, page 752 is performed. Example: 

Operate block properties dialog box

Calling the properties dialog box

see Displaying the properties, page 770

Structure of the properties dialog box

The properties dialog box consists of two tabs:

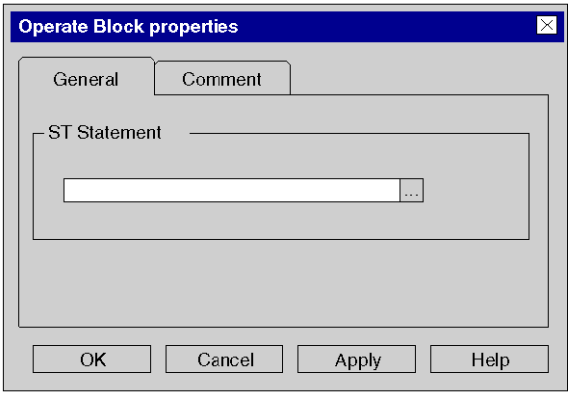
- **General**

  Enter the operation, page 802 in this tab.
- **Comment**

  In this tab, a comment about the operate block can be entered.

General tab

Representation of the **General** tab:

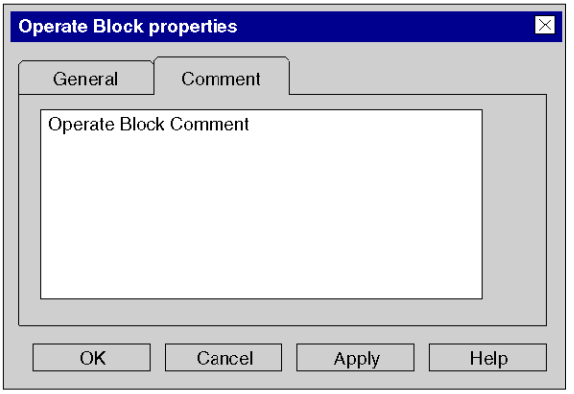


Elements of the **General** tab:

Element	Description
ST Statement	Enter the operation in this text box, see also Placing Operate Blocks, page 800. You have the following options to select variables for the operation: • You can enter the name of the variable/address directly or copy it from the clipboard. • Using the command button **...**to open a variable selection dialog box, page 1181. An operate block can contain up to 4,096 characters. If not all characters can be displayed, the beginning of the string sequence is followed by an ellipsis (...).
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Comment tab

Representation of the **Comment**tab:



Elements of the **Comment** tab:

Element	Description
Text box	Enter a comment about the operate block. This comment is displayed above the operate block if the mixed display mode, page 753 is activated.
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Editing FFBs

Overview

This section describes the various possibilities to edit elementary functions, elementary function blocks, derived function blocks and procedures in the LD programming language.

General information on calling an FFB

Introduction

FFB is the generic term for:

- Elementary Function (EF)
- Elementary Function Block (EFB)
- Derived Function Block (DFB)
- Procedure

  (Procedures are an expansion to IEC 61131-3 and must be enabled explicitly in the **Tools > Project Settings** dialog box, on the **Languages > Common** tab by selecting the **Allow procedures** checkbox).

FFBs can be placed in any free space.

If an FFB is to be placed in an area that is already occupied by an object (contact, coil, other FFB, link), an error message is returned.

For more information, refer to chapter *Elementary Functions, Elementary Function Blocks, Derived Function Blocks and Procedures (FFBs)* (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).

Declaring block instances

Function blocks (elementary and derived) have internal states. If the inputs have the same values, the value on the output can have another value during the individual executions of the function block. For example, with a counter the value on the output is incremented.

The instance name serves as a unique identification for the function block in a project.

Instances can be declared:

- By selecting a function block type and its placement in a section.
- By the declaration of the block instance in the FFB input assistant, page 1199 and its placement in the section.
- By the declaration of the block instance in the Data editor, page 380.

Calling FFBs

FFBs can be called in the following ways:

- Via the FFB Input Assistant, page 808 menu command
  - Use the **Edit > FFP input assistant...** menu command.
  - Use the menu command **FFB Input Assistant...**in the shortcut menu.

    or
  - Press the **Ctrl+I** key combination.
- Via the data selection, page 812
  - Use the menu command **Edit > Data Selection**.
  - Use the menu command **Data selection...** in the shortcut menu.
  - Press the **Ctrl+D** key combination.

    or
  - Select the  symbol.
- Via the **Types library browser** per drag & drop
  - Use the **Tools > Types Library Manager** menu command.

    or
  - Press the **Alt+3** key combination.

NOTE: Timers and counters EFB can be called directly via the LD Instruction Bar, page 886. You can also customize the LD toolbar to call directly a maximum of 20 EF, EFB or DFB via the LD Favorite Toolbar, page 887.

Calling an FFB via the FFB input assistant

Activating the FFB input assistant

You have the following options to activate the FFB input assistant:

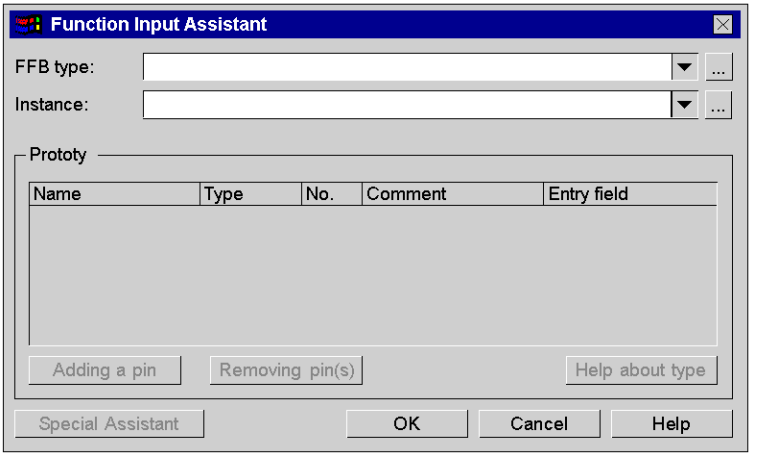
- Use the **Edit > FFB Input Assistant... menu command.** (no object may be selected).
- Use the **menu command** from the shortcut menu (no object may be selected).

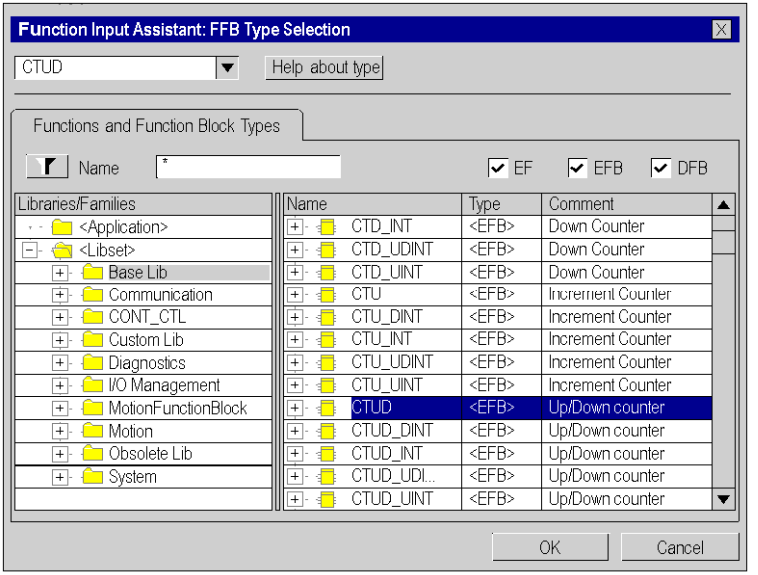
  or
- Press the key combination **Ctrl+I** (no object may be selected).

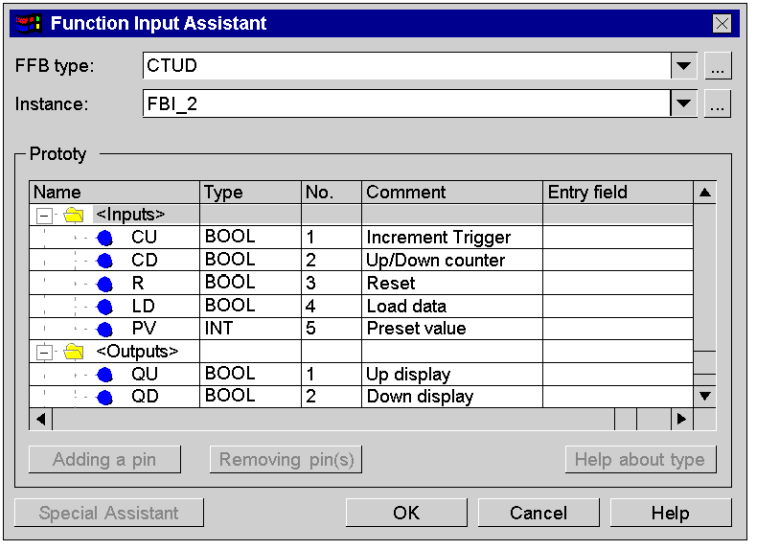
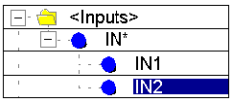
The active FFB placement mode is indicated by the  cursor symbol.

Selecting an FFB type via the Input Assistant

Process the following steps to select an FFB type via the FFB input assistant:

Step	Action
1	From the **FFB input assistant....**from the shortcut menu (right-click) for the FFB input assistant, page 812section. **Result:** The FFB input assistant, page 1199 is opened. 
2	You have the following options to enter the FFB type with the:**FFB Type** text box: • You can enter the FFB type in the textbox: In this case, continue with step 6 of the procedure. • You can select the FFB type from the list of recently used names using the ▼ symbol. In this case, continue with step 6 of the procedure. • Using the button **...** open the FFB selection dialog, page 1190.
3	Select the library from the **Libraries/Families** column which contains the desired function block. Tip: If you do not know where to find the FFB you are looking for select the **Libset**, entry to see a list of all the FFBs available independently of their libraries.
4	Select the desired function block from the **Name** column.

Step	Action
	Example: 
5	Confirm the selection with **OK**. **Result:** The dialog box is closed and FFB input assistant is activated.

Step	Action
	Example: 
6	You have the following options to enter the block instance for function blocks in the **instance** text box: • You can accept the name which is suggested. • You can change the name which is suggested.
7	The number of inputs, page 692can be increased with some elementary functions. Expandable pins can be recognized from the following display:  To add further pins select the last pin in the structure and press **Add pin**.
8	To assign the current parameter double-click the **Entry field** cell of the first formal parameter and enter the parameter to be used. You have the following options to enter the variables/addresses: • You can enter the variable name and confirm using **Enter**. • You can select the variable/address from the list of recently used variables/addresses using the ▼ symbol. or • Using the button **...**to open a variable selection dialog box, page 1181. Assign an actual parameter to all formal parameters of the function block in this way.

Step	Action
9	Confirm the entries using the **OK** button. **Result:** The FFB input assistant is closed and the FFB input mode is activated. The active FFB placement mode is indicated by the cursor symbol.
10	To place the FFB , and click the target cell in the LD section. or Use the **arrow keys** to move the gray field to the target position and press **Enter**. **Result:** The FFB is inserted and a syntax and semantics check, page 752 is performed.

Calling an FFB via the data selection

Set data selection to active

You have the following options to activate the data exchange:

- Use the **Edit > Data Selection menu command**.
- Use the menu command**Data selection...**from the shortcut menu (right-click) for the section.
- Press the **Ctrl**+**D** key combination.

  or

- Select the  symbol.

  NOTE: When executing the commands no object can be selected in the section.

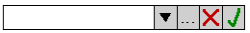
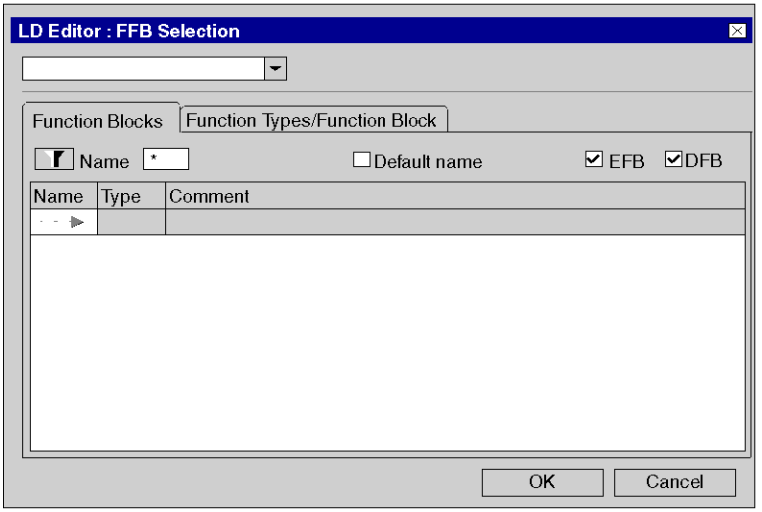
The active FFB placement mode is indicated by the cursor symbol.

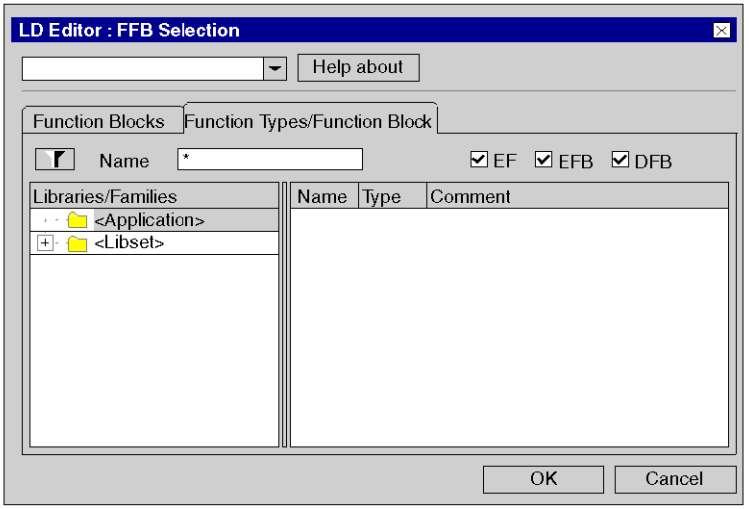
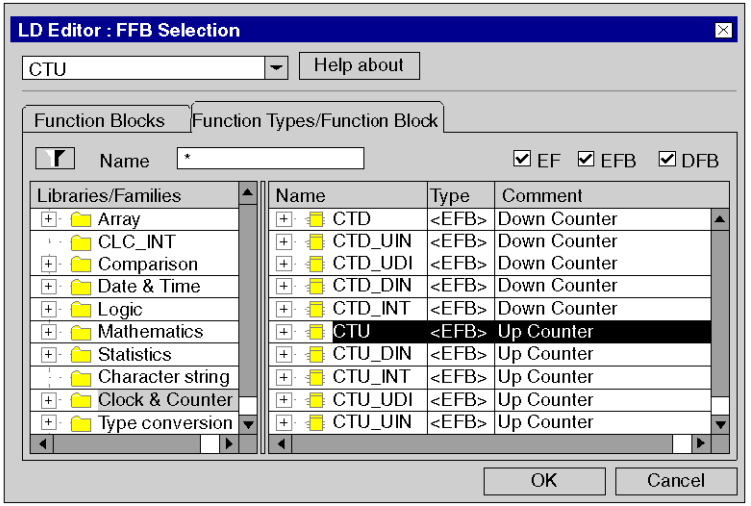
Note

In the **Tools > Options** dialog box on the **Data and Languages** tab, if the **Automatically assign a variable to a new graphical object** checkbox is selected, then the corresponding properties dialog box is automatically opened when an object is placed. The procedures explained here are used when the checkbox is **cleared**.

Selecting an FFB type via the Data Selection

Process the following steps to select an FFB type via the Data selection:

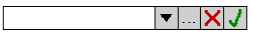
Step	Action
1	From the **Data selection....**from the shortcut menu (right-click) for the Data selection, page 812section. **Result:** A data selection box, page 1850 is opened. 
2	You have the following options to enter the FFB type: • You can now enter the FFB type and confirm using **Enter**or the ✓button. In this case, continue with step 7 of the procedure. • You can select the name from the list of recently used names using the FFB type▼ symbol and confirm using **Enter**or the button ✓. In this case, continue with step 7 of the procedure. • You can delete your entry by pressing **Esc**or clicking the ✗ button. • Using the button **...you can open** the dialog box for FFB selection, page 1186. Example of an FFB type selection dialog box: 
3	Select the **Function Types, Function Block Types** tab. **Result:**

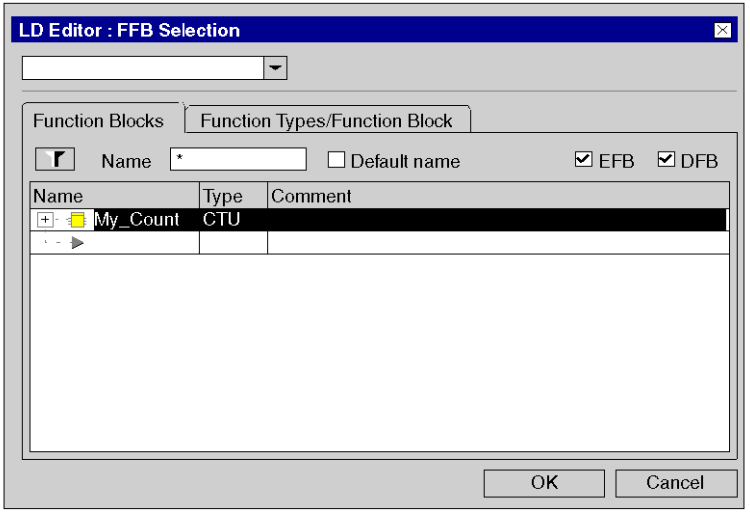
Step	Action
	The tab used for selecting an FFB type, page 1188 is opened. 
4	Select the library and family from the **Libraries/Families** column which contains the desired function block. **Tip:** If you do not know where to find the FFB you are looking for select the **Libset**, entry to see a list of all the FFBs available independently of their libraries.
5	Select the desired function block from the **Name** column. Example: 

Step	Action
6	Confirm the selection with **OK**. **Result:** The dialog box is closed and FFB placement mode is activated.
7	To place the FFB , and click the target cell in the LD section. or Use the **arrow keys** to move the gray field to the target position and press **Enter**. **Result:** For the selected FFB type, an instance name is automatically generated (only valid for function blocks), the FFB is inserted and placement mode remains active so that further FFBs can be inserted. To end insert mode press **Esc**. **Note:** The automatically generated instance names can be changed to improve clarity, also see Managing of instances of data belonging to the family of function blocks (EF), page 380.

Selecting a block instance

Process the following steps to select a block instance that has already been declared:

Step	Action
1	Activate FFB placement mode. **Result:** A data selection box, page 1850 is opened. 
2	Press the button **...**, to open the FFB selection dialog, page 1186.
3	Select the **Function Blocks** tab (if it is not already active). **Result:**

Step	Action
	The tab used for selecting a block instance, page 1187 is opened and the desired selection of block instances is displayed. 
4	Select the desired block instance from the **Name** column.
5	Confirm the selection with **OK**. **Result:** The dialog box is closed and FFB placement mode is activated.
6	Click the target cell in the LD section. or Use the **arrow keys** to move the gray field to the target position and press **Enter**. **Result:** The selected block instance is inserted and the placement mode remains active so that more block instances can be inserted. To end insert mode press **Esc**.

Connection of FFBs to the Left Power Rail

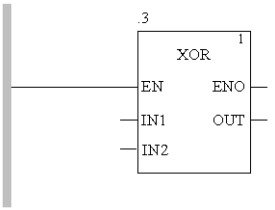
Introduction

FFBs will only be processed when they are directly or indirectly connected to the left bus bar.

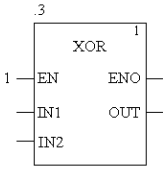
If the FFB should be executed conditionally, the EN input can be pre-linked using contacts or other FFBs, see also section EN and ENO (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the User manual.

Examples

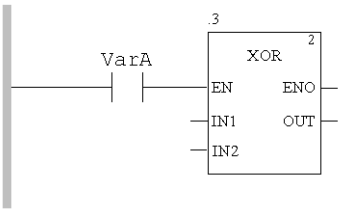
Unconditional FFB call



Unconditional FFB call



Conditional FFB call



Assigning Actual Parameters

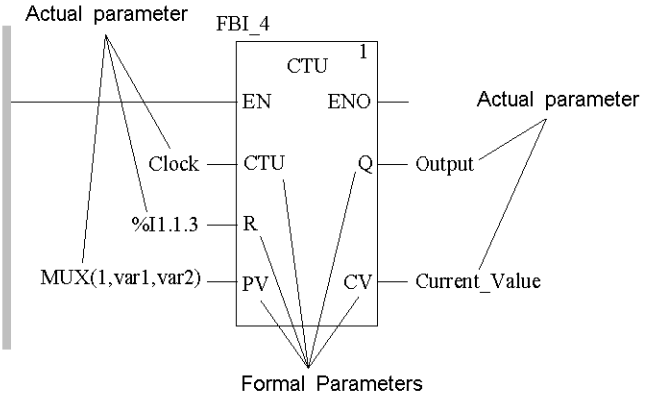
Introduction

Input and output are required to transfer values to or from an FFB. These are called formal parameters.

Objects are linked to formal parameters; these objects contain the current process states. They are called actual parameters.

The data type of the actual parameter must match the data type of the input/output (formal parameter). The only exceptions are generic inputs/outputs whose data type is determined by the actual parameter. If all actual parameters consist of literals, a suitable data type is selected for the function block.

Formal and Actual Parameters:



Permitted actual parameters are:

- Variables
  - Enter already declared variables, page 819
  - Enter undeclared variables, page 821
- Formal parameters of other function blocks, page 823
- Literals, page 825
- ST expressions, page 826

  ST expressions as formal parameters on FBB inputs are an extension of IEC 61131-3 and must be explicitly enabled via the **Tools > Project Settings** dialog box, in the **Language extensions** tab by selecting the **Usage of ST expressions** checkbox.
- Links, page 853 to other LD objects

  NOTE: Please take note of the section Information about Programming (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.

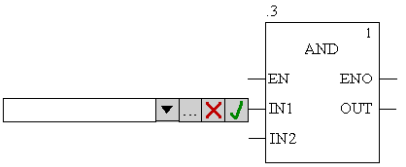
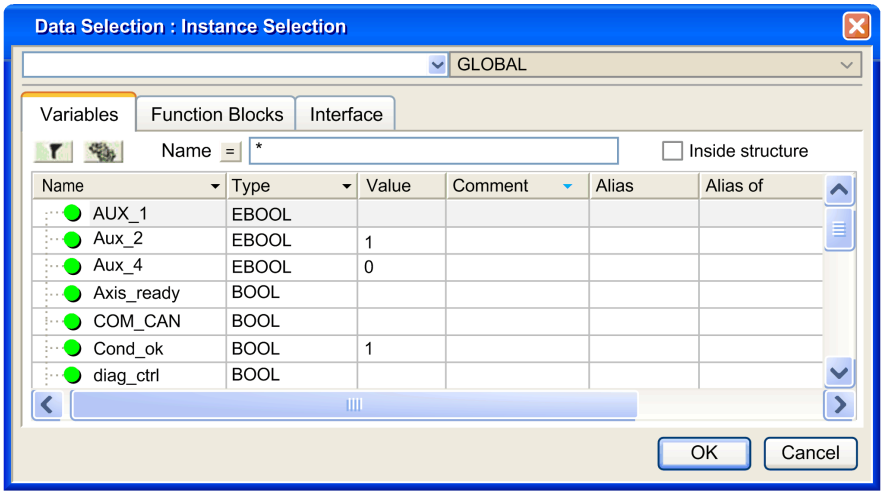
You can make actual parameter assignments for

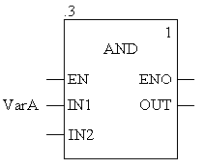
- each pin individually

  or
- with the function input assistant, page 828 for all pins

Assign declared variables to a pin

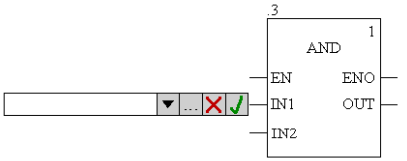
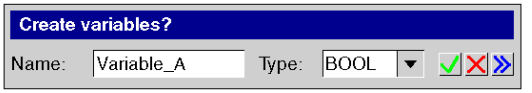
Process the following steps to assign undeclared variables to a pin. (See also Creation of EDT instances, page 390.)

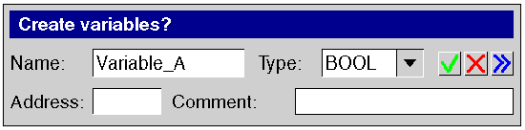
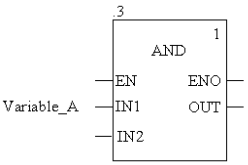
Step	Action
1	Open a data selection box for the desired FFB pin by: • double-clicking the pin. • selecting the pin and clicking the **Edit > Data Selection...** menu command. • selecting the pin and clicking the **Data Selection** menu command on the shortcut menu. • selecting the pin and pressing the **Ctrl+D** key combination. or • selecting the pin and clicking the  symbol. **Result:** The data selection box, page 1850 opens. 
2	You have the following options to enter variables: • You can now enter the variable name and confirm using **Enter**or the ✓button. • You can select the variable names from the list of recently used names using the ▼ symbol and confirm using **Enter** or the ✓button. • You can delete your entry by pressing **Esc** or the ✕ symbol. • Using the button **...** you can open an **Instance selection** dialog box and confirm the selected variables there with **OK**. Example of a variable selection dialog box: 

Step	Action
	Result: The selected variable is inserted and a syntax and semantics check, page 752 is performed. Example: 

Assign undeclared variables to a pin

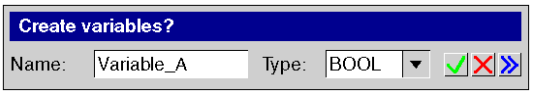
Process the following steps to assign undeclared variables to a pin:

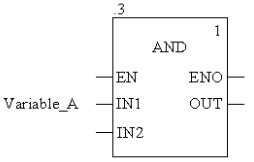
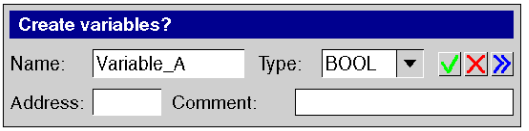
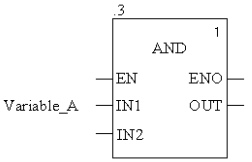
Step	Action
1	Open a data selection box for the desired FFB pin by: • double-clicking the pin. • selecting the pin and clicking the **Edit > Data Selection...** menu command. • selecting the pin and clicking the **Data Selection** menu command on the shortcut menu. • selecting the pin and pressing the **Ctrl+D** key combination. or • selecting the pin and clicking the  symbol. **Result:** A data selection box, page 1850 opens. 
2	Enter the variable name and confirm using **Enter**or the  symbol. **Result:** The variable declaration dialog box opens. 

Step	Action
3	Change the default data type, if desired. **Note:** If you wish to declare the variable later, close the dialog box using the symbol. In this case, the variable name is enabled on the pin, but not declared.
4	If you do want to assign the variable an address and/or comment, use the symbol for obtaining an advanced dialog and enter the address and/or comment. 
5	Confirm with **Enter** or the symbol. **Result:** The variable is declared and enabled on the selected pin. 
6	Enter all actual parameters in this way.

Declaring variables on a pin

Process the following steps to declare variables on a pin:

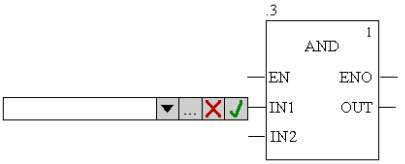
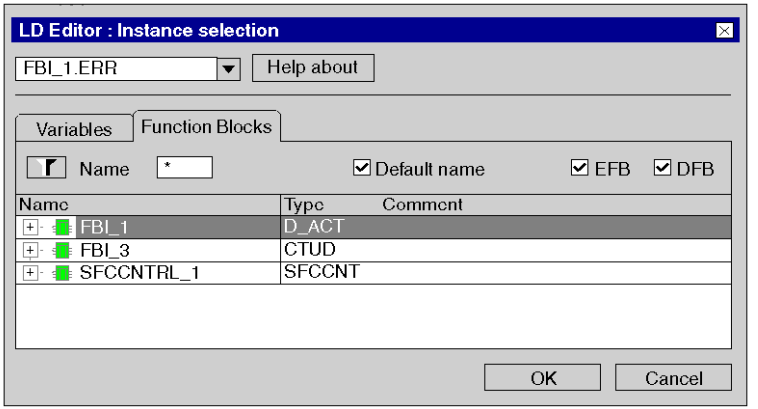
Step	Action
1	Open the variable declaration dialog box for the desired variable by. • Select an undeclared variable (red wavy line under the variable name) and the **Create Variable** command from the shortcut menu. • Select an undeclared variable (red wavy line under the variable name) and the **Shift+Enter** key combination. **Result:** The variable declaration dialog box opens. 
2	If you do not want to assign an address or comment, confirm using **Enter** or the symbol.

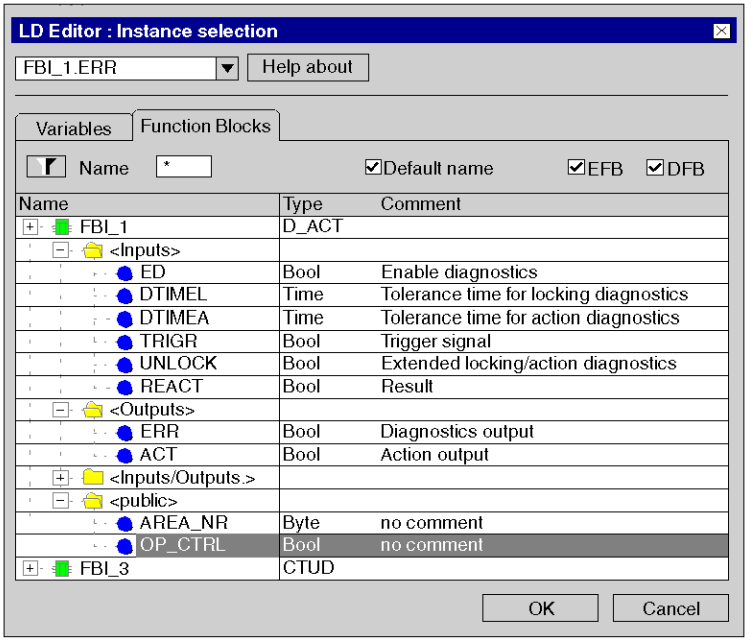
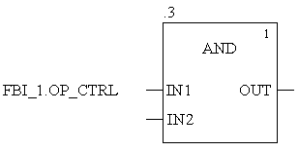
Step	Action
	Result: The variable is declared and the red wavy line under the variable name disappears. 
3	If you do want to assign the variable an address and/or comment, use the  symbol for obtaining an advanced dialog and enter the address and/or comment. 
4	Confirm with **Enter** or the  symbol. **Result:** The variable is declared and the red wavy line under the variable name disappears. 

Assigning formal parameters to a pin

Process the following steps to assign formal parameters to a pin:

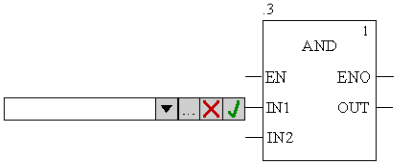
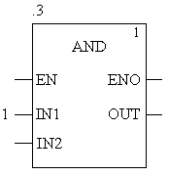
Step	Action
1	Select the desired FFB pin.
2	Open a data selection box by: • double-clicking the pin. • the **Edit > Data Selection** menu command. • the **Data Selection** menu command from the shortcut menu. or • the **Ctrl+D** key combination. **Result:**

Step	Action
	The data selection box, page 1850 opens. 
3	You have the following options to enter the formal parameter: • You can enter the formal parameter name and confirm using **Enter** or the ✓ button. • You can select the formal parameter name from the list of recently used names using the ▼ symbol and confirm using **Enter** or the ✓ button. • You can delete your entry by pressing **Esc** or clicking the ✗ button. • Using the **button...**to open a variable selection dialog box, page 1181.
4	Select the **Function Block** tab. **Result:** The function block instances are displayed. 
5	Expand the display of the inputs, outputs and public variables by clicking their **+** symbols. **Result:**

Step	Action
	You can now see all available inputs, outputs and public variables of the function block. 
6	Select the desired formal parameter and confirm with **OK**.
7	**Result:** The formal parameter selected is inserted and a syntax and semantics check, page 654 is performed. Example: 

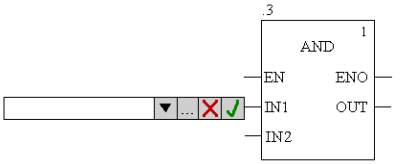
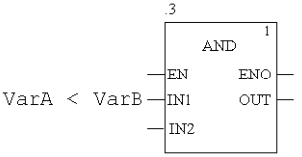
Assigning literals to a pin

Process the following steps to assign literals to a pin:

Step	Action
1	Select the desired FFB pin.
2	Open a data selection box by: • double-clicking the pin. • selecting the pin and clicking the **Edit > Data Selection** menu command. • selecting the pin and clicking the **Data Selection** menu command on the shortcut menu. • selecting the pin and pressing the **Ctrl+D** key combination. or • selecting the pin and clicking the  symbol. **Result:** The data selection box, page 1850 opens. 
3	Enter the literal (e.g. `0`, `1`, `TRUE`, `FALSE`, `3.5`, `t#2ms`) and confirm using **Enter**. **Tips:** If the in the dialog box **Tools > Project settings** in the **Language Extensions** tab **Allow leading digits** checkbox is cleared, the following simplifications for entering literals are possible: • When assigning `TIME` literals to a pin of the `TIME` data type, it suffices to enter the numerical value and the unit (such as `h`, `m`, `s`). The prefix (`t#`) is automatically added. • When assigning `REAL` literals to a pin of the `REAL` data type, for integers it suffices to enter the numerical value. The decimal marker (`t#`) is automatically added. **Result:** The literal is inserted and a syntax and semantics check, page 752 is performed. Example: 

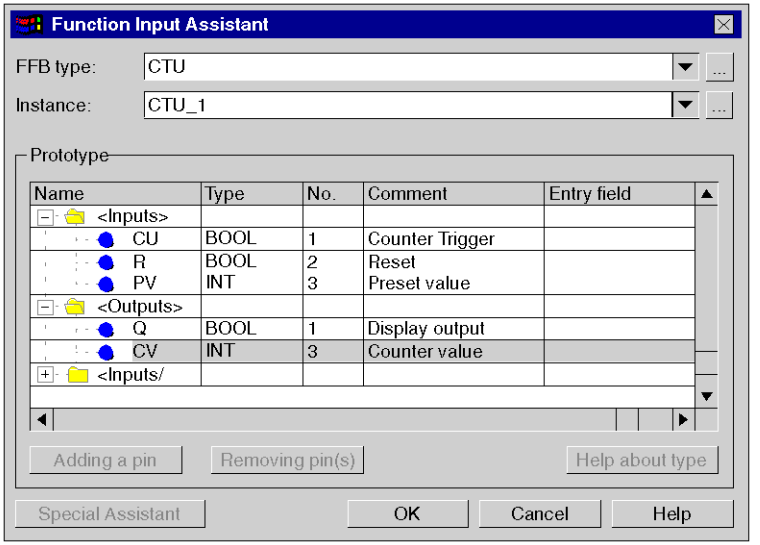
Assigning ST expressions to a pin

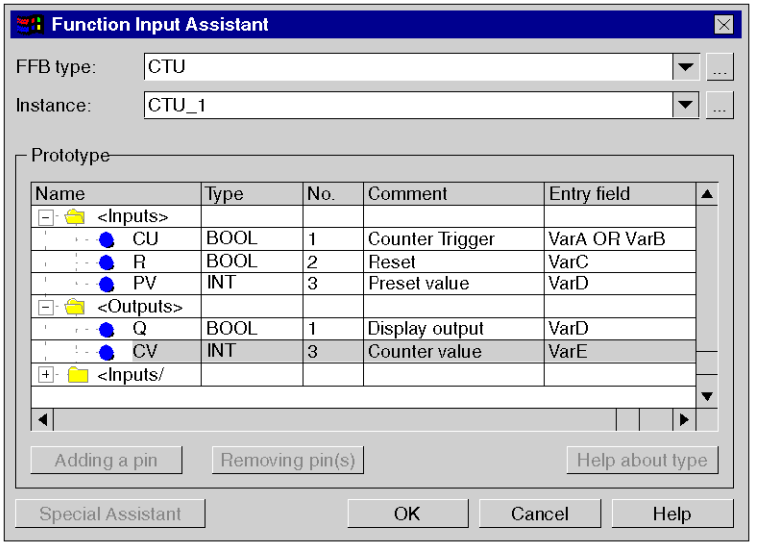
Process the following steps to assign ST expressions to a pin:

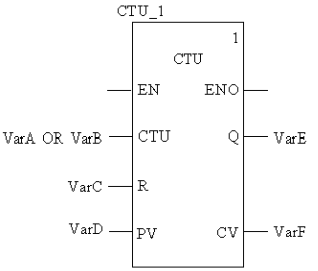
Step	Action
1	Make sure that the **Usage of ST expressions** checkbox is activated in the **Tools > Project Settings** dialog box.
2	Select the desired FFB pin.
3	Open a data selection box by: • double-clicking the pin. • selecting the pin and clicking the **Edit > Data Selection** menu command. • selecting the pin and clicking the **Data Selection** menu command on the shortcut menu. • selecting the pin and pressing the **Ctrl+D** key combination. or • selecting the pin and clicking the  symbol. **Result:** The data selection box, page 1850 opens. 
4	Enter the ST expression [e.g. `MUX(1,var1,var2)`, `VarA * VarB`, `VarA < VarB`] and confirm using **Enter**. **Result:** The ST expression selected is inserted and a syntax and semantics check, page 752 is performed. Example:  If the variables have already been declared (see also Creation of EDT instances, page 390), the procedure ends here. If the variables have not yet been declared, continue with step 5.
5	Declare all variables used, either: • using the data editor, page 390 or • using **Create variable** in the context menu (possible when only one variable of the ST statement is undeclared).

Changing the actual parameter assignment using the function input assistant

Process the following steps to change actual parameter assignment with the function input assistant:

Step	Action
1	Select the FFB.
2	Open the function input assistant, page 1199: • The **Edit > FFB Input Assistant...** menu command. • The **FFB Input Assistant...** menu command from the shortcut menu. or • The **Ctrl+I** key combination. **Result:** The function input assistant will be opened. 

Step	Action
3	Double click the **Entry Field** cell of the first formal parameter and enter the actual parameter to be used. You have the following options to enter the variables/addresses: • You can enter the variable name and confirm using **Enter**. • You can select the variable/address from the list of recently used variables/addresses using the ▼ symbol. or • Using the button **...** to open an **Instance selection** dialog box. Assign an actual parameter to all formal parameters of the function block in this way. Example: 
4	Confirm the entries using the **OK** button. **Result:** The actual parameter is inserted and a syntax and semantics check, page 752 is performed. Example:

Step	Action
	

Using Public Variables

Introduction

In addition to inputs/outputs, some function blocks also provide public variables.

These variables transfer statistical values (values that are not influenced by the process) to the function block. They are used for setting parameters for the function block.

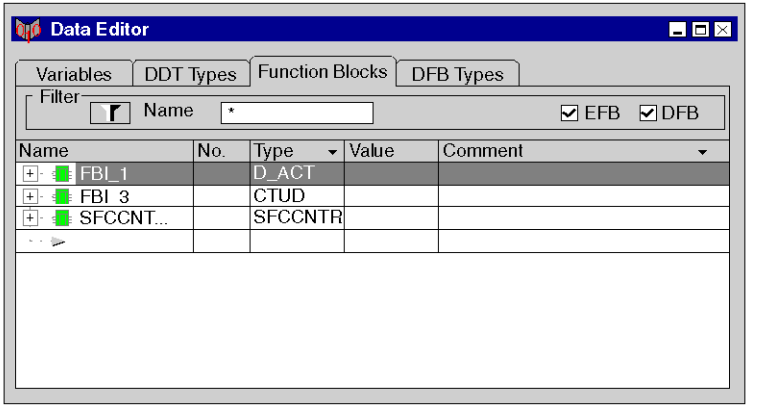
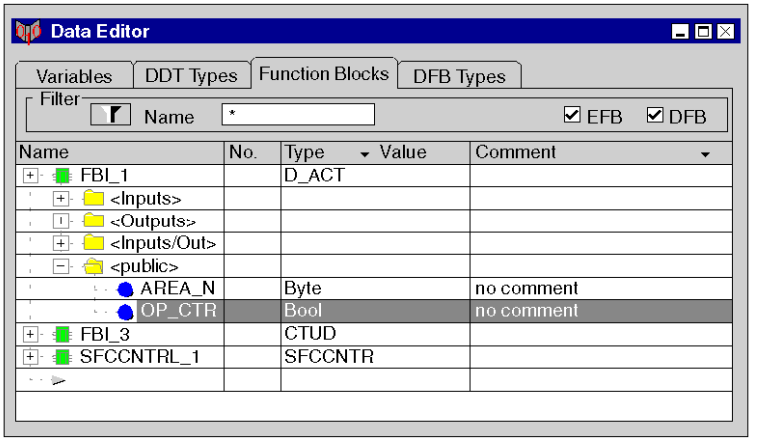
The assignment of values to public variables is made via their initial values.

Public variables are read via the instance name of the function block and the names of the public variables.

Assigning Values to Public Variables

Carry out the following steps to assign a value to a public variable:

Step	Action
1	Place a function block in the section, see also Calling an FFB via the data selection, page 812.
2	Open the data editor, see also Data Editor Access, page 349.
3	Select the **Function Block** tab. **Result:**

Step	Action
	The function block instances are displayed. 
4	Expand the display of the public variables by clicking the respective **+** symbols. **Result:** You can now see all available public variables of the function block. 
5	Enter the desired value for the public variable in the **Value** box.

Reading public variables

See Assigning formal parameters to a pin, page 823

Expanding Functions

Introduction

The number of inputs can be increased with some elementary functions.

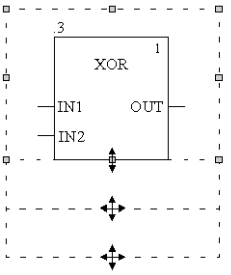
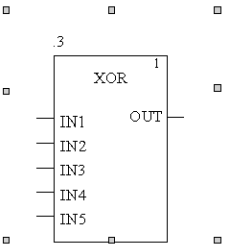
You find out which functions can be expanded by referring to the descriptions for the individual functions.

NOTE: Only expand the function by the number of inputs actually required as non-occupied inputs are assigned a 0 as standard.

Expanding a function

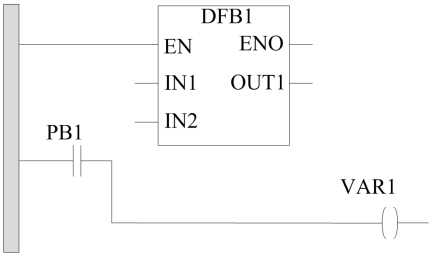
Carry out the following steps to expanding a function:

Step	Action
1	Select the function.
2	Position the mouse pointer on the bottom point to be expanded.

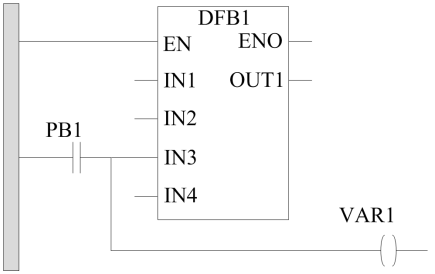
Step	Action
3	Press the left mouse button and drag the function to the required size. Example: 
4	Release the mouse key. **Result:** The desired number of inputs/outputs have been added. Example: 

NOTE: Expanding a DFB can cause an automatic link with a line placed just under the block.

The following figure shows an example of LD program before a DFB expansion:



The following figure shows the program modification that occurs after a DFB expansion:



Extending the DFB1 will automatically link PB1 to the new input IN3.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

Do not extend an FFB over another one before adding manually the necessary line.

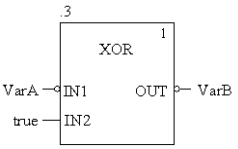
Failure to follow these instructions can result in death, serious injury, or equipment damage.

Negating FFB Pins

Introduction

FFB pins (inputs and outputs) of the data type BOOL can be negated.

Negated pins are identified with a circle symbol:



NOTE: Negated pins must be connected with a graphical link or a variable. Open negated pins are not allowed.

Negating an FFB pin

FFB pins can be negated in the following ways:

- Select the pin to be negated and execute the menu command **Negate Pin** in the shortcut menu (right-click).

  or

- Launch the Inversion tool by clicking on the  icon, then click on the pin to be negated.

EN and ENO Show and Hide

Introduction

With all FFBs, one EN input and one ENO output can be configured, see also EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual in the Reference manual.

EN and ENO show or hide

To show or hide EN and ENO, carry out the following steps:

Step	Action
1	Open the properties dialog box, page 770 for the FFB.
2	Check the **FFB Properties** check box **Show EN/ENO** to show EN/ENO, or uncheck it to hide EN/ENO. **Note:** If EN or ENO is occupied by an actual parameter or connected with a link, EN/ENO cannot be hidden.

Properties dialog box for elementary functions, procedures and sub-routine blocks.

Calling the properties dialog box

see Displaying the properties, page 770

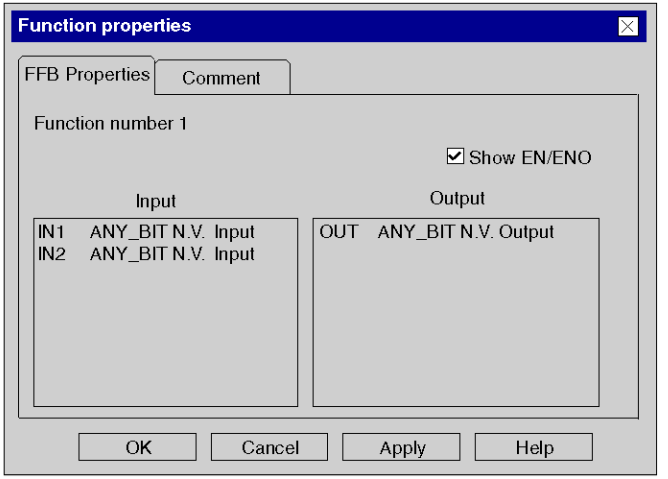
Structure of the properties dialog box

The properties dialog box consists of two tabs:

- **FFB Properties**
  General information regarding functions/procedures is displayed in this register.
- **Comment**
  You can enter a comment about the functions/procedures/sub-routine blocks in this tab.

FFB Properties tab

Representation of the **FFB Properties** tab:



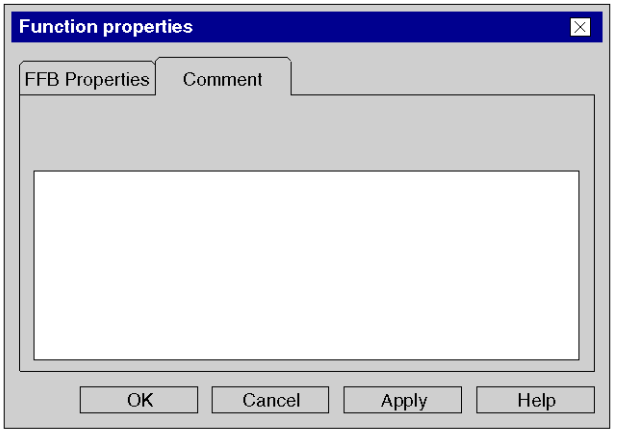
Elements of the **FFB Properties** tab:

Element	Description
Function	Specifies the name of the function number and the current execution order. This name cannot be modified.
Show EN/ENO	When you activate this checkbox the pins EN/ENO are displayed. When you uncheck this checkbox the pins EN/ENO are no longer displayed. **Note:** If EN or ENO is occupied by an actual parameter or connected with a link, the pins EN/ENO cannot be hidden.
Input	Displays the formal parameters, the data types and the actual parameter of the inputs. If the input is not yet linked, the entry **N.L. appears**.
Output	Displays the formal parameters, the data types and the actual parameter of the outputs. If the output is not yet linked, the entry **N.L. appears**.

Element	Description
OK	Use this button to accept all entries and close the dialog box.
Apply	Use this button to accept all entries without closing the properties dialog box.

Comment tab

Representation of the tab **Comment**:



Elements of the **Comment** tab:

Element	Description
Text box	Enter a comment about the FFB. This comment is displayed in the tooltip when the cursor is placed over the function/ procedure.
OK	Use this button to accept all entries and close the dialog box.
Apply	Use this button to accept all entries without closing the properties dialog box.

Properties dialog box for elementary and derived function blocks (FBs)

Calling the properties dialog box

see Displaying the properties, page 770

Structure of the properties dialog box

The properties dialog box consists of two tabs:

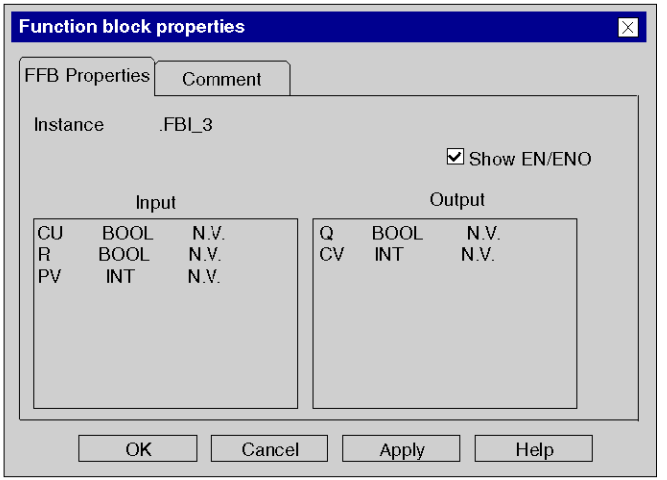
- **FFB Properties**

  General information regarding the FB is displayed in this register.
- **Comment**

  A comment about the FB can be entered in this tab.

FFB Properties tab

Representation of the **FFB Properties** tab:

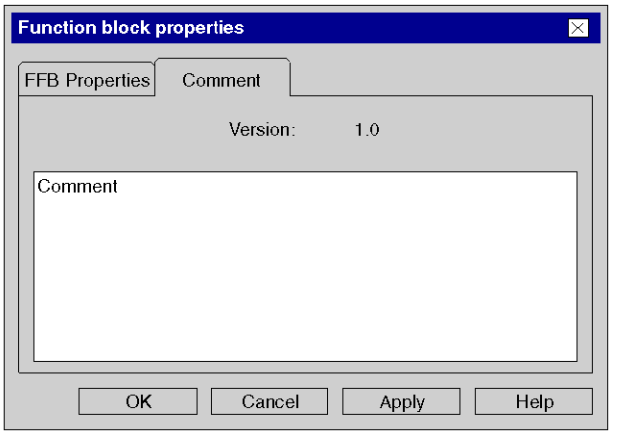


Elements of the **FFB Properties** tab:

Element	Description
Instance	Specifies the name of the block instance. You can change this name in the data editor, see also Managing of instances of data belonging to the family of function blocks (EF), page 380.
Show EN/ENO	When you activate this checkbox the pins EN/ENO are displayed. When you uncheck this checkbox the pins EN/ENO are no longer displayed. **Note:** If EN or ENO is occupied by an actual parameter or connected with a link, the pins EN/ENO cannot be hidden.
Input	Displays the formal parameters, the data types and the actual parameter of the FB inputs. If the input is not yet linked, the entry **N.L. appears**.
Output	Displays the formal parameters, the data types and the actual parameter of the FB outputs. If the output is not yet linked, the entry **N.L. appears**.
OK	Use this button to accept all entries and close the dialog box.
Apply	Use this button to accept all entries without closing the properties dialog box.

Comment tab

Representation of the tab **Comment**:



Elements of the **Comment** tab:

Element	Description
Version:	Specifies version of the FB code.
Text box	Enter a comment about the FB. This comment is displayed in the tooltip when the cursor is placed over the FB.
OK	Use this button to accept all entries and close the dialog box.
Apply	Use this button to accept all entries without closing the properties dialog box.

Returning from a Subroutine or DFB

Overview

This section describes how to return from a subroutine or DFB to the LD programming language.

Returning from a subroutine or DFB

Introduction

Every subroutine and every DFB (derived function block) is exited after being processed, i.e. you return to the main program called.

If the subroutine/DFB is left prematurely, the return to the main program can be forced by the return object.

If the status of the left link is ON, a return from the subroutine or DFB to the main program is performed.

Return objects can only be used in subroutines or DFBs. They cannot be used in main programs.

To generate a conditional return, a return object is placed at the end of a series of contacts.

Selecting the return object

You have the following options to select return objects:

- Use the **Edit > New > Return** menu command.

- Use the **Common Objects > Return** menu command from the shortcut menu.

  or

- Choose the  symbol.

The active placement mode for return objects is indicated by the  cursor symbol.

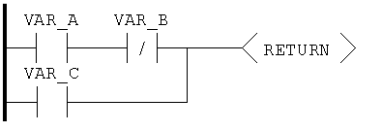
Note

In the **Tools > Options** dialog box on the **Data and Languages** tab, if the **Automatically assign a variable to a new graphical object** check box is selected, then the corresponding properties dialog box is automatically opened when an object is placed. The procedure explained here is valid for **deactivated** check boxes.

Placing return objects:

Placing return objects:

Step	Action
1	Create a subroutine, page 522 or a DFB, page 1256 in the LD programming language.
2	Create the logic for the subroutine/DFB.
3	Select the return object.

Step	Action
4	Click the target cell in the LD section. or Use the **arrow keys** to move the gray field to the target position and press **Enter**. **Result:**The return object is inserted. Example: 
5	To insert further return objects: • Click the target cell in the LD section. or Use the **arrow keys** to move the gray field to the target position and press **Enter**. • To return to selection mode: Press the **Esc** key. • To insert other objects: Select the object you wish to insert.

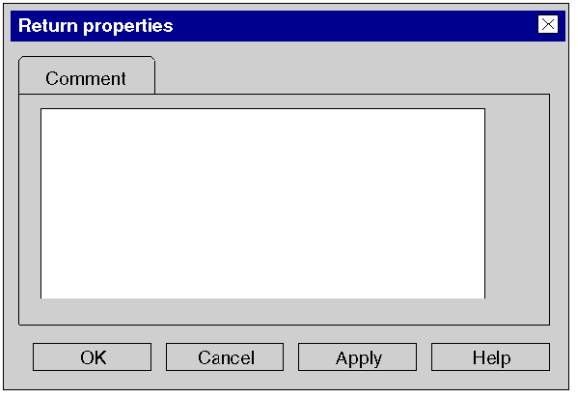
Return object properties dialog box

Calling the properties dialog box

see Displaying the properties, page 770

Structure of the properties dialog box

Representation of the properties dialog box:



Elements of the properties dialog box:

Element	Description
Text box	Enter a comment about the return object.
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Jumps within the Current Section

Overview

This section describes jumps within the current LD section.

Jumps within the current section

Introduction

Jump objects execute jumps to a jump target, page 848 within the current section.

When the status of the left link is 1, a jump is made to a label (in the current section).

To generate an unconditional jump, the jump object must be placed directly on the left power rail.

To generate a conditional jump, a jump object is placed at the end of a series of contacts.

Selecting the jump object

You have the following options to select jump objects:

- Use the **Edit > New > Jump** menu command.
- Use the **Common Objects > Jump** menu command from the shortcut menu.

  or

- Choose the  symbol.

The active placement mode for jump objects is indicated by the  cursor symbol.

Note

In the **Tools > Options** dialog box on the **Data and Languages** tab, if the **Automatically assign a variable to a new graphical object** check box is selected, then the corresponding properties dialog box is automatically opened when an object is placed. The procedure explained here is valid for **deactivated** check boxes.

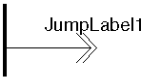
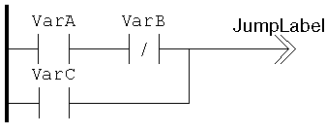
Placing jump objects:

Placing jump objects:

Step	Action
1	Select the jump object.
2	Click the target cell in the LD section. or Use the **arrow keys** to move the gray field to the target position, and press **Enter**. **Result:**The jump object is inserted.
3	To insert further jump objects: • Click the target cell in the LD section. or Use the **arrow keys** to move the gray field to the target position, and press **Enter**. • To return to selection mode: Press the **Esc** key. • To insert other objects: Select the object you wish to insert.

Defining the jump target

Defining the jump target:

Step	Action
1	Open the properties dialog box, page 770 for the jump object. **Result**: The properties dialog box for the jump object is opened, see also Jump object properties dialog box, page 846.
2	You have the following options to enter the jump target: • You can now enter the name of the jump target and confirm using **Enter**. or • You can select the name of the jump target from the list of available jump targets using the ▼ symbol and confirm using **Enter**. **Result:** The jump target is inserted and a syntax and semantics check, page 752 is performed. Example of an unconditional jump:  Example of a conditional jump:  If the jump target has already been defined (see also Jump target, page 848), the procedure ends here. If the jump target has not yet been declared, continue with step 3.
3	Specify the jump target, see also Jump target, page 848.

Jump object properties dialog box

Calling the properties dialog box

see Displaying the properties, page 770

Structure of the properties dialog box

The properties dialog box consists of two tabs:

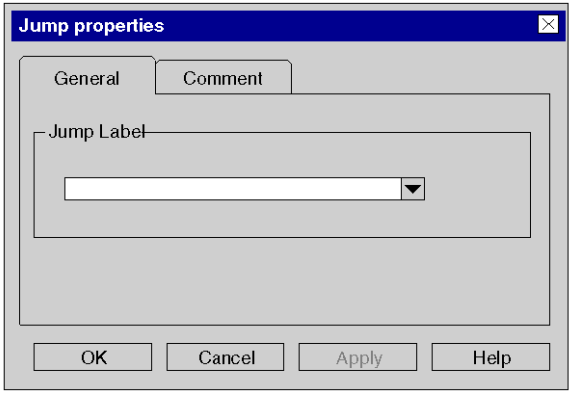
- **General**

  Enter the jump target of the jump, page 844 in this tab.
- **Comment**

  In this tab, a comment about the jump can be entered.

General tab

Representation of the **General** tab:

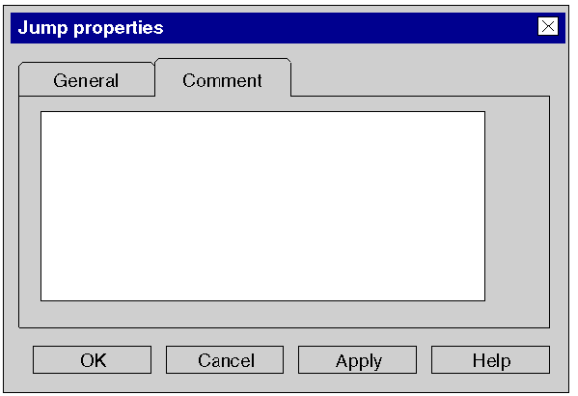


Elements of the **General** tab:

Element	Description
Jump Label	Enter the jump target in this text box, see also Definitions of jump targets (jump labels), page 848.
OK	Use this command button to accept all entries and close the properties dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Comment tab

Representation of the **Comment** tab:



Elements of the **Comment** tab:

Element	Description
Text box	Enter a comment about the jump object.
OK	Use this command button to accept all entries and close the properties dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Definition of the Jump Target (Jump Labels)

Overview

This section describes how to define jump targets in the LD programming language.

Definitions of jump targets (jump labels)

Introduction

Jump labels create a jump target for jumps, page 843 within the current section.

See reference manual to have more details: place, number of characters, naming conventions...

To find a specific jump label use the **Go To** dialog box.

Selecting the jump label

You have the following options to to select jump labels:

- Use the **Edit > New > Jump Label** menu command.
- Use the **Common Objects > Jump Label** menu command from the shortcut menu,

  or
- Choose the JL symbol.

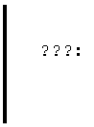
The active placement mode for jump labels is indicated by the JL cursor symbol.

Note

In the **Tools > Options** dialog box on the **Data and Languages** tab, if the **Automatically assign a variable to a new graphical object** checkbox is selected, then the corresponding properties dialog box is automatically opened when an object is placed. The procedure explained here is valid for **deactivated** checkboxes.

Placing jump objects

Placing jump objects

Step	Action
1	Select the jump label.
2	Click the target cell on the left power rail. or Use the **arrow keys** to move the gray field to the target position on the left power rail, and press **Enter**. **Result:**The return object is inserted. Example: 
3	Placing other jump label objects: • Click the target cell in the LD section. or Use the **arrow keys** to move the gray field to the target position and press **Enter**. • To return to selection mode: Press the **Esc** key. • To insert other objects: Select the object you wish to insert.

Position of the Name

Position of the Name

Step	Action
1	To enter the name, open the properties dialog box, page 770 for the jump label. **Result**: The jump label properties dialog box is opened, see also Jump label properties dialog box, page 851.
2	You have the following options to enter the name of the jump label: • You can now enter the name and confirm using **Enter**. or • You can select the name of the jump from the list of available jumps using the ▾ symbol and confirm using **Enter**. **Result:** The name is inserted and a syntax and semantics check, page 752 is performed. Example: `JumpLabel1:`

Jump label properties dialog box

Calling the properties dialog box

see Displaying the properties, page 770

Structure of the properties dialog box

The properties dialog box consists of two tabs:

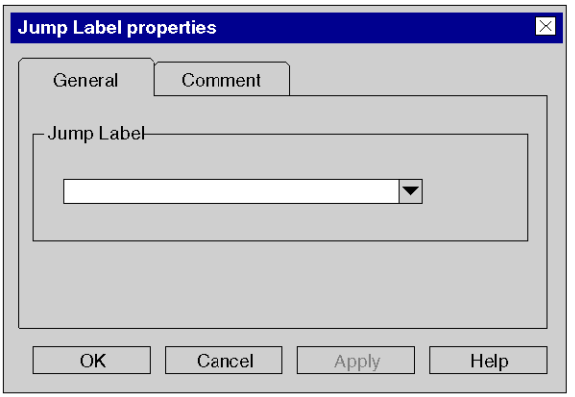
- **General**

  Enter the name of the jump label, page 849 in this tab.
- **Comment**

  In this tab, a comment about the jump label can be entered.

General tab

Representation of the **General** tab:

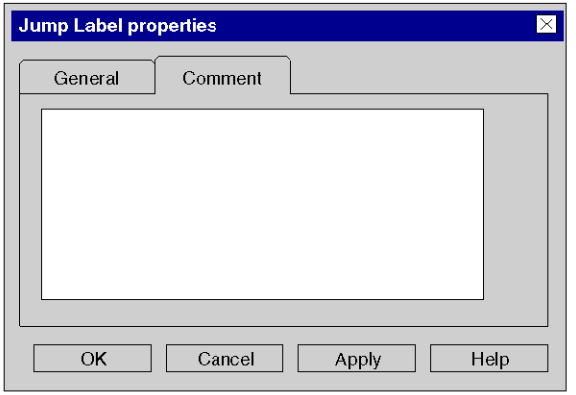


Elements of the **General** tab:

Element	Description
Jump Label	Enter the name of the jump label in this list box. The text is limited to 32 characters and must be unique across the entire section. The text must conform to the general naming conventions.
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Comment tab

Representation of the **Comment** tab:



Elements of the **Comment** tab:

Element	Description
Text box	Enter a comment about the jump label object.
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Editing Links

Overview

Editing links in the LD programming language.

Select links

Introduction

Links are connections between LD objects (contacts, coils, FFBs etc).

There are two types of links:

- Boolean links

  Boolean links comprise one or more segments that connect Boolean objects (contacts, coils) with one another.

  With Boolean links there are:

  - Horizontal connections

    Horizontal connections enable series switching of contacts and coils.

    The segments of this link can be created individually or as a complex link comprising several segments.

  - Vertical connections

    Vertical connections enable parallel switching of contacts and coils.

- FFB links

  FFB links are a combination of horizontal and vertical segments that connect FFB inputs/outputs with other objects.

  To avoid links crossing each other, FFB links can also be represented in the form of connectors, page 627.

FFB links are represented with double line thickness to distinguish them from Boolean links.

NOTE: A detailed description about links, see also section Links (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.

Selecting links

Selecting links:

Type of connection	Call via menu command	Call via symbol	Call via key	Cursor symbol
Horizontal connection, page 857	**Objects > Boolean Connection** or **Edit > New > Boolean Connection** or **Connection > Boolean Connection** from the shortcut menu		**F7**	
Link tool mode, page 859 (horizontal connection)	**Objects > Boolean Link** or **Edit > New > Boolean Link**		**Alt** + **F6**	
Vertical connection, page 861	**Objects > Vertical Connection**		**Shift** + **F7**	

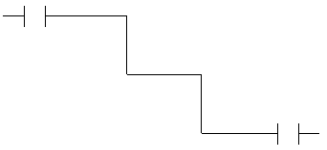
Type of connection	Call via menu command	Call via symbol	Call via key	Cursor symbol
	or **Edit > New > Vertical Connection** or **Link > Vertical connexion** from the shortcut menu			
FFB link, page 863	**Objects > Link** or **Edit > New > Link**		**F6**	
Derivation, page 871	–		–	

Combining Links

Boolean links

Horizontal and vertical Boolean links can be combined in any way.

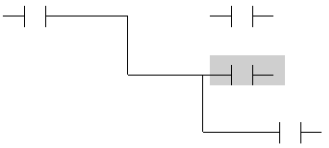
Example:



Contacts and coils

Contacts and coils link automatically to existing horizontal or vertical links if no free cells are found between them.

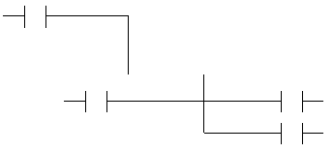
Example:



Crossing Boolean links

If two Boolean links cross each other they are automatically connected. Since Boolean links may not cross each other there is no special label for them.

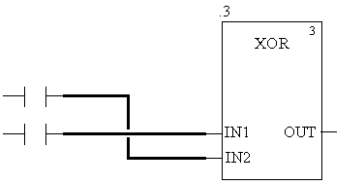
Example:



Crossing FFB links

If two FFB links cross each other a connection is not made between them. Crossed links are indicated by a "broken" link.

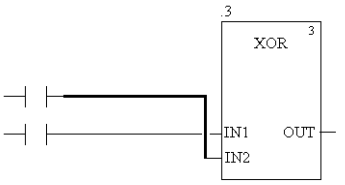
Example:



Crossing FFB links and Boolean links

If FFB links and Boolean links cross each other, a connection is not made between them. Crossed links are indicated by a "broken" link.

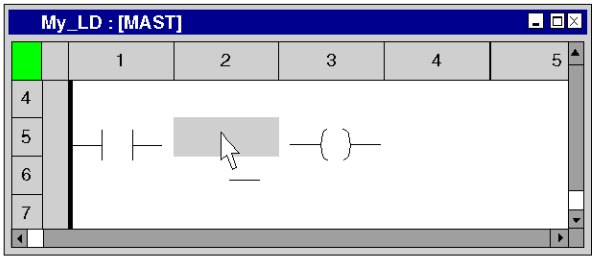
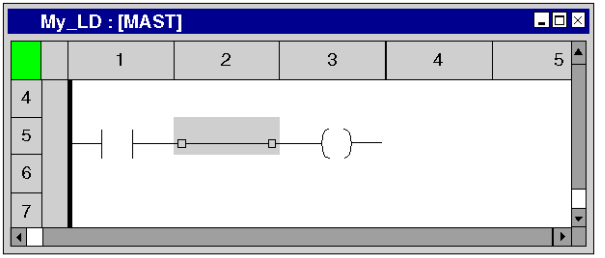
Example:



Placing links

Placing horizontal links

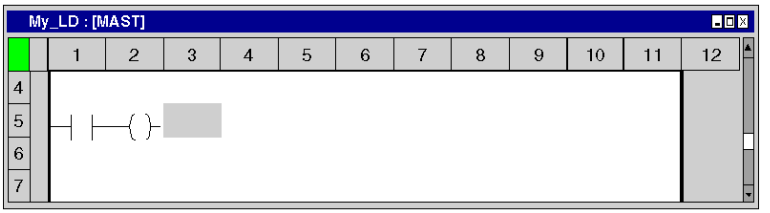
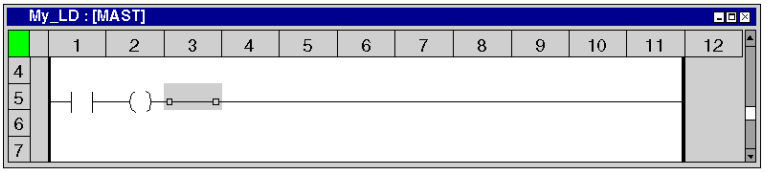
Process the following steps to place a horizontal link:

Step	Action
1	Activate the placement mode for horizontal links, see also Select links, page 853. Example: 
2	Click the target cell in the LD section. or Use the **arrow keys** to move the gray field to the target position and press **Enter**. **Result:** The link is inserted. Example:  See also • Combining Links, page 855 • Placing horizontal and vertical links with FBB links using the mouse, page 867 using the keyboard, page 869 • Editing links, page 872
3	To place further links of the same type: • Click the target cell in the LD section. or Use the **arrow keys** to move the gray field to the target position and press **Enter**. • To return to selection mode: Press the **Esc** key. • To insert other objects:

Step	Action
	Select the object you wish to insert.

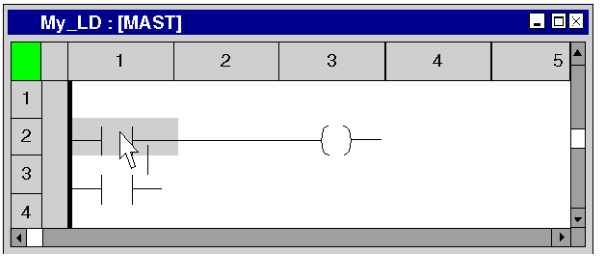
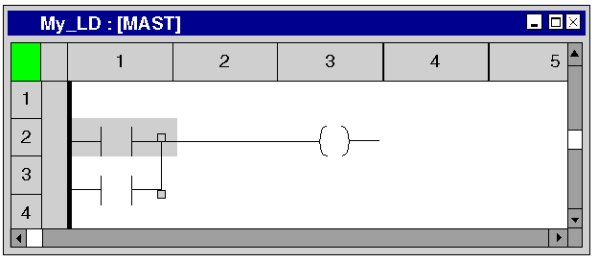
Placing horizontal links in link-tool mode

Process the following steps to place a horizontal link in link-tool mode:

Step	Action
1	Activate the link-tool mode, see also Select links, page 853. Example: 
2	Click the target cell in the LD section. or Use the **arrow keys** to move the gray field to the target position and press **Enter**. **Result:** The link is inserted and a connection is created to the neighboring Boolean objects to the left and right. If there are no neighboring Boolean objects available, a connection to the power rail is made. This function is a good way to connect coils with the right power rail. Example:  See also • Combining Links, page 855 • Placing horizontal and vertical links with FBB links using the mouse, page 867 using the keyboard, page 869 • Editing links, page 872
3	To place further links of the same type: • Click the target cell in the LD section. or Use the **arrow keys** to move the gray field to the target position and press **Enter**. • To return to selection mode: Press the **Esc** key. • To insert other objects: Select the object you wish to insert.

Placing vertical links

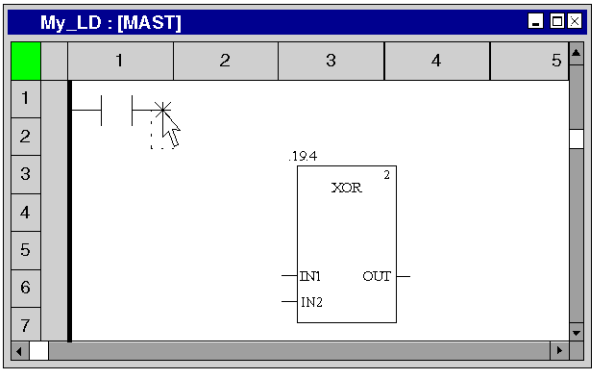
On the right side of the cells, there is space to create a vertical link to the cell below. Process the following steps to place a vertical link:

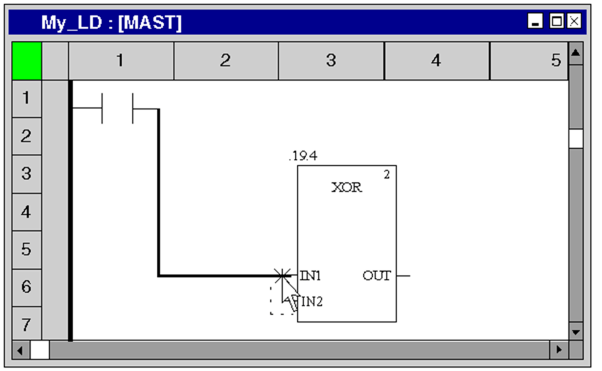
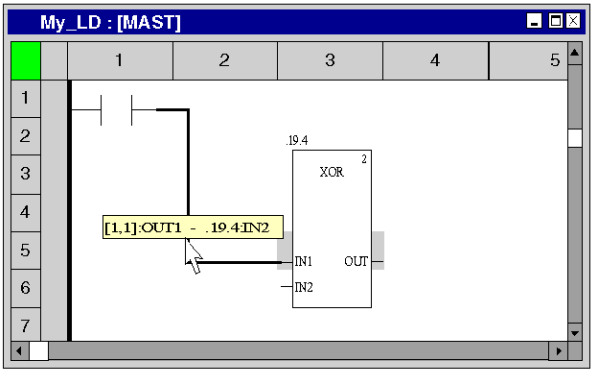
Step	Action
1	Activate the placement mode for vertical links, see also Select links, page 853. Example: 
2	Click the contact which should be connected with the cell below. or Use the **arrow keys** to move the gray field to the target position and press **Enter**. **Result:** The link is inserted and the selection mode is active again. Example:  See also • Combining Links, page 855 • Placing horizontal and vertical links with FBB links using the mouse, page 867 using the keyboard, page 869 • Editing links, page 872
3	To place further links of the same type: • Click the target cell in the LD section. or Use the **arrow keys** to move the gray field to the target position and press **Enter**. • To return to selection mode: Press the **Esc** key. • To insert other objects:

Step	Action
	Select the object you wish to insert.

Placing an FFB link with the mouse

Process the following steps to place an FFB link using the mouse:

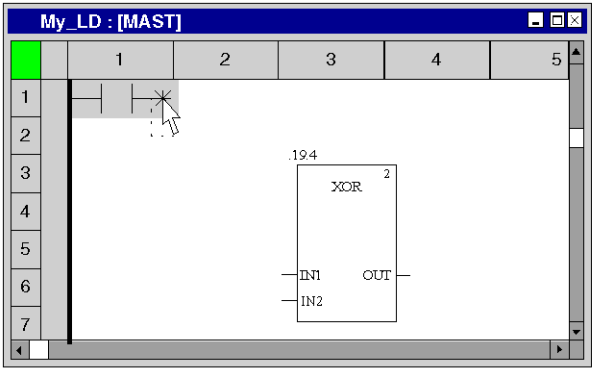
Step	Action
1	Activate the placement mode for FFB links. See also Select links, page 853.
2	Place the mouse pointer on the start point of the link. **Notes:** When creating an FFB link, first select the output pin and then the input pin. The cursor symbol indicates the correct position . Example: 

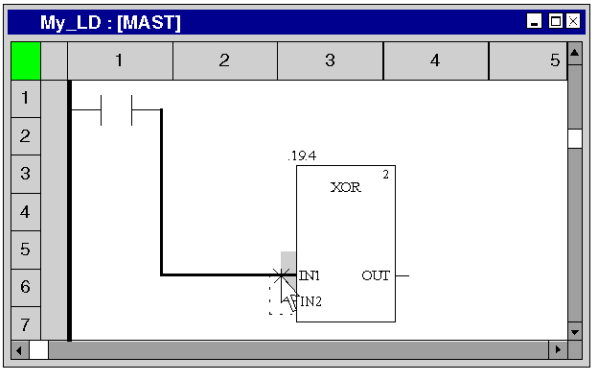
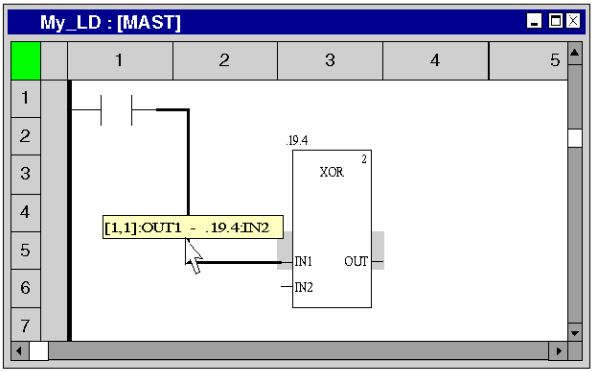
Step	Action
3	Click the left mouse button on the start point and drag mouse pointer to the target point of the link. Example:  **Note:** For FFB links, at least one FFB input or output must be used. An FFB link between contacts and coils is automatically resolved into a combination of horizontal and vertical links. **Tips:** • By clicking between the start point and target position you can place intermediate points; for example, to avoid crossing other objects. • By clicking **Esc**, you quit the mode and the delete the link segments you created. • By clicking **Enter** you quit the mode, and the link segments you created are resolved into a combination of horizontal and vertical links.
4	Click on the target position with the left mouse button **Result:** The link is inserted. When the cursor is placed over an FFB link, the tooltip displays the source and the destination of the link. Example:  See also

Step	Action
	• Combining Links, page 855 • Editing links, page 872

Placing an FFB link with the keyboard

Process the following steps to place an FFB link using the keyboard:

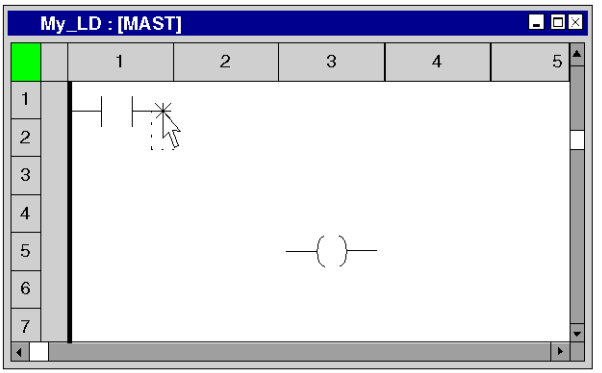
Step	Action
1	Use the **arrow keys** to move the gray cell cursor to the source object (for example, contact) of the link. **Note:** When creating an FFB link, first select the output pin and then the input pin.
2	Activate the placement mode for horizontal links using **F6**. See also Select links, page 853.
3	Press the **Ctrl**+**Enter** key combination. **Result:** The output pin of the object in the gray cell is selected as the start point of the FFB link. Example: 

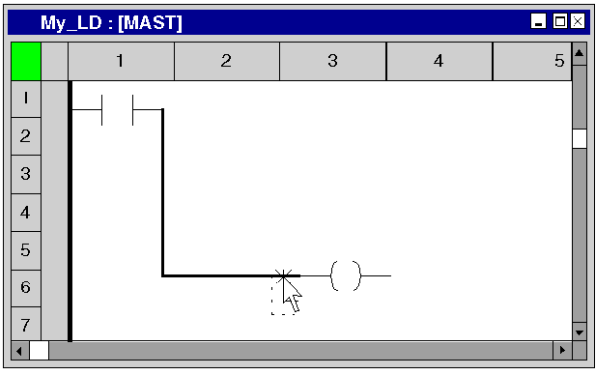
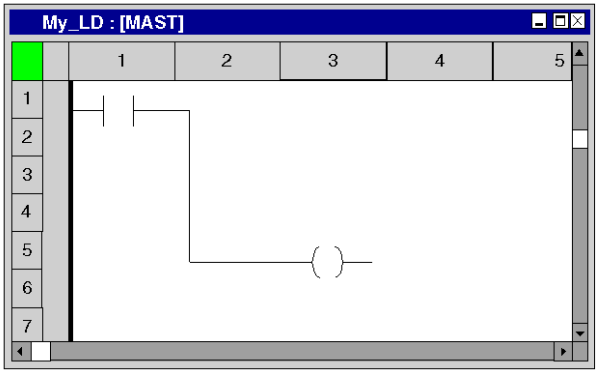
Step	Action
4	Use the **Ctrl**+**arrow keys** to move the cursor to the target position of the link. **Tips:** • By pressing the **Ctrl**+**Enter** key combination between the start point and target position you can place intermediate points; for example, to avoid crossing other objects. • By clicking **Esc**, you quit the mode and the delete the link segments you created. • By clicking **Enter** you quit the mode, and the link segments you created are resolved into a combination of horizontal and vertical links. Example:  **Note:** For FFB links, at least one FFB input or output must be used. An FFB link between contacts and coils is automatically resolved into a combination of horizontal and vertical links.
5	Press the **Ctrl**+**Enter** key combination. **Result:** The link is inserted. **Tip:**When the cursor is placed over an FFB link, the tooltip displays the source and the destination of the link. Example: 

Step	Action
	See also • Combining Links, page 855 • Editing links, page 872

Placing horizontal and vertical links using FFB links with the mouse

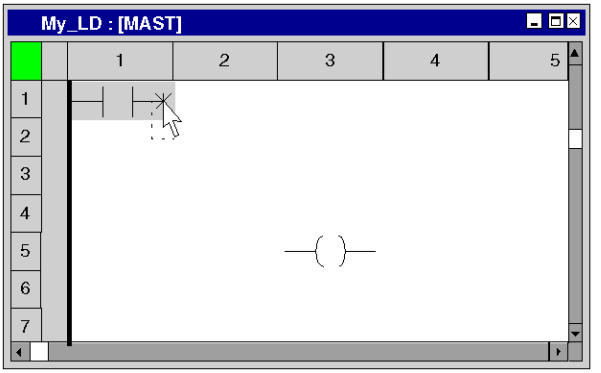
The functions for creating FFB links can also be used to create horizontal and vertical links between other objects (not FFBs). If an FFB link involves no input or output, the FFB link is automatically resolved into a combination of horizontal and vertical links. To do this, process the following steps:

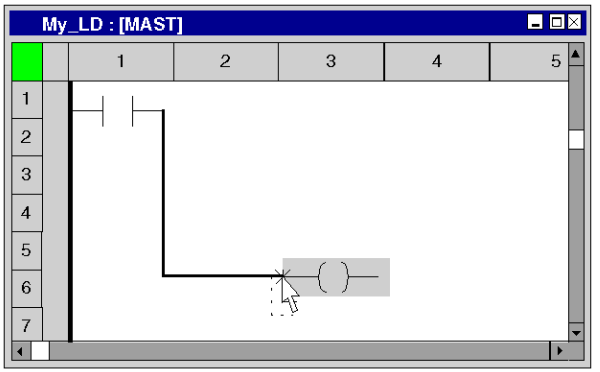
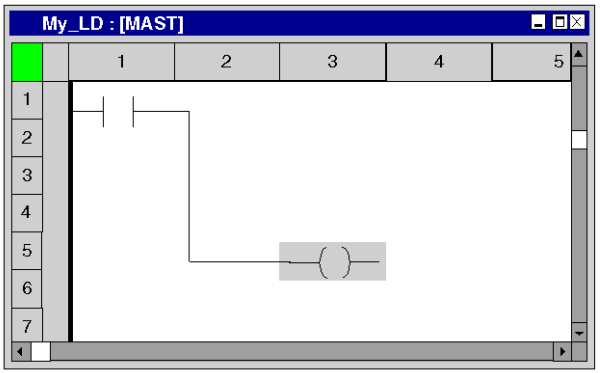
Step	Action
1	Activate the placement mode for FFB links. See also Select links, page 853.
2	Place the mouse pointer on the start point of the link. **Notes:** When creating an FFB link, first select the output pin and then the input pin. The cursor symbol indicates the correct position . Example: 

Step	Action
3	Click the left mouse button on the start point and drag mouse pointer to the target point of the link. Example:  **Tips:** • By clicking between the start point and target position you can place intermediate points; for example, to avoid crossing other objects. • By clicking **Esc**, you quit the mode and the delete the link segments you created. • By clicking **Enter** you quit the mode, and the link segments you created are resolved into a combination of horizontal and vertical links.
4	Click on the target position with the left mouse button **Result:**The link is inserted and the FFB link is automatically resolved into a combination of horizontal and vertical links. Example:  See also • Combining Links, page 855 • Editing links, page 872

Placing horizontal and vertical links using FFB links with the keyboard

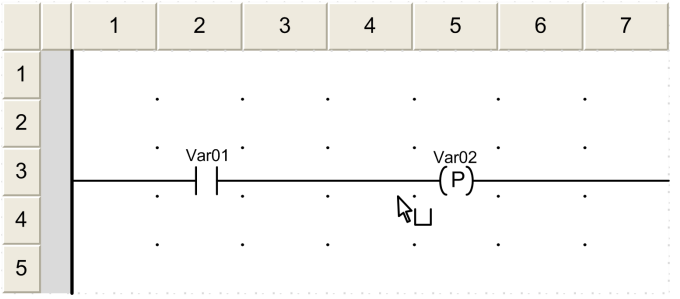
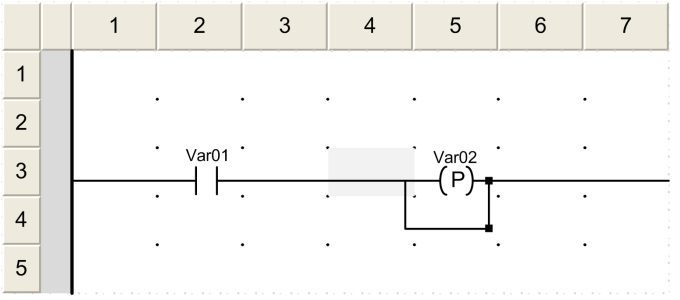
The functions for creating FFB links can also be used to create horizontal and vertical links between other objects (not FFBs). If an FFB link involves no input or output, the FFB link is automatically resolved into a combination of horizontal and vertical links. To do this, process the following steps:

Step	Action
1	Use the **arrow keys** to move the gray cell cursor to the source object (for example, contact) of the link. **Note:** When creating an FFB link, first select the output pin and then the input pin.
2	Activate the placement mode for horizontal links using **F6**. See also Select links, page 853.
3	Press the **Ctrl**+**Enter** key combination. **Result:** The output pin of the object in the gray cell is selected as the start point of the FFB link. Example: 

Step	Action
4	Use the **Ctrl**+**arrow keys** to move the cursor to the target position of the link. **Tips:** • By pressing the **Ctrl**+**Enter** key combination between the start point and target position you can place intermediate points; for example, to avoid crossing other objects. • By clicking **Esc**, you quit the mode and the delete the link segments you created. • By clicking **Enter** you quit the mode, and the link segments you created are resolved into a combination of horizontal and vertical links. Example: 
5	Press the **Ctrl**+**Enter** key combination. **Result:**The link is inserted and the FFB link is automatically resolved into a combination of horizontal and vertical links. Example:  See also • Combining Links, page 855 • Editing links, page 872

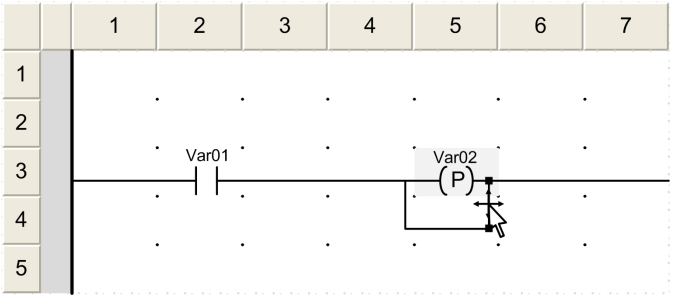
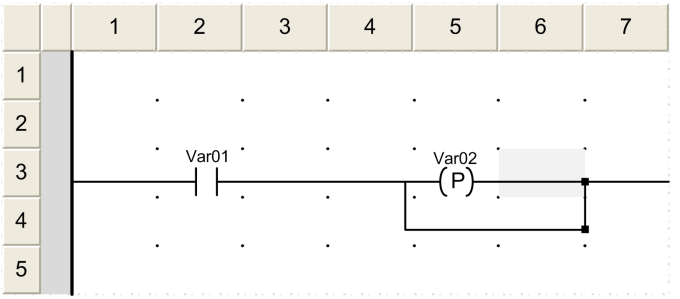
Placing derivation links

Process the following steps to place a derivation link:

Step	Action
1	Activate the derivation-tool, page 853. 
2	Click the target cell in the LD section. This cell is the starting point of the derivation. **Result:** The derivation is inserted.  **NOTE:** The width and height of the derivation link is one cell. The derivation can be enlarged.

Resizing derivation links

Process the following steps to enlarge a derivation link:

Step	Action
1	Activate the select mode.
2	Left-click and keep it pressed one vertical link of the derivation you want to enlarge. Example: 
3	Drag the mouse to the new position for the vertical link. **Result:** The derivation link is enlarged. Example: 

Editing links

Editing Boolean links

Horizontal and vertical Boolean links contain one or more independent segments.

When cutting, page 763, deleting, page 763, copying, page 763 and moving, page 766 horizontal and vertical links, the action is executed for the explicitly selected, page 760 segment.

Editing FFB links

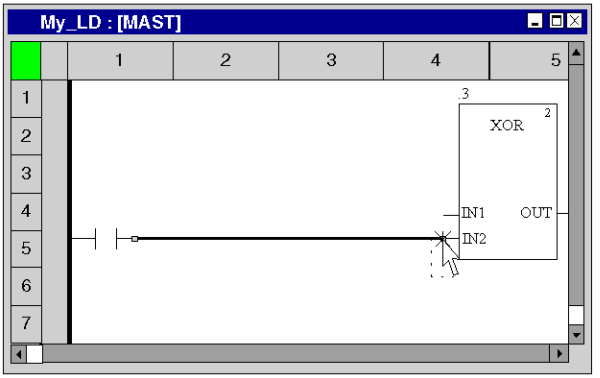
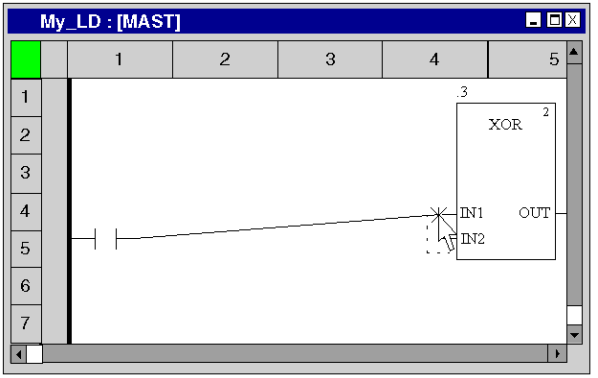
FFB links comprise one or more segments connected to one another.

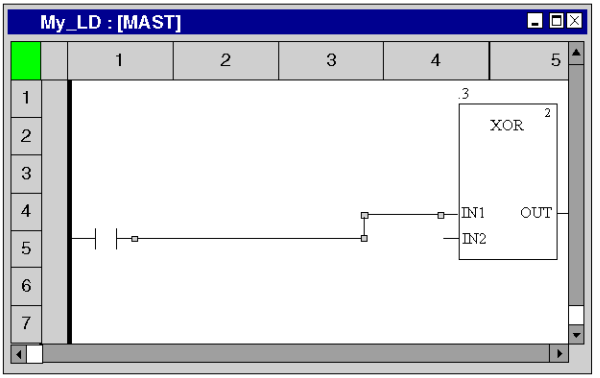
When cutting, page 763, deleting, page 763, copying, page 763 and moving, page 766 FFB links, the action is executed for the entire link.

If an FFB link is selected, the target points are displayed at their beginnings, ends and at every direction change

Modifying connections

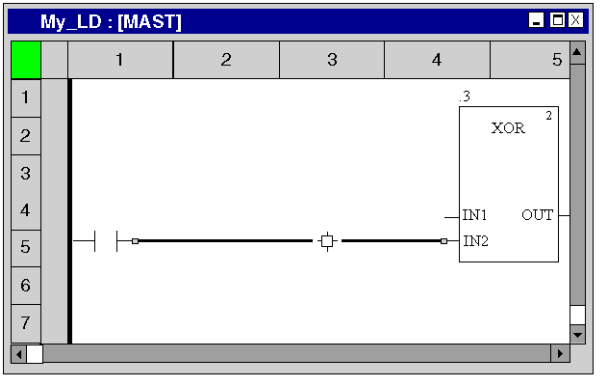
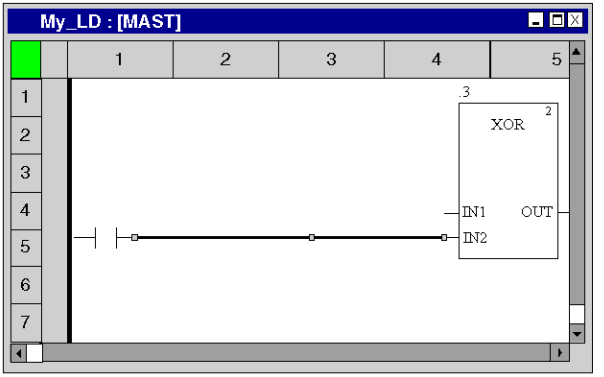
Modifying connections:

Step	Action
1	Select the link and place the mouse pointer on the connection to be modified. Example: 
2	Left-click the connection to be modified and drag the mouse pointer to the new connection. Example: 
3	Left-click the new target position. **Result:** The link route is changed. Example:

Step	Action
	

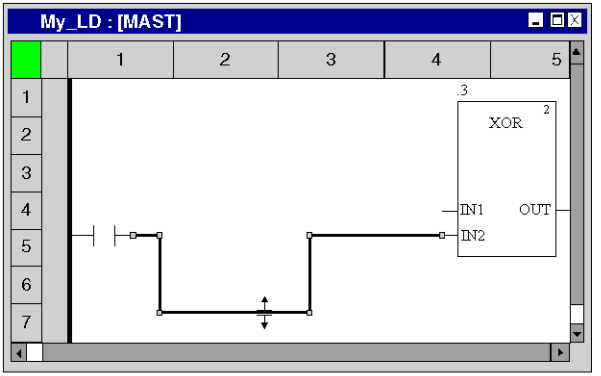
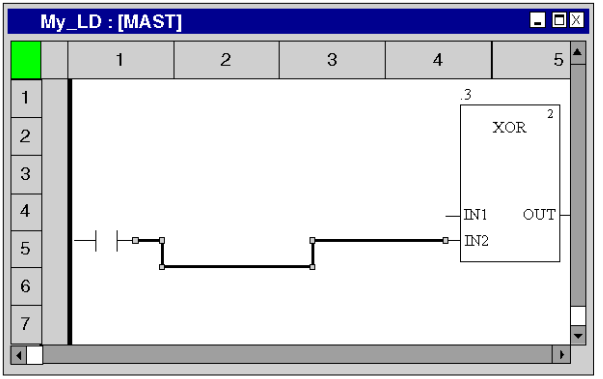
Creating sizing handles

The creation of angles in FFB links can only be made using sizing handles. If no sizing handles are available they can be created. Process the following steps to create sizing handles in FFB links:

Step	Action
1	Select the FFB link.
2	Press the **Ctrl** key and place the mouse pointer at the target position. **Note:** The correct position is indicated by the mouse pointer symbol. Example: 
3	Left-click the target position. **Result:** The sizing handle is created. 

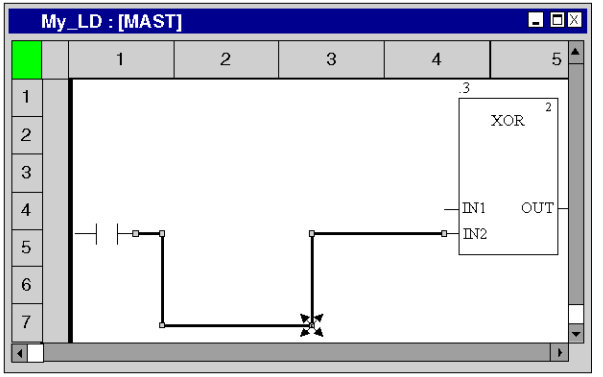
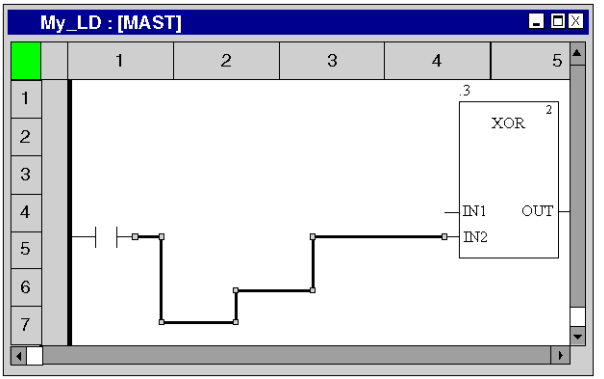
Moving FFB links vertically/horizontally

Process the following procedures to move an FFB link segment vertically or horizontally:

Step	Action
1	Select the FFB link.
2	Place the mouse pointer on the item to be moved. **Note:** The correct position is indicated by the mouse pointer symbol. Example: 
3	Click the left mouse button and drag the segment to the target position.
4	Release the mouse button. **Result:** The change is made. 

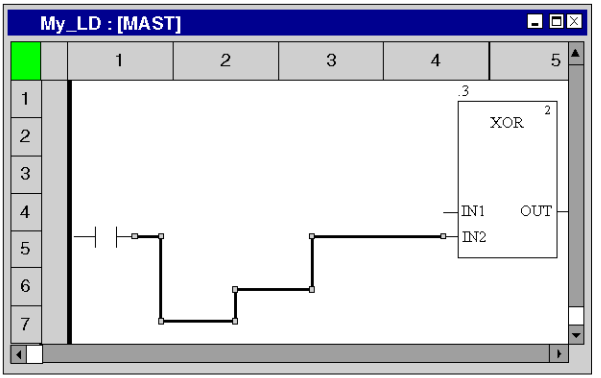
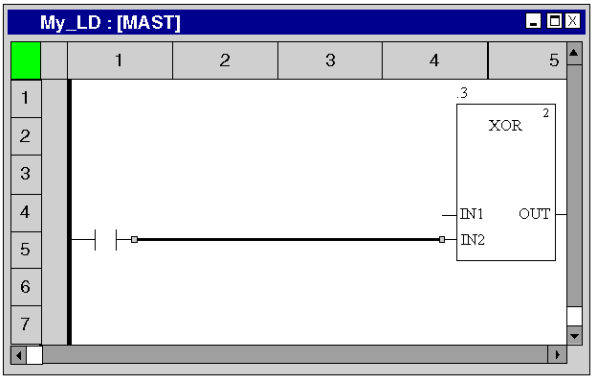
Creating angles in FFB links

Process the following steps to create angles in FFB links:

Step	Action
1	Select the FFB link.
2	Place the mouse pointer on the source position. **Note:** The correct position is indicated by the mouse pointer symbol. Example: 
3	Click the left mouse button and drag the source point to the target position.
4	Release the mouse button. **Result:** The change is made. 

Determining the optimum link route

Determining the optimum link route:

Step	Action
1	Select the link. Example: 
2	Use the **Edit > Link > Route** menu command. or Use the **Route** menu command from the shortcut menu. **Result:** The current link route is changed to the optimum route. Example: 

Entering Comments

Overview

This section describes how to enter comments in the LD programming language.

Entering comments

Introduction

Comments can be placed in ladder diagram LD in the form of text objects.

Text objects can overlap other objects.

The size of the object, depending on the size of the text, can be extended vertically and horizontally to fill further grid units.

The same rules apply for entering text and navigating within text objects as for editing ASCII text in standard text editors. (To create a line break in text objects you must press the **Ctrl** +**Enter** key combination).

NOTE: Keep in mind that each change to a comment (e.g. change to a comment text, change to the size of the text object) makes it necessary to recreate the section involved (**Generation > Generate project**).

Displaying or hiding the comments

Use the  button in the toolbar to show or hide the comments.

The background color of the button in blue indicates that the comments are hidden.

NOTE: To select or place text objects, check that the  button in the toolbar is not activated.

Selecting the text object

You have the following options to select text objects:

- Use the **Objects > Comment** menu command.
- Use the **Edit > New > Comment** menu command.

- Use the **Common objects > Comment** menu command from the shortcut menu.
- Press the **F8** key.

  or

- Select the  symbol.

The active placement mode for text objects is indicated by the  cursor symbol.

Placing text objects:

NOTE: A double-click on a line number in the vertical ruler creates a comment block all along the entire line.

Placing text objects:

Step	Action
1	Select the text object.
2	Click the target cell in the LD section. or Use the **arrow keys** to move the gray field to the target position and press **Enter**. **Result:**The text object is inserted. Example: 
3	Enter the comment.
4	Confirm the text entered with: • Clicking with the mouse outside of the text object or • Pressing the **Enter** key.
5	To insert further text objects: • Click the target cell in the LD section. or Use the **arrow keys** to move the gray field to the target position and press **Enter**. • To return to selection mode: Press the **Esc** key. • To insert other objects: Select the object you wish to insert.

Online functions

Overview

This section describes the online functions of the LD programming language.

Online Functions

Online Functions

The description of online functions is found in chapter Debugging in Ladder Language, page 1413.

LD Reference Data Type

Reference Data Type in LD

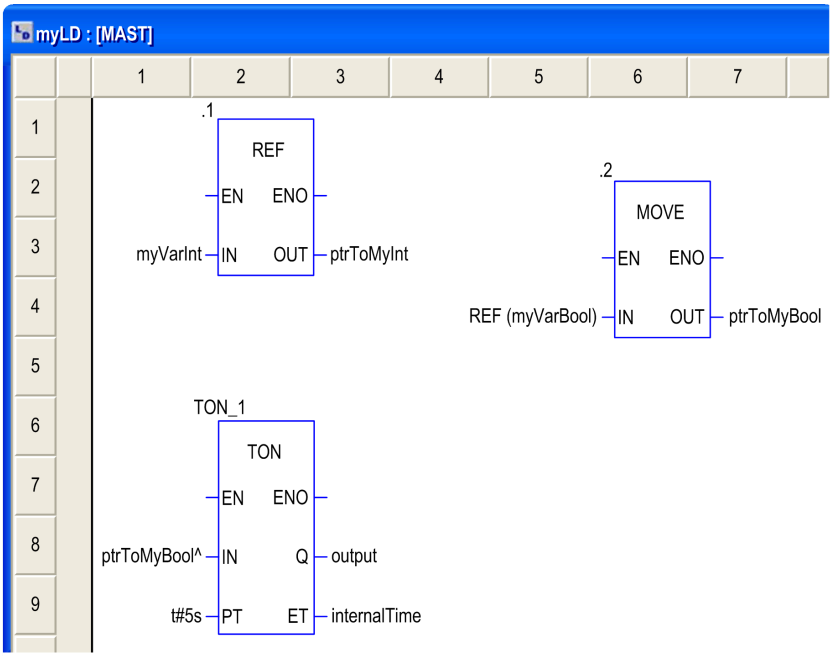
Introduction

References to variables can be assigned in LD application programs.

A reference can be de-referenced in LD application programs.

For more information, refer to Reference Data Type (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).

LB Examples using the Reference Data Type:



The EF, `REF`, takes a variable as an input parameter (type *ANY*) and provides its address as output parameter (type *REF_ANY*).

The EF, `MOVE`, can assign a reference to a variable to another reference. If the input parameter is (*REF(MyVariable)*), `MOVE` has the same effect as the `REF` EF.

The EF, `TON`, takes *ptrToMyBool^* as an input parameter to dereference the reference ptrToMyBool.

Export/Import

Overview

This section describes the Import/Export functions of the LD programming language.

Export/Import LD Sections

Export/Import

The description for exporting/importing sections is found in the chapter Import / Export, page 1686.

Customize LD Editor

Overview

This section describes how to customize the toolbar and the colors of the LD editor.

Customize LD Editor Colors

Overview

To customize LD editor colors, open the **Tools > Colors Settings...** dialog box.

The **Color Settings** dialog box allows to:

- Modify the color of the selected item in the list.
- Reset the colors with initial values.
- Import an *.ini file that defines the colors associated to the LD editor.
- Export the definition colors associated to the LD editor.

Color Setting File

Depending on the operating system of the PC, the color setting file LDColors.ini is located in one of the following folders:

- `c:\Program Files\Schneider Electric\Control Expert ••.•\`
- `c:\Program Files (x86)\Schneider Electric\Control Expert ••.•\`

For more detail on destination folder when Control Expert is installed, refer to *EcoStruxure™ Control Expert, Installation Manual*.

NOTE: Make a copy of this file before modifying it.

Description

This table describes the default color values for each parameter in the LDColors.ini file:

Parameter	Default value (R, G, B)	Description
BkAnimation	192,192,192	Editor background color when animation is ON and you are connected in programming mode.
StatementError	255,0,0	Element in error is underlined with this color.
InspectBk	0,255,255	Inspect window background color
InspectText	0,0,0	Inspect window text color
InspectBkMin	255,255,0	Inspect window background color when the value is less than the minimum value defined in the inspect window settings.
InspectBkMax	255,0,255	Inspect window background color when the value is greater than the maximum value defined in the inspect window settings.
BkAnimMonitoring	255,211,211	Editor background color when animation is ON and you are connected in monitoring mode.
AnalyzeError	0,0,255	Drawing color when an element generates an analyze error.
GraphAnimBoolTrue	0,150,0	Color of true BOOL in animation mode.
GraphAnimBoolFalse	255,0,0	Color of false BOOL in animation mode.
BkComment	255,255,204	Background color of comment area
BkCommentEdit	225,225,225	Background color of comment area when you are editing it.
BkAnimNumeric	255,255,0	Background color of numeric variables
BlockBreakpoint	128,0,0	Bullet color in the element where the breakpoint is set.
BlockCurrentStep	255,255,0	Color of the current element in debug mode
AnchorLink	128,0,0	Color of anchor link
LLFgExpression	0,0,0	Text color of boolean expressions
LLFgAddress	100,100,230	Text color of variable address
LLFgSymbol	0,0,0	Text color of variable name
LLFgComment	0,128,0	Text color of variable comment
LLBkExpression	255,255,255	Background color of boolean expressions
LLBkAddress	255,255,255	Background color of variable address
LLBkSymbol	255,255,255	Background color of variable name

Parameter	Default value (R, G, B)	Description
LLBkComment	`255,255,255`	Background color of variable comment
Background	`255,255,255`	Editor background color.

LD Instruction Bar

Overview

The toolbar **LD Instruction Bar** contains the timers and counters EFB for direct call.

For detailed information about these EFBs refer to chapter *Timers and Counters*.

Showing and Hiding the Toolbar

You have two options to show the toolbar **LD Instruction Bar** (to open it):

- Right-click in the common toolbar and select the toolbar from the shortcut menu.
- Open the **Tools > Customize** dialog box and select the toolbar,

You have three options to hide the toolbar **LD Instruction Bar** (to close it):

- Right-click in the common toolbar and deselect the toolbar from the shortcut menu,
- Open the **Tools > Customize** dialog box and deselect the toolbar,
- With the close symbol (x) if the toolbar is non-anchored.

Timers and Counters EFB

The toolbar **LD Instruction Bar** contains the following EFBs organized in 4 drop-list buttons 

:

CTD	Downcounter EFBs • CTD • CTD_DINT • CTD_INT • CTD_UDINT • CTD_UINT
CTU	Upcounter EFBs • CTU

	• CTU_DINT • CTU_INT • CTU_UDINT • CTU_UINT
CTUD	Up and down counter EFBs • CTUD • CTUD_DINT • CTUD_INT • CTUD_UDINT • CTUD_UINT
TIME	Timer EFBs • TOF • TON • TP

LD Favorite Bar

Overview

In the current LD editor toolbar, the F button allows to manage the toolbar **LD Favorite Bar** which can contain a maximum of 20 EF, EFB or DFB for direct call.

Showing and Hiding the Toolbar

You have two options to show the toolbar **LD Favorite Bar** (to open it):

- Right-click in the common toolbar and select the toolbar from the shortcut menu.
- Open the **Tools > Customize** dialog box and select the toolbar,

You have three options to hide the toolbar **LD Favorite Bar** (to close it):

- Right-click in the common toolbar and deselect the toolbar from the shortcut menu,
- Open the **Tools > Customize** dialog box and deselect the toolbar,
- With the close symbol (x) if the toolbar is non-anchored in the current LD editor toolbar.

Managing the Toolbar

To manage the toolbar **LD Favorite Bar**, opens the **Add/Delete Favorite** dialog box by clicking the  button.

The **Add/Delete Favorite** dialog box allows to:

- Add component via the **FFB Type Selection** dialog box.
- Delete component.
- Import an `*.ini` file that define the components for the **LD Favorite Bar**.
- Export the components of the **LD Favorite Bar**.

  NOTE: The components of the **LD Favorite Bar** are saved in the `Favorite.ini` file.

  For example with Control Expert 14.0, the `Favorite.ini` file is located in `C:\ProgramData\Schneider Electric\Control Expert\14.0\Favorite\` folder.

SFC Editor

What's in This Chapter

Creating a program with the SFC programming language 890
General Information about Steps 920
Editing "normal" steps 923
Editing macro steps 936
Editing input steps 947
Editing Output Steps 953
Editing actions 957
Editing Transitions 968
Editing Jumps 977
Editing Branches and Joints 983
Inserting Alternative Strings and Simultaneous Strings 991
Inserting Step-Transitions and Transition-Step-Sequences 995
Editing Links 999
Entering Comments 1010
Online functions 1012
Export/Import 1012
Customize SFC Editor Colors 1012

Overview

This chapter describes the menus and dialogs of the SFC editor.

For a description of the syntax for the SFC programming language, please refer to the Sequential Function Chart SFC (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) chapter in the Reference manual.

Creating a program with the SFC programming language

Overview

This section describes how to create a program in the SFC programming language.

Structure of an SFC Program (Sequential Function Chart)

Introduction

IEC conforming sequential control is created from SFC sections (top level), transition sections and action sections.

SFC sections are only permitted in the projects master task. SFC sections can not be used in other tasks or DFBs.

Each SFC section contains one or more SFC networks (sequences).

Structure of an SFC section

Zero or more actions belong to every step. A transition condition belongs to every transition.

The last transition of the sequence is always connected to the first step of the sequence (using a graphical link or jump label). Step sequences run cyclically.

Properties of an SFC program

Properties of an SFC program:

- SFC section always have a grid background.
- Because of performance reasons, it is strongly recommended to create less than 100 SFC sections in a project (macro section are not counted).
- An SFC section contains a maximum of 200 lines and 64 columns.
- SFC objects can theoretically be placed in every unoccupied cell.
- Steps, transitions and jumps each require a cell. Branches and joints do not require their own cell, but are inserted into the respective cell of the step or transition. Text objects can be placed anywhere in the window.
- Steps and transitions are linked with one another through directional links. Two steps can never be directly linked, and must always be separated by a transition, see also section Connection rules (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.
- To prevent step sequences being subdivided, 99 linked steps with the transitions are vertically displayed along with a locking jump with its transition.
- A maximum of 1024 steps can be placed per SFC section (including all their macro sections).

- A maximum of 100 steps can be active per SFC section (including all their macro sections) (multi token), see also **Tools > Project Settings** dialog box, **Language extensions** tab, **Sequential Function Chart (SFC)** area.
- A maximum of 64 steps can be set simultaneously per SFC section (multi token), see also **Tools > Project Settings** dialog box, **Language extensions** tab, **Sequential Function Chart (SFC)** area.
- A maximum of 100 actions are permitted using the S qualifier per SFC Section.
- A maximum of 20 actions can be assigned to each SFC step.
- The processes of the active signal status take place along the directional links, triggered by the connecting of a transition. The direction of the string process follows the directional links and runs from the under side of the predecessor step to the top side of the successive step. Branches are processed from left to right.
- A syntax and semantics check, page 894 is performed immediately after the statement is entered. The result of this check is displayed in colored text.
- Syntactically or semantically incorrect sections can be saved.

SFC program objects

The objects of the programming language SFC (Sequential Function Chart) help to divide a section into a number of:

- Step, page 923
- Macro Step, page 936 (embedded sub-step sequence)
- Transitions, page 968 (transition conditions)
- Jump, page 977
- Alternative Sequence, page 983
- Parallel Sequence, page 983

These objects can be linked with each other using links, page 999.

Comments for the logic of the program can be added using text objects, page 1010.

Edit and view functions

Edit and view functions of the SFC editor:

- Selecting objects, page 897
- Deleting objects, page 900
- Cutting, page 901, copying, page 902 and pasting, page 903 objects
- Moving, page 904 objects

- Undo, page 899 and Redo, page 899
- Using bookmarks, page 615
- Finding and replacing, page 1239 variables

Online functions

Online functions of the SFC editor:

- Setting breakpoints, page 1443
- Controlling, page 1445 SFC sequences

SFC save and restore

The `SFC_RESTORE` function block is used to restart all SFC charts of an application with a given set of active steps at a state, saved before a controller error.

In a recovery situation it is possible to set a group of steps that represent a given state in the process and to continue at that point.

(See SFC_RESTORE: SFC Save and Restore (see EcoStruxure™ Control Expert, System, Block Library) in the *System Library*).

Creating an SFC Program

Creating an SFC program

Carry out the following procedures to create an SFC program:

Step	Action
1	Creating an SFC Section., page 510
2	Place the desired SFC element in the section and define its properties: • Step, page 920 • Macro step, page 936 • Transitions, page 968 • Jumps, page 977 • Alternative strings, page 983 • Simultaneous strings, page 983 • Links, page 999
3	Create the last transition of the string and the first step of the string using graphical links or a jump link (cyclical string processing).

Syntax and Semantics Check during Programming

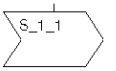
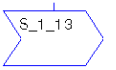
Introduction

A syntax and semantics check is carried out during program creation. The result of this check is displayed using color objects and text.

Representation

Representation of colors and labels:

Color	Description	Example
Black step symbol	Error free step	
Blue step symbol	Possible errors are: • Input pin not connected • Output pin not connected • Assigned action variable not declared • Assigned `SFCSTEP_TIMES` variable not declared	
Filled yellow step symbol (only Online)	Minimum supervision time (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) for the step below limit	

Color	Description	Example
Filled magenta step symbol (only Online)	Maximum supervision time (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) for the step exceeded	
Black transition symbol	Error free transition variable	
Black transition symbol	Error free transition section	
Blue transition symbol	Possible errors are: • Input pin not connected • Output pin not connected	
Red wavy line	Assigned transition variable not declared	
Black jump symbol	Error free jump	
Blue jump symbol	Possible errors are: • Input pin not connected • Jump target not defined • Invalid jump target	

Quickinfo

If the cursor is placed over erroneous text or object, a Quickinfo (Tooltip) with a brief description of the cause of the error appears.

Navigating with the keyboard

Using the keyboard

The following keys and key combinations are provided for navigation:

Key combinations	Motion
Cursor left	Moves the cursor to the left
Cursor right	Moves the cursor to the right
Cursor up	Moves the cursor up
Cursor down	Moves the cursor down
Ctrl+Cursor left	Moves the cursor one cell left
Ctrl+Cursor right	Moves the cursor one cell right
Ctrl+Cursor up	Moves the cursor one cell up
Ctrl+Cursor down	Moves the cursor one cell down
Home	Moves the cursor to the first displayed pixel of the current line
End	Moves the cursor to the last displayed pixel of the current line
Ctrl+Home	Shows the top left cell of the section (the position of the cursor is not affected by this function)
Ctrl+End	Shows the bottom right cell of the section (the position of the cursor is not affected by this function)
Page Up	Scrolls one page up (the position of the cursor is not affected by this function)
Page Down	Scrolls one page down (the position of the cursor is not affected by this function)
Ctrl+Page Up	Scrolls one page left (the position of the cursor is not affected by this function)
Ctrl+Page Down	Scrolls one page to the right (the position of the cursor is not affected by this function)
Ctrl+Alt+Page Up	Displays the previous section (including via the **View > Previous Section**menu).
Ctrl+Alt+Page Down	Displays the next section (including via the **View > Next Section**menu).
Spacebar	Selects the objects in the cell where the cursor is currently positioned.
Alt+Enter	Opens the properties dialog box for the selected object.

Selecting Objects

Select and Placement Mode

Selecting objects occurs in select mode.

Select mode can be activated by:

- Using the **Edit > Select Mode** menu command
- Using the  symbol

  or
- Pressing the **Esc** key

The  cursor symbol indicates that select mode is active.

Selecting an Object

Using the mouse	Using the keyboard
Left-click the object you want to select.	1. Move the cursor to the object to be selected, see also Navigating with the keyboard, page 895. 2. Press the **spacebar**.

Selecting Several Objects

Using the mouse	Using the keyboard
1. Click the left mouse button and keep it pressed. 2. Drag the mouse across the objects you want to select. or 1. Left-click the first object you want to select. 2. Press the **Ctrl** key and keep it pressed. 3. Left-click the next object you want to select. 4. Repeat these steps until the desired objects are selected.	1. Move the cursor to the first object to be selected, see also Navigating with the keyboard, page 895. 2. Press the **spacebar**. 3. Move the cursor to the next object to be selected 4. Press the **Ctrl**+**spacebar** key combination. 5. Repeat these steps until the desired objects are selected.

Selecting by Rows

Using the mouse	Using the keyboard
Selecting the contents of a row: 1. Left-click the number of the row you want to select in the vertical ruler. Selecting the contents of several rows: 1. Left-click the number of the first row you want to select in the vertical ruler. 2. Press the **Shift** key and keep it pressed. 3. Left-click the number of the last row you want to select in the vertical ruler. **Note:** Pressing the **Shift** key makes it possible to combine this procedure with the procedure for selecting by columns.	-

Selecting by Columns

Using the mouse	Using the keyboard
Selecting the contents of a column: 1. Left-click the number of the column you want to select in the vertical ruler. Selecting the contents of several columns: 1. Left-click the number of the first column you want to select in the vertical ruler. 2. Press the **Shift** key and keep it pressed. 3. Left-click the number of the last column you want to select in the vertical ruler. **Note:** Pressing the **Shift** key makes it possible to combine this procedure with the procedure for selecting by rows.	-

Selecting All (Entire Contents of the Section)

Using the mouse	Using the keyboard
Use the **Edit** > **Select All** menu command.	Press the **Ctrl**+**A** key combination.

Deselecting Objects

Using the mouse	Using the keyboard
Left-click an empty space in the section.	1. Move the cursor to an empty space in the section, see also Navigating with the keyboard, page 895. 2. Press the **spacebar**.

Undo and Redo Changes

Undo changes

Undo the last changes in the following ways:

- Use the **Edit > Undo** menu command.
- Press the **Ctrl**+**Z** key combination.

  or

- Select the  symbol.

Undo one change each time the function is carried out.

The function can be executed a maximum of 10 times.

If steps or transitions have been deleted, "Undo" means that the step and transition are restored, including the information assigned to them.

If a deleted step/deleted transition has an action/transition section assigned to it, only the reference to the action/transition section is restored and not the action/transition section itself.

Redo changes

Using **Redo**, you can redo changes that have been made:

- Use the menu command **Edit > Redo**.
- Press the **Ctrl**+**Y** key combination.

  or

- Select the  symbol.

Redo one undone action each time the function is carried out.

The function can be executed a maximum of 10 times.

Limits

After any of the following actions has been carried out**Undo** and **Redo** cannot be used.

- Deleting a step or a macro step in online mode.
- Toggling between **Normal View** and **Advanced View** and vice versa.

Delete, Cut, Copy, Paste, and Move Objects

Delete Objects

Using the mouse	Using the keyboard
1. Select an object to be deleted, page 897. 2. Use the **Edit** > **Delete** menu command.	1. Select an object to be deleted, page 897. 2. Press the **Entf** key.

Result:

- The selected object (and its graphical link when available) is deleted from the section.
- If action sections are assigned to a step (in offline mode), you are asked to confirm the deletion of the action sections. (In Online mode no confirmation is sought and the action sections remain until they are explicitly deleted.)
- If a transition section is assigned to a transition, (in offline mode) you are asked to confirm the deletion of this transition section. (In online mode no confirmation is sought and the transition section remains until it is explicitly deleted.)
- (If a transition variable is assigned in transition it remains until it is explicitly deleted.)
- If steps or transitions are to be deleted with step variables that are used in the program (including on the operating screen or in animation tables),
  - a message appears in the online mode that the step variables are used in the program and can only be deleted in the offline mode. If the question whether the operation it to be continued is confirmed with **Yes**, the mode changes to offline and the object is deleted. **No** terminates the operation.
  - An on-screen message in offline mode reports that step variables are used in the program and prompts you to confirm the operation:
    - **Yes:** Delete the object.
    - **No:** End the operation.
- When deleting a macro step, the associated macro step section is moved to the **Unused macros** directory for the active SFC section.
- If a step or macro step is deleted in online mode it cannot be undone using **Undo**.

NOTE: To delete entire rows or columns from the section, select the lines or columns in the line box, see also Deleting Rows and Columns, page 907.

Cut Objects

Using the mouse	Using the keyboard
1. Select an object to be cut, page 897. 2. Use the **Edit > Cut** menu command. or Use the menu command **Cut** in the shortcut menu (right mouse button). or Click the  symbol.	1. Select an object to be cut, page 897. 2. Press the **Ctrl**+**X** key combination.

Result:

- The selected object (and its graphical link when available) is deleted from the section.
- If action sections are assigned to a step, (in offline mode) you are asked to confirm the deletion of the action sections. (In Online mode no confirmation is sought and the action section remains until it is explicitly deleted.)
- If a transition section is assigned to a transition, (in offline mode) you are asked to confirm the deletion of this transition section. (In online mode no confirmation is sought and the transition section remains until it is explicitly deleted.)
- (If a transition variable is assigned in transition it remains until it is explicitly deleted.)
- If steps or transitions are to be deleted with step variables that will be used in the program (including on the operating screen or in animation tables),
  - An on-screen message in offline mode reports that step variables are used in the program and prompts you to confirm the operation:
    - **Yes:** Delete the object.
    - **No:** End the operation.
  - An on-screen message in online mode reports that **Create project** is available only in offline mode.
- When deleting a macro step, the associated macro step section is moved to the **Unused macros** directory for the active SFC section.
- If a step or macro step is deleted in online mode it cannot be undone using **Undo**.

The cut object can be inserted, page 903 in another position or in a different SFC section.

Copy Objects to the Clipboard

Using the mouse	Using the keyboard
1. Select an object to be copied, page 897. 2. Use the **Edit > Copy** menu command. or Use the **Copy** menu command from the shortcut menu (right-click). or Click the  symbol.	1. Select an object to be copied, page 897. 2. Press the **Ctrl+C** key combination.
Result: • The selected object and associated information is copied to the clipboard. • Graphical links can only be copied if they and their partner objects are explicitly selected. • If action sections are assigned to a step, only the reference to the action section is copied to the clipboard, not the action section itself. • If transition sections are assigned to a transition, only the reference to the transition section is copied to the clipboard, not the transition section itself. • When copying macro steps, only the macro step is copied and not its macro step section.	

The copied object can be inserted, page 903 in another position or in a different SFC section.

Paste Objects from the Clipboard

Using the mouse	Using the keyboard
1. Use the menu command **Edit > Insert**. or Use the menu command **Paste** in the shortcut menu (right mouse button). or Click the  symbol. 2. Left-click the target position.	1. Press the **Ctrl+V** key combination. 2. Move the cursor to the target position using the arrow keys. 3. Press **Enter**.

Result:

- The objects contained in the clipboard and their associated information are inserted into the section.
- Graphical links can only be inserted if they and their partner objects have been explicitly copied to the clipboard.
- If the objects to be inserted are placed on a graphical link, the link is undone and the objects are inserted into the structure.
- When inserting steps and macro steps they are automatically assigned a unique name.
- If an action section is assigned to a step, only the reference to the action section is inserted into the section together with the step and not the action section itself.

  If an action section is not declared in the active section, the action with the action section is deleted when it is inserted.
- If a transition section is assigned to a transition, only the reference to the transition section is inserted into the section together with the transition and not the transition section itself.

  A transition section may only be referenced once in a project. That means if a reference is available for a transition during insertion, only the transition object is inserted (without a reference).
- When inserting macro steps, only the macro step is inserted and not its macro step section. A new macro step section is created instead. The names of the steps within this macro step section are automatically matched to the (new) macro step names.

Move Objects with Drag and Drop

Using the mouse	Using the keyboard
1. Select an object to be moved, page 897. 2. Position the mouse pointer on the selected object. (on one of the selected objects if several are selected). **Result:** The mouse pointer changes its symbol to  . 3. Click the left mouse button and keep it pressed. **Result:** The mouse pointer changes its symbol to  . 4. Drag the object to the new position and release the mouse button.	1. Position the cursor on an object to be moved, page 897. 2. Press the **spacebar**. 3. Press the **Shift** key and keep it pressed. 4. Move the object to the target position using the arrow keys.

Result:

- The selected object is moved from its original position to the target position.
- Graphical links can only be moved if they and their partner objects are explicitly selected.
- If the objects to be moved are placed on a graphical link, the link is undone and the objects are inserted into the structure.
- Graphical links can only be kept if they and their partner objects are explicitly selected.

NOTE:

- The move operation is also possible across sections between different opened SFC sections.
- When objects are moved beyond the visible window of the editor window, automatic scrolling of the window does not begin until you reach the edge of the window with the mouse pointer, not as soon as the objects are moved beyond the edge.

Copy Objects by Moving with Drag and Drop

Using the mouse

1. Select an object to be copied, page 897.
2. Position the mouse pointer on the selected object.

   Result: The mouse pointer changes its symbol to  .
3. Click the left mouse button and keep it pressed.
4. Press the **Ctrl** key and keep it pressed.

   Result: The mouse pointer changes its symbol to  .
5. Drag the object to the target position.
6. Release the mouse button.

Result:

- The copy of the selected object and its associated information is inserted into the target position.
- Graphical links can only be copied if they and their partner objects are explicitly selected.
- If the objects to be copied are placed on a graphical link, the link is undone and the objects are inserted into the structure.
- When copying steps and macro steps they are automatically assigned a unique name.
- If an action section is assigned to a step, only the reference to the action section is inserted into the section together with the step and not the action section itself.

  If an action section is not declared in the active section, the action with the action section is deleted when it is inserted.
- If a transition section is assigned to a transition, only the reference to the transition section is inserted into the section together with the transition and not the transition section itself.

  A transition section may only be referenced once in a project. That means if a reference is available for a transition section during insertion, only the transition object is inserted (without a reference).
- When copying macro steps, only the macro step is copied and not its macro step section. A new macro step section is created instead. The names of the steps within this macro step section are automatically matched to the (new) macro step names.

Inserting Rows and Columns

Rules for Inserting Rows

The following rules apply when inserting rows:

- The total number of rows in an SFC section cannot be changed.

  However, the objects of an SFC section can be moved downwards by rows within the available section and therefore empty rows can be inserted into the string.
- The same number of rows is inserted as selected in the vertical ruler.

- As many rows can be inserted as there are empty rows at the end of the section.
- The new rows are inserted above the first selected row (i.e. all objects of the selected rows are moved down).
- When rows are inserted within existing links, these links are kept and are adjusted to the new position of the objects.

Inserting Rows

Step	Action
1	Select, page 898 the desired number of rows in the vertical ruler (click the row numbers).
2	Use the **Insert cells** menu command from the shortcut menu of the vertical ruler or the **Edit > Row operation > Insert cells** command. **Result:** The number of selected rows is inserted above the first selected row.

Rules for Inserting Columns

The following rules apply when inserting columns:

- The total number of columns in an SFC section cannot be changed.

  However, the objects of an SFC section can be moved right by columns within the available section and therefore empty columns can be inserted into the string.
- The same number of columns is inserted as is selected in the horizontal ruler.
- As many columns can be inserted as there are empty columns at the right border of the section.
- The new columns are inserted above the first selected column (i.e. all objects of the selected columns are moved right).
- When columns are inserted within existing branches or joints, the affected branches or joints are expanded by the corresponding number of columns.

Inserting Columns

Step	Action
1	Select, page 898 the desired number of columns in the horizontal ruler (click the row numbers).
2	Use the **Insert cells** menu command from the shortcut menu of the horizontal ruler or the **Edit > Row operation > Insert cells** command. **Result:** The number of selected columns is inserted in front of the first selected column.

Deleting Rows and Columns

Rules for Deleting Rows

The following rules apply when deleting rows:

- The total number of rows in an SFC section cannot be changed.

  However, the objects of an SFC section can be deleted by rows and the objects of the following rows can be moved up.
- The same number of rows are deleted as are selected in the vertical ruler.
- The objects of the selected row(s) are deleted and the objects of the following rows are moved up by the respective number of rows.
- If rows are deleted within existing links, these links are kept and are adjusted to the new position of the objects.

Deleting Rows

Step	Action
1	Select, page 898 the desired number of rows in the vertical ruler (click the row numbers).
2	You have the following options to delete the selected rows: • Select the menu command **Move and delete cells** in the rows shortcut menu, • execute the command **Edit > Vertical ruler > Move and delete cells**, or • Press the **Del** key. **Result:** The selected number of rows is deleted.

Rules for Deleting Columns

The following rules apply when deleting columns:

- The total number of columns in an SFC section cannot be changed.

  However, the objects of an SFC section can be deleted by columns and the objects of the following rows can be to the left.
- The same number of columns are deleted as are selected in the horizontal ruler.
- The objects of the selected column(s) are deleted and the objects of the following columns are moved to the left by the respective number of columns.
- If columns are deleted within existing branches or joints, the affected branches or joints are contracted by the respective number of columns.

Deleting Columns

Step	Action
1	Select, page 898 the desired number of columns in the horizontal ruler (click the row numbers).
2	You have the following options to delete the selected columns: • Select the menu command **Move and delete cells** in the rows shortcut menu or, • execute the command **Edit > Vertical ruler > Move and delete cells**, or • Press the **Del** key. **Result:** The selected number of columns is deleted.

Displaying the object properties

Displaying the object properties

You have the following options to display the object properties dialog box:

- Double-click the object.
- Select, page 897 the object and use the **Edit > Properties... dialog box.**.
- Select, page 897 the object and use the **Properties...command.**from the shortcut menu.
- Select, page 897 the object and then press the **Alt**+**Enter** key combination.

When the properties dialog box is called without an object being selected, the section properties dialog box , page 513 is opened.

If several objects are selected, the commands do not work.

View data properties

You have the following options to display the section properties dialog box, page 1202:

- using the data editor, page 348

1. Selecting one or several lines in a list editor
2. Using the shortcut menu select the **Properties**command.

- via the SFC section

1. Selecting one or several transitions in the SFC section.
2. From the context menu select the command **Data properties** or press **Ctrl** + **Enter**.

Initialize Search

Introduction

The **Initialize Search** function transfers the name of the selected element (search text) to the **Element** text box in the **Cross-References** window.

Selecting the search text

Process the following steps to select the search text:

Step	Action
1	Select the element to search for (search text). The following elements can be searched for: • step variables (To search for a step variable, select the step symbol.) • transition variables (To search for a transition variable, select the transition symbol.)
2	Use the **Services > Initialize Search** menu command. or Select the menu command **Initialize search** from the sections shortcut menu. or Press the **Ctrl+U** key combination. **Result:** The search text is transferred to the **Element** text box in the **Cross-References** window.

Initialize Animation Table

Introduction

The **Initialize Animation Table** transfers the selected variables to the Animation table, page 1481.

Selecting the variables

Process the following steps to select variables:

Step	Action
1	Select the variables to transfer to the Animation table. The following items can be selected: • step variables (To transfer step variables, select the step symbol.) • transition variables (To accept a transition variable, select the transition symbol.)
2	Use the **Services > Initialize Animation table** menu command. or Select the menu command **Animation Table** from the sections context menu. or Press the **Ctrl**+**T** key combination. **Result:** The variable(s) is/are transferred to the Animation table, page 1481.

Initialize Animation Table

Using **Initialize Animation Table** (**Ctrl**+**T**) you create an animation table with a default name e.g. `Table[SFC Editor - Mixer : [MAST]]`. You can rename the table.

All the variables you selected are listed in the animation table.

Selecting a new set of variables and using **Initialize Animation Table** again, all new variables are added to this animation table (if you did not rename it).

Initialize New Animation Table

Using **Initialize New Animation Table** (**Ctrl**+**Shift**+**T**) you create a new animation table with an indexed default name e.g. `Table[SFC Editor - Mixer : [MAST]1]`. You can rename the table.

All the variables you selected are listed in the table.

But if you click anywhere in the section and do not select any variable, you create a new empty animation table.

Every time you use **Initialize New Animation Table** you create a new animation table.

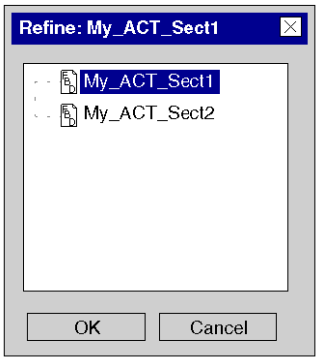
Refining steps, macro steps and transitions

Introduction

This function makes it possible to "look into" a macro step or a transition (section). This makes it possible to see the internal state of the macro/transition section when animation is used, or to make changes to the logic very quickly.

Refining steps

Process the following steps to refine steps:

Step	Action
1	Select the step.
2	You have the following options to display to refine steps: • Use the **Services > Refine** menu command. • Use the **Refine** menu command from the shortcut menu. or • Press the **Ctrl+Q** key combination. **Result:** If the step contains only one action section, this will be opened automatically. If the step contains several action sections, a selection dialog box is opened. Example: 
3	Select the desired action section. **Result:** The action section is opened.

Refining macro steps

Process the following steps to refine macro steps:

Step	Action
1	Select the macro step.
2	You have the following options to refine macro steps: • Use the **Services > Refine** menu command. • Use the **Refine** menu command from the shortcut menu. or • Press the **Ctrl**+**Q** key combination. **Result:** The macro section is opened. **Note:** With nested macro steps several **refine** processes are possible.

Refining transitions

Process the following steps to refine transitions:

Step	Action
1	Select the transition.
2	You have the following options to refine transitions: • Use the **Services > Refine** menu command. • Use the **Refine** menu command from the shortcut menu. or • Press the **Ctrl**+**Q** key combination. **Result:** The transition section is opened.

Go to

Introduction

The **Go to** function is used to jump to:

- a specific location in the current section,
- a specific bookmark in the current section or
- a specific step or macro step in the current section or
- a specific jump in the current section.

Open the dialog

The following options are available to open the**Go to** dialog box.

- Use the menu command **Edit > Go to**,
- select the menu command **Go to** from the sections shortcut menu,
- Press the key combination **Ctrl**+**G**.

  or

- Choose the  symbol.

Jump to a Position

Go to a certain location in the current section by carrying out the following steps:

Step	Action
1	Open, page 913 the **Go to** dialog box.
2	Select the **Position** tab.
3	Enter the line and/or column number.
4	Confirm the entry using the command button **Go to**. **Result:** The desired position is displayed.

Jump to Bookmark

Go to a certain bookmark in the current section by carrying out the following steps:

Step	Action
1	Open, page 913 the **Go to** dialog box.
2	Choose the**Bookmark** tab.
3	Select a bookmark in the list.
4	Double click on the selected bookmark or press the **Go to** command button. **Result:** The bookmark is shown and marked as the current bookmark, page 615.

Jump to a Step

Jump to a certain step or macro step in the current section by carrying out the following steps:

Step	Action
1	Open, page 913 the **Go to** dialog box.
2	Select the **Step** tab.
3	Select a step or a macro step in the list.
4	Double click on the selected step or macro step or press the **Go to** command button. **Result:** The desired step is displayed.

Jump to a Jump

Go to a certain jump in the current section by carrying out the following steps:

Step	Action
1	Open, page 913 the **Go to** dialog box.
2	Select the **Jump** tab.
3	Select a jump in the list.
4	Double click on the selected jump or press the **Go to** command button. **Result:** The desired jump is displayed.

Go to dialog

Call the dialog box:

Refer to the Opening the dialog, page 913

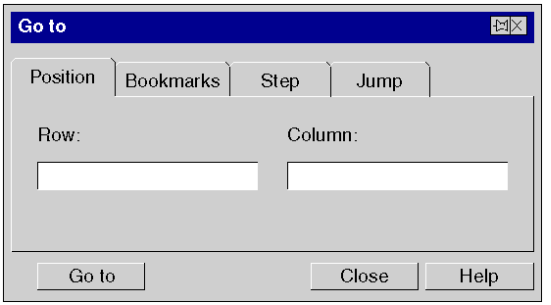
Call the dialog box

The **Go to** dialog box is made up of three tabs:

- **Location**
  Go to a certain position in the current section using this tab:
- **Bookmarks**
  Go to a certain bookmark in the current section using this tab:
- **Step**
  This tab is used to a jump to a certain step or macro step in the current section
- **Jump**
  Go to a certain jump in the current section using this tab:

Position tab

Representation of the tab **Position**:

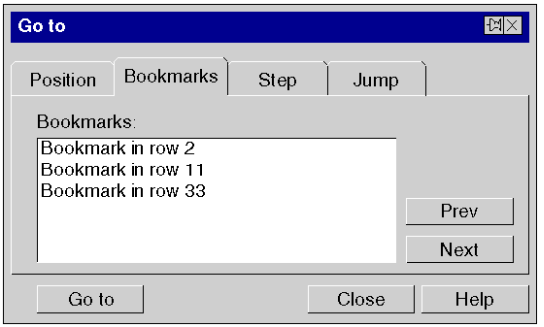


Elements of the **Position** tab

Element	Description
	For symbols that are not printed, the dialog box is closed after every search. For printed symbols the dialog box remains open until you close it.
Row	The number of the row to be displayed can be entered in this text box.
Column	The number of the column to be displayed can be entered in this text box.
Go to	Display the current position using the button desired position, page 913.
Close	Use this button to close the dialog.
Help	Use this button to call up the dialog help.

Register tab Bookmarks

Representation of the**Bookmark** tab:

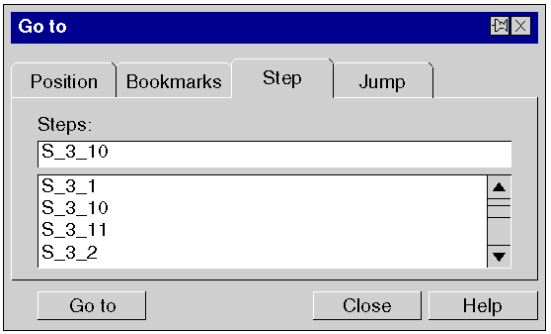


Elements of the tab **Bookmark**:

Element	Description
	For symbols that are not printed, the dialog box is closed after every search. For printed symbols the dialog box remains open until you close it.
Bookmark:	Select the bookmark to be displayed from this list box.
Prev	If a current bookmark, page 615 is selected, a jump is made to the previous one and the last one is marked as the current bookmark. If no current bookmark is selected, the margin is search from the bottom to the top for the next bookmark, which is marked as the current bookmark. When the start of the section is reached, the search starts again from the bottom.
Next	If a current bookmark, page 615 is selected, a jump is made to the previous one and the last one is marked as the next bookmark. If no current bookmark is selected, the margin is search from top to bottom for the next bookmark, which is marked as the current bookmark. When the end of the section is reached, the search starts again from the top.
Go to	Display the current position using the button desired bookmark, page 913.
Close	Use this button to close the dialog.
Help	Use this button to call up the dialog help.

tab Step

Display of the Register **Step**:

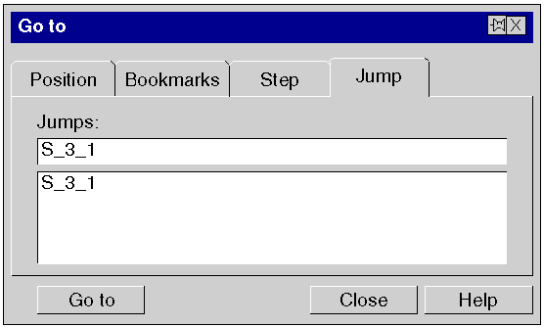


Elements of the register tab **Step**:

Element	Description
	For symbols that are not printed, the dialog box is closed after every search. For printed symbols the dialog box remains open until you close it.
Steps:	You can enter the names of the steps and macro steps to be shown in this text box. You can enter the name of the step or macro step in the text box with the keyboard or using the drop-down list by clicking with the mouse.
Go to	Display the desired step or macro step, page 914 using the button.
Close	Use this button to close the dialog.
Help	Use this button to call up the dialog help.

Tab Jump

Display of the Register **Jump**:



Elements in the **Jump**tab:

Element	Description
	For symbols that are not printed, the dialog box is closed after every search. For printed symbols the dialog box remains open until you close it.
Locations:	You can enter the names of the jump to be shown in this text box. You can enter the name of the jump in the text box with the keyboard or using the drop-down list by clicking with the mouse.
Go to	Display the current position using the button desired jump, page 914.
Close	Use this button to close the dialog.
Help	Use this button to call up the dialog help.

List and display jumps

Introduction

The **Display jumps** function is used to list all jumps referencing this step and display a selected position of the current section.

Open the dialog

To open the dialog, select the jump and execute the **Display jumps** command in the context menu of the jump.

List in the dialog

Since the jumps do not have their own names, the dialog lists the positions of the jumps and names of their preceding steps.

Display a selected position

Follow the steps below to display a selected position in the current section.

Step	Action
1	Open the **Display jumps** dialog box:
2	Select a position from the list.
3	Confirm the selection using the command button **Go to**. **Result:** The desired position is displayed. **Note:** Instead of selecting the position and clicking on the button, you can also double-click on the position.

Go to Step

Introduction

The **Go to Step**function displays the referenced step of a jump.

Display the referenced step

Perform the steps below to display the referenced step of a jump.

Step	Action
1	Select a jump,
2	Open the context menu (right-click).
3	Click on **Go to Step**. **Result:** The referenced step is displayed.

General Information about Steps

Overview

This section provides general information about using steps in the SFC programming language.

Selecting steps

Introduction

An SFC section is a "status machine", i.e. the status is indicated by the active steps. Zero or more actions belong to every step.

Selecting steps

Selecting steps:

Step type	Description	Call via menu command	Call via symbol	Call via keyboard	Cursor symbol
"Normal" step, page 924	A step becomes active, when the upstream transition is satisfied, and is normally inactive when the downstream is satisfied. Zero or more actions belong to every step. Steps without action are known as waiting steps. See also section Step (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.	**Edit > New > Step** or **Step** menu command from the shortcut menu		**F3**	
Initial Step, page 925	The initial status of a sequence is characterized by the initial step, which is active when initializing the project containing the SFC section. Initial steps are not usually assigned an action. With single token (conforming to IEC 61131-3), only one initial step is permitted per sequence.	**Edit > New > Step** or **Step** menu command from the shortcut menu		**F3**	

Step type	Description	Call via menu command	Call via symbol	Call via keyboard	Cursor symbol
	With multi token, any number (0 up to all steps in the string) of initial steps are possible. See also section Step (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.				
Macro Step, page 937	Macro steps are used to call macro sections and therefore the hierarchical structure of sequence control. Macro sections calls are expansions to IEC 61131-3 and must be explicitly enabled via the **Tools > Project Settings** dialog box, in the **Language extensions** tab by activating the **Allow macro sections** check box. See also section Macro steps and macro sections (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.	**Edit > New > Macro Step** or **Macro Step** menu command from the shortcut menu		**Ctrl+F3**	

Step type	Description	Call via menu command	Call via symbol	Call via keyboard	Cursor symbol
InStep, page 947	Every macro section begins with an InStep. InSteps are created automatically in macro sections by the SFC editor and cannot be deleted, copied or inserted manually. See also section InStep (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.	-	-	-	-
OutStep, page 954	Every macro section ends with an OutStep. OutSteps are created automatically in macro sections by the SFC editor and cannot be deleted, copied or inserted manually. See also section OutStep (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.	-	-	-	-

Placing steps

Introduction

Steps can be placed in any free cell.

If a step is placed in a cell that is already occupied by an object, an error message is returned.

If a step is placed in a cell where the neighboring cell above or below already contains a step, an error message is returned because a transition must be placed between two steps (see also section Connection rules (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual).

A step which is placed always connects automatically with the neighboring objects above and below if there are no free cells between them.

Note

In the **Tools > Options** dialog box on the **Data and Languages** tab, if the **Automatically assign a variable to a new graphical object** check box is selected, then the corresponding properties dialog box is automatically opened when an object is placed. The procedures explained here are used when the check box is **cleared**.

Placing steps:

Placing steps:

Step	Action
1	Select the desired step, see also Selecting steps, page 920.
2	Click the target cell in the SFC section. or Use the **arrow keys** to move the cursor to the target position, and press **Enter**. **Result:** The selected step is inserted.
3	To place further steps of the same type: • Click the target cell in the SFC section. or Use the **arrow keys** to move the cursor to the target position, and press **Enter**. • To return to selection mode: Press the **Esc** key. • To insert other objects: Select the object you wish to insert.

Editing "normal" steps

Overview

This section describes how to edit "normal" steps in the SFC programming language.

Defining the properties of steps

Introduction

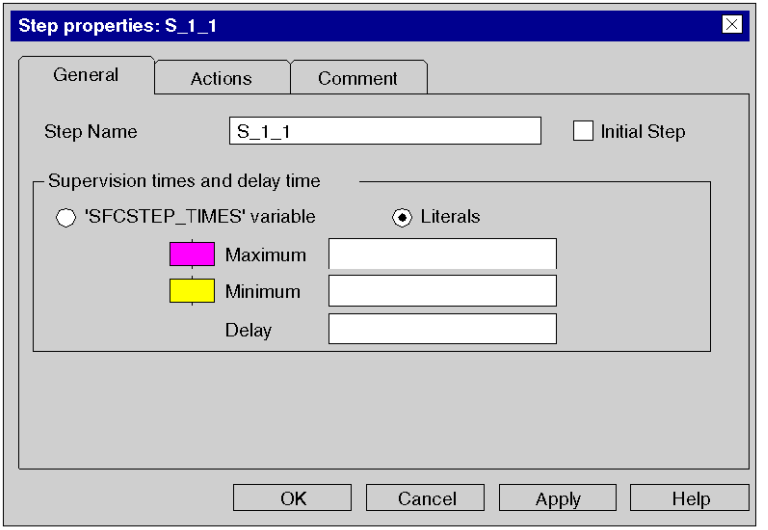
The following can be defined for steps:

- the step name, page 924
- whether the step is an initial step, page 925
- the supervision times and the delay time
  - as a literal, page 926
  - as a SFCSTEP_TIMES variable, page 927
- the actions assigned, page 958
- a comment, page 931

To find a specific jump in the current section use the **Go to** dialog box.

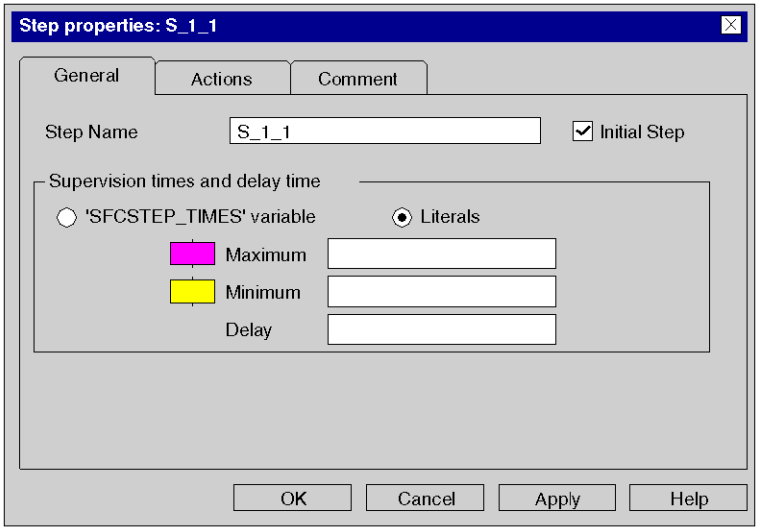
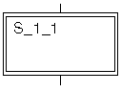
Defining the step name

Defining the step name:

Step	Action
1	Open the properties dialog box, page 908 for the step. **Result**: The step properties dialog box, page 931 is opened. Example: 
2	In order to change the name of the step, enter a step name in the **Step Name** text box, see also section Step names (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.
3	Confirm the step names with **OK**. **Result:** The selected step name is enabled, the properties dialog box is closed, and a syntax and semantics check, page 894 is performed.

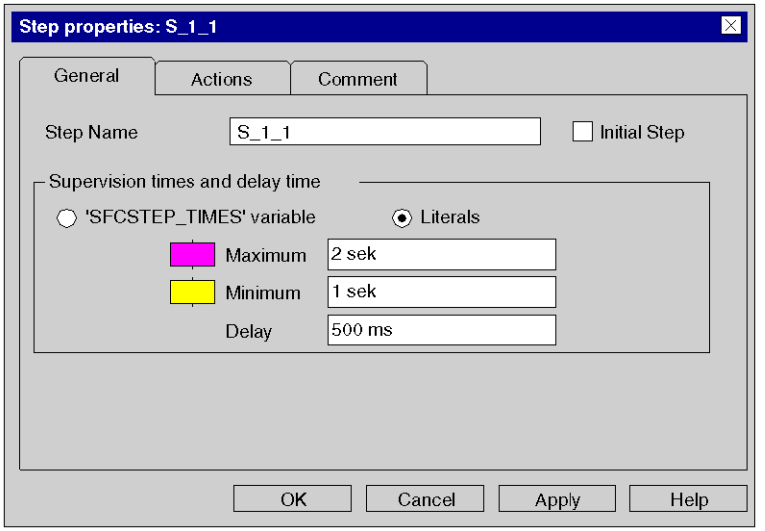
Defining an initial step

Defining an initial step:

Step	Action
1	Open the properties dialog box, page 908 for the step. **Result**: The step properties dialog box, page 931 is opened.
2	In order to define an initial step, activate the **Initial Step** check box, see also section Step types (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual. Example: 
3	Confirm the entry with **OK**. **Result:** The selected step is enabled in the section as an initial step, the properties dialog box is closed, and a syntax and semantics check, page 894 is performed. Example: 

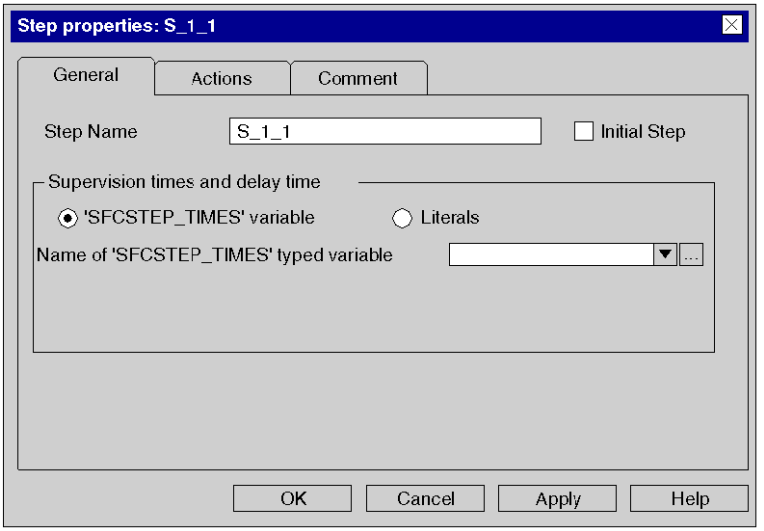
Defining the supervision time/delay time as a literal

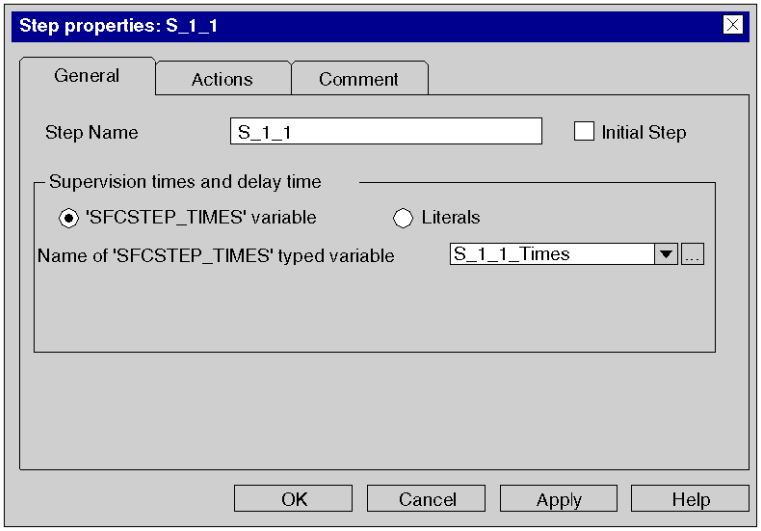
Defining the supervision time/delay time as a literal:

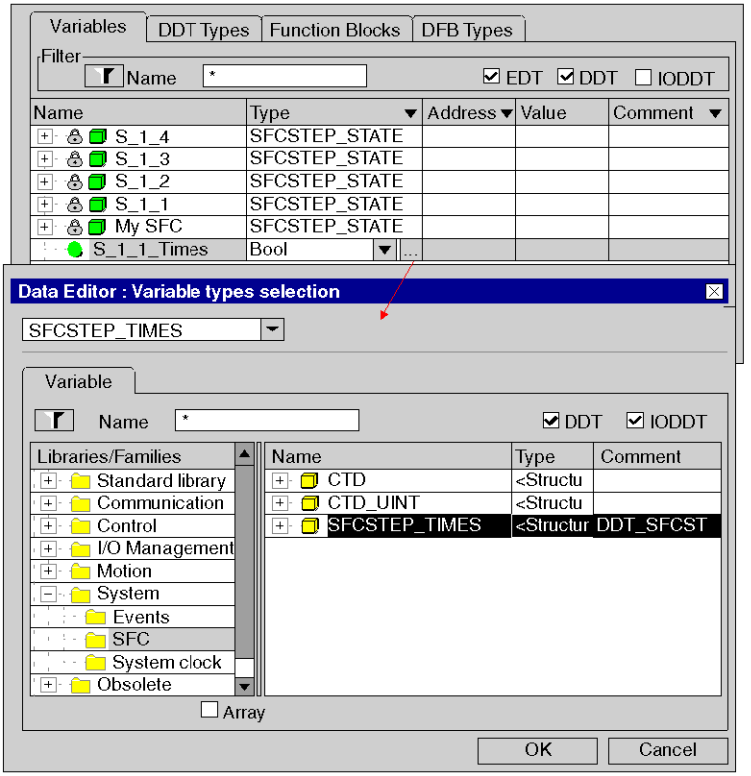
Step	Action
1	Open the properties dialog box, page 908 for the step. **Result**: The step properties dialog box, page 931 is opened.
2	In the **Supervision times and delay time** area, activate the **Literal** option button if it isn't already activated.
3	Enter a value for the minimum supervision time, the maximum supervision time and the delay time, see also section Step times (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual. Example: 
4	Confirm the entries with **OK**. **Result:** The step times are enabled, the properties dialog box is closed, and a syntax and semantics check, page 894 is performed.

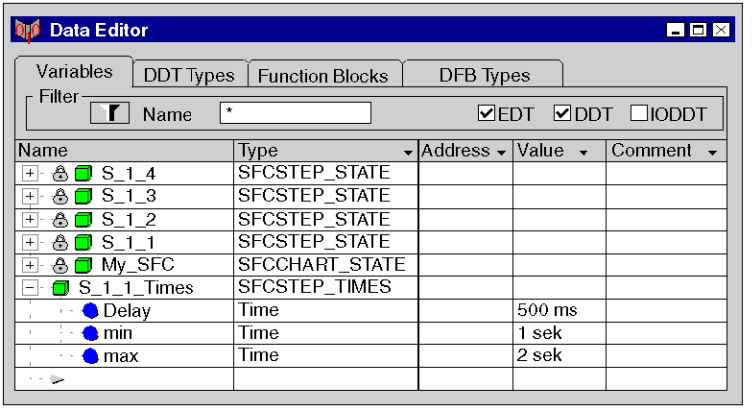
Defining the supervision time/delay time as an `SFCSTEP_TIMES` variable

Defining the supervision time/delay time as an `SFCSTEP_TIMES` variable:

Step	Action
1	Open the properties dialog box, page 908 for the step. **Result**: The step properties dialog box, page 931 is opened.
2	In the **Supervision times and delay time** area, activate the **'SFCSTEP_TIMES' variable** option button if it isn't already activated. Example: 
3	In the data selection box, page 1850, specify a variable of the `SFCSTEP_TIMES (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual)` data type. You have the following options to enter variables: • You can enter the name of the variable. • You can select the name of the variable from the list of recently used names using the ▼ symbol. or • Using the command button **...** you can open a variable selection dialog box, page 1181 and confirm the selected variables there with **OK**.

Step	Action
	Example: 
4	Confirm the entries with **OK**. **Result:** The selected `SFCSTEP_TIMES` variable is enabled, the properties dialog box is closed, and a syntax and semantics check, page 894 is performed. If the variable has already been declared (see also Derived variable declaration, page 394), the procedure ends here. If the `SFCSTEP_TIMES` variable has not yet been declared, process the following steps.

Step	Action
5	Declare a derived variable of the `SFCSTEP_TIMES` data type, see also Derived variable declaration, page 394. Example: 
6	Enter a value for the minimum supervision time, the maximum supervision time and the delay time, see also section Step times (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual. Example:

Step	Action
	 **Result:** The `SFCSTEP_TIMES` variable is declared and the step frame is displayed in black.

Enter a comment

Enter a comment:

Step	Action
1	Open the properties dialog box, page 908 for the step.
2	Select the **Comment** tab and enter a step comment.
3	Confirm the comment with **OK**. **Result:** The comment is accepted and the properties dialog box is closed.

Steps properties dialog box

Calling the properties dialog box

see Opening the properties dialog box, page 908

Structure of the properties dialog box

The steps properties dialog box consists of three tabs:

- **General**

  In this tab you can define:

  - the step name, page 924
  - whether it is an initial step, page 925
  - the step supervision times and delay time as a variable, page 927 or literal, page 926

- **Actions**

  In this tab you can define:

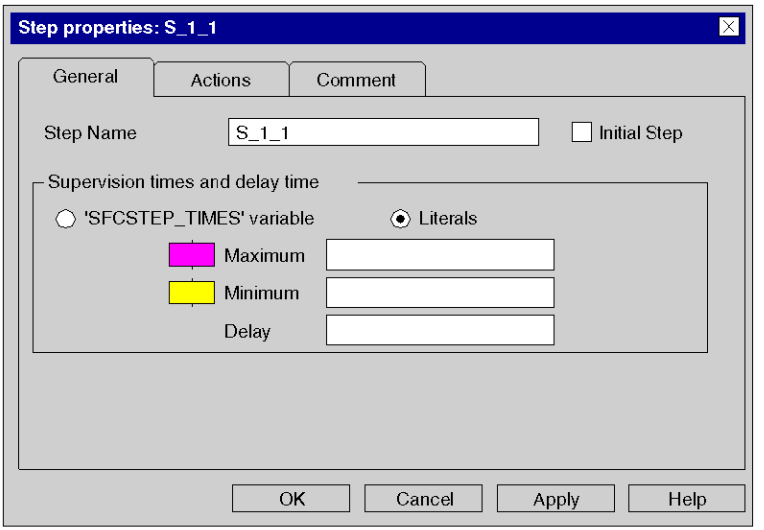
  - the actions assigned to the step as an action variable, page 961 or as an action section, page 963
  - the qualifier, page 964 allocated to the action
  - the time period for the qualifier, page 964
  - the execution order, page 966 of the actions

- **Comment**

  In this tab, you can enter a comment, page 931 about the step.

General tab

Representation of the **General** tab:

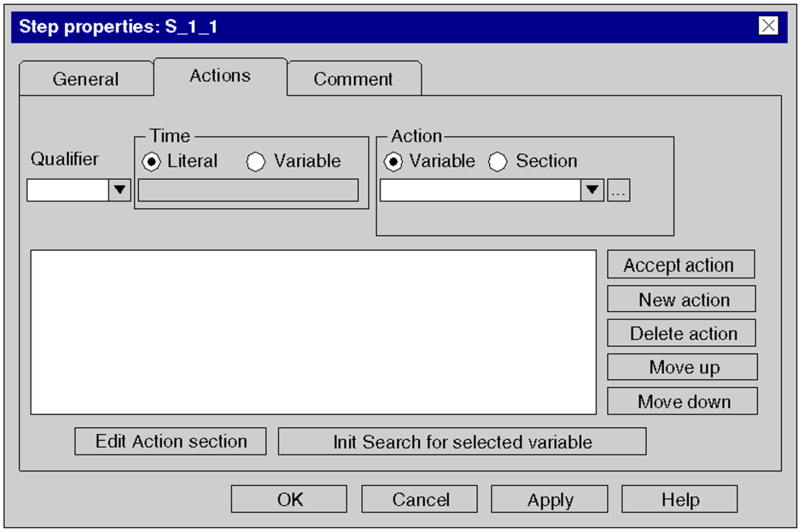


Elements of the **General** tab:

Element	Description
Step Name	Enter the step name, page 924 in this text box.
Initial Step	When this check box is activated, the current step becomes an initial step, page 925.
Supervision times and delay time	Define the step supervision times and delay time in this area, see also section Step times (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual. The contents of this area depends on the selected option: • **'SFCSTEP_TIMES' variable** When you activate this option button, you can enter the step supervision and delay times as variables, page 927 of the `SFCSTEP_TIME` data type in the **Name of 'SFCSTEP_TIMES' typed variable** data selection box, page 1850, see also section Step times (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual. • **Literals** When you activate this option button, you can enter the step supervision and delay times as literals, page 926, see also section Step times (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual. ◦ **Maximum** Enter the maximum supervision time for the current step in this text box. ◦ **Minimum** Enter the minimum supervision time for the current step in this text box. ◦ **Delay** Enter the delay time for the current step in this text box.
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Actions tab

Representation of the register tab **Actions**:



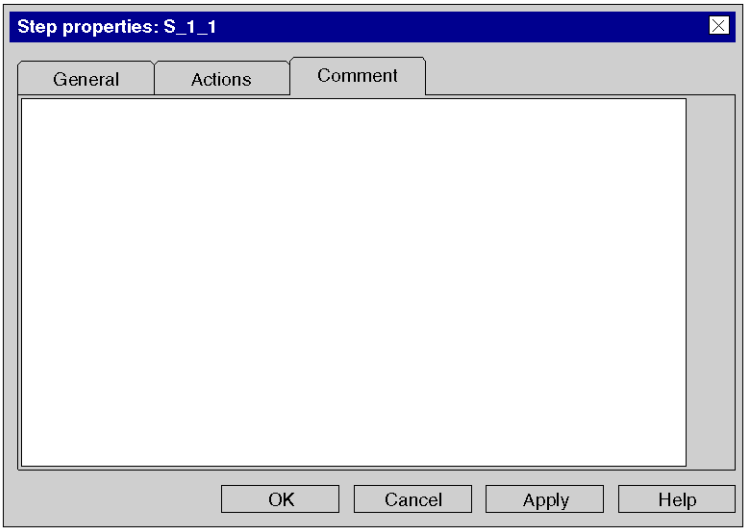
Elements of the **Actions** tab:

Element	Description
Qualifier	Select the qualifier, page 964 for the current action from this list box, see also section Qualifiers (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.
Time	Define the time period for the qualifiers, page 964 `L` (time limited), `D` (time delayed) or `DS` (delayed and saved) in this area. The contents of this area depends on the selected option: • **Literal** When you activate this option button, you can enter the time period as literals in the text box. • **Variable** When you activate this check box, you can enter the time period as a `TIME` data type in the data selection box, page 1850.
Action	Define the action in this area. The contents of this area depends on the selected option: • **Variable** When you activate this check box, you can enter the action as an action variable, page 961 in the data selection box, page 1850.

Element	Description
	• **Section** When you activate this check box, you can enter the action as an action section, page 963 in the list box.
Action list box	All actions of the current step are displayed in this list box.
Edit Action section	This command button is only available when an action section is selected in the action list box. Use this command button to close the properties dialog box and open the selected action section.
Init Search for selected variable	This command button is only available when an action variable is selected in the action list box. Use this command button to close the properties dialog box and open the selected action section. The name of the action variable is applied using this command button in the **Cross-references** window.
Accept action	Use this command button to replace the definitions of the selected action in the action list box with the modified definitions, page 960.
New action	Use this command button to insert the new action, page 958 into the action list box.
Delete action	Use this command button to delete the selected action in the action list box., page 966 **Note:** When deleting actions with action sections you are asked to confirm whether the action section should be deleted. Please note that deleted sections cannot be restored using **Edit > Undo**.
Move up	Use this command button to move the selected actions in the action list box one position up, see also Changing the execution order of actions, page 966.
Move down	Use this command button to move the selected actions in the action list box one position down, see also Changing the execution order of actions, page 966.
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Comment tab

Representation of the **Comment** tab:



Elements of the **Comment** tab:

Element	Description
Text box	Enter a comment about the step.
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Editing macro steps

Overview

This section describes how to edit macro steps in the SFC programming language.

Macro steps are expansions to IEC 61131-3 and must be explicitly enabled via the **Tools > Project Settings** dialog box, in the **Language extensions** tab by activating the **Allow macro sections** check box.

Defining the properties of macro steps

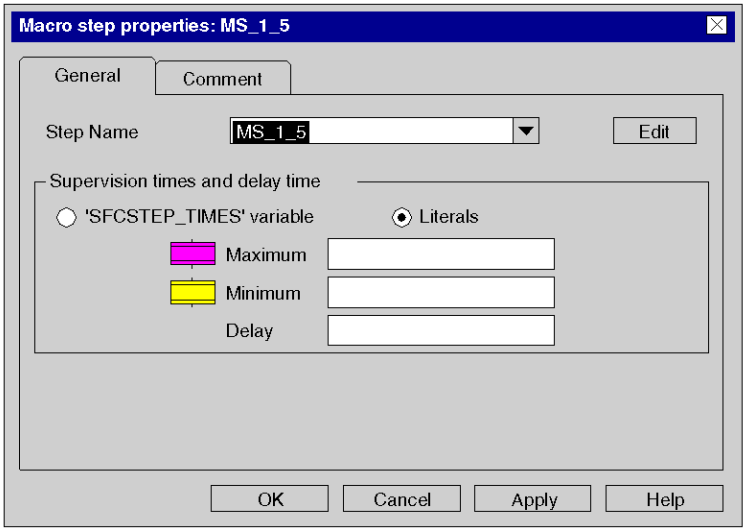
Introduction

The following can be defined for macro steps:

- the macro step name, page 937
- the logic of the macro section, page 938
- the supervision times and the delay time
  - as a literal, page 939
  - as a SFCSTEP_TIMES variable, page 940
- a comment, page 944

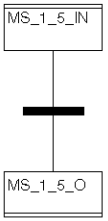
Defining the macro step name

Defining the macro step name:

Step	Action
1	Open the properties dialog box, page 908 for the macro step. **Result:** The macro step properties dialog box is opened, see also Macro Steps Properties Dialog Box, page 944. Example: 
2	In order to change the name of the step, enter a name in the **Step Name** text box, see also section Macro step names (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.
3	Confirm the macro step name with **OK**. **Result:** The macro step name selected is enabled for the macro step and the steps within the macro step section. The properties dialog box is closed and a syntax and semantics check, page 894 is performed.

Editing the macro section

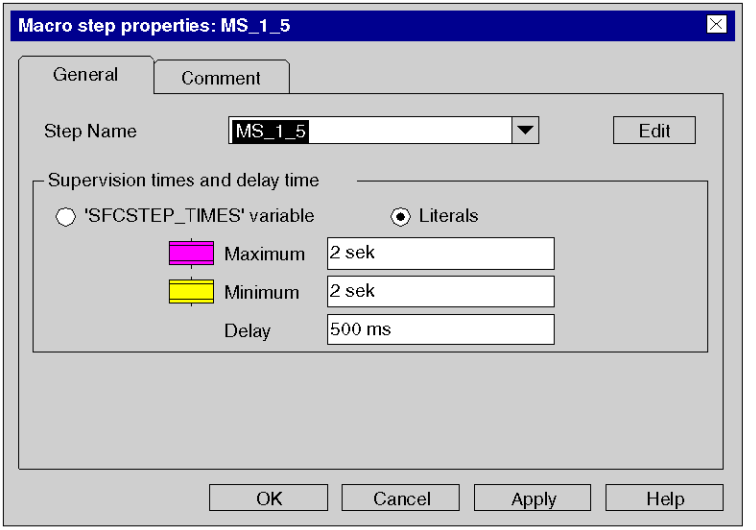
Editing the macro section:

Step	Action
1	Open the properties dialog box, page 908 of the macro step, and press the **Edit** command button. or Select the macro step and process the **Services > Refine** menu command. **Result:** The macro step properties dialog box is closed and the macro section of the macro step is opened. Example: 
2	Create the macro section logic, see also section Macro sections (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.

Defining the supervision time/delay time as a literal

Defining the supervision time/delay time as a literal:

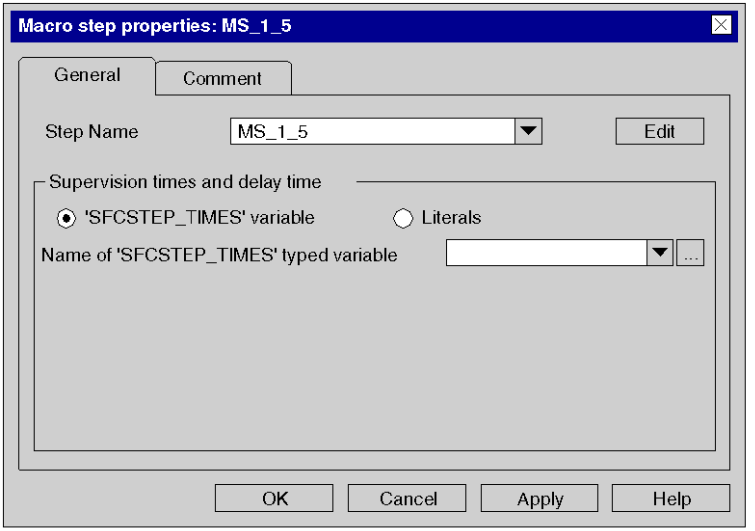
Step	Action
1	Open the properties dialog box, page 908 for the macro step. **Result:** The macro step properties dialog box is opened, see also Macro Steps Properties Dialog Box, page 944.
2	In the **Supervision times and delay time** area, activate the **Literal** option button if it isn't already activated.

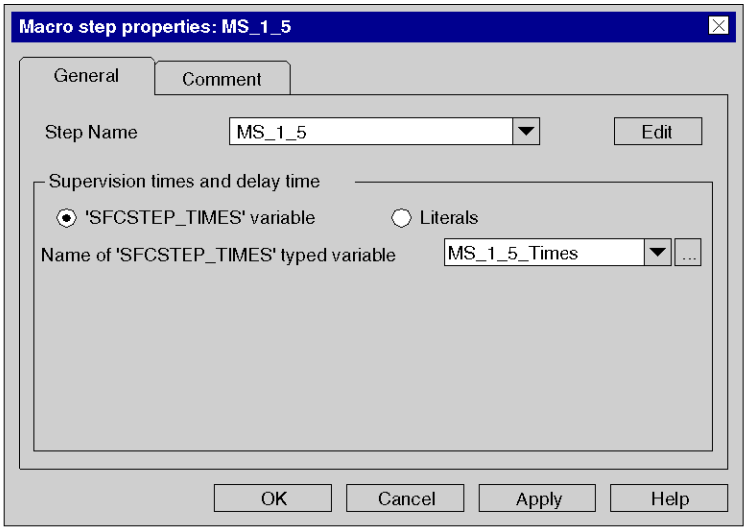
Step	Action
3	Enter a value for the minimum supervision time, the maximum supervision time and the delay time, see also section Step times (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual. Example: 
4	Confirm the entries with **OK**. **Result:** The step times are enabled, the properties dialog box is closed, and a syntax and semantics check, page 894 is performed.

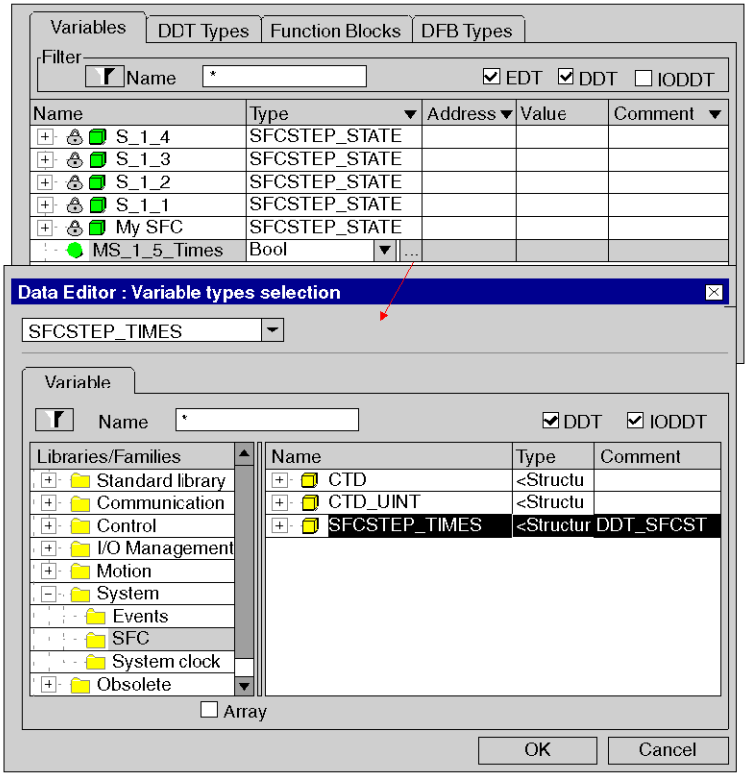
Defining the supervision/delay time as an `SFCSTEP_TIMES` variable.

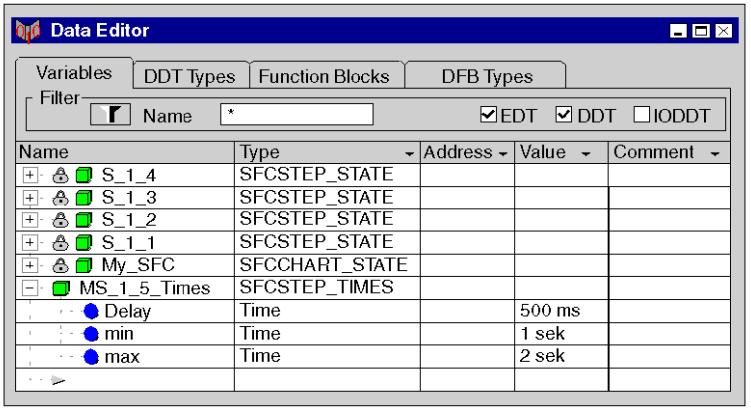
Defining the supervision time/delay time as an `SFCSTEP_TIMES` variable:

Step	Action
1	Open the properties dialog box, page 908 for the macro step. **Result:** The macro step properties dialog box is opened, see also Macro Steps Properties Dialog Box, page 944.
2	In the **Supervision times and delay time** area, activate the **'SFCSTEP_TIMES' variable** option button if it isn't already activated.

Step	Action
	Example: 
3	In the data selection box, page 1850, specify a variable of the `SFCSTEP_TIMES` data type, see also section SFCSTEP_TIMES (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual. You have the following options to enter variables: • You can enter the name of the variable. • You can select the name of the variable from the list of recently used names using the ▼ symbol. or • Using the command button **...** you can open a variable selection dialog box, page 1181 and confirm the selected variables there with **OK**.

Step	Action
	Example: 
4	Confirm the entries with **OK**. **Result:** The selected `SFCSTEP_TIMES` variable is enabled, the properties dialog box is closed, and a syntax and semantics check, page 894 is performed. If the variable has already been declared (see also Derived variable declaration, page 394), the procedure ends here. If the `SFCSTEP_TIMES` variable has not yet been declared, process the following steps.

Step	Action
5	Declare a derived variable of the `SFCSTEP_TIMES` data type, see also Derived variable declaration, page 394. Example: 
6	Enter a value for the minimum supervision time, the maximum supervision time and the delay time, see also section Step times (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual. Example:

Step	Action
	 **Result:** The `SFCSTEP_TIMES` variable is declared and the macro step frame is displayed in black.

Enter a comment

Enter a comment:

Step	Action
1	Open the properties dialog box, page 908 for the macro step.
2	Select the **Comment** tab and enter a macro step comment.
3	Confirm the comment with **OK**. **Result:** The comment is accepted and the properties dialog box is closed.

Macro Steps Properties Dialog Box

Calling the properties dialog box

see Displaying the object properties, page 908

Structure of the properties dialog box

The macro steps properties dialog box consists of 3 tabs:

- **General**

  In this tab you can define:

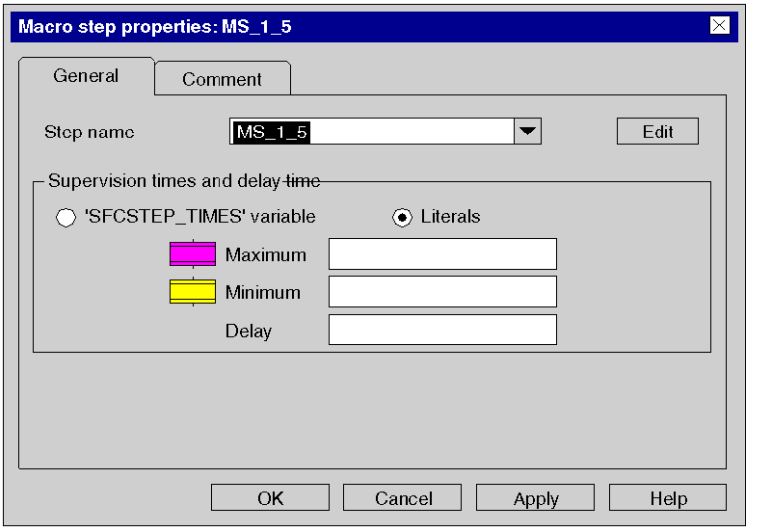
  - the macro step names, page 937
  - the step supervision time and delay time as a variable, page 940 or literal, page 939

- **Comment**

  A comment, page 944 about the step can be entered in this tab.

Tab General

Representation of the **General** tab:



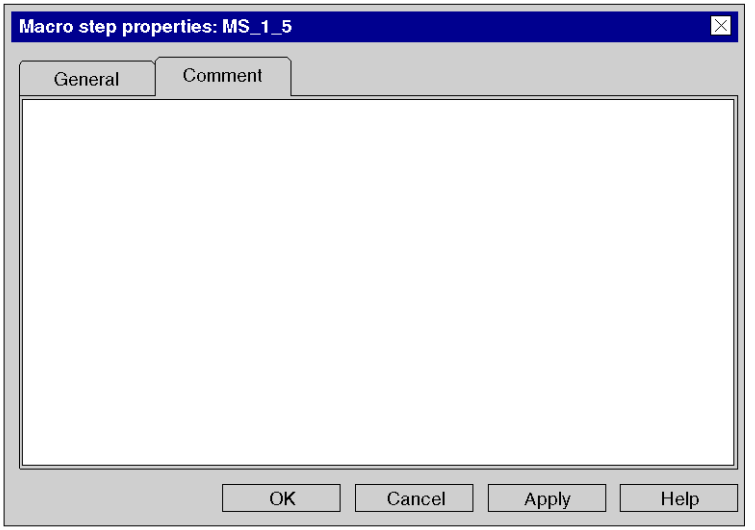
Items of the **General** tab:

Item	Description
Step name	Enter the macro step name, page 937 in this text box.
Edit	Use this command button to close the properties dialog box and open the macro step section.
Supervision times and delay time	You define the step supervision time and delay time in this range, see also section Step times (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual. The contents of this area depends on the option selected:

Item	Description
	• **'SFCSTEP_TIMES' variable** If you check this option button you can enter in the data selection box , page 1850 **Name of 'SFCSTEP_TIMES' typed variables** the Step supervision and delay times as variables, page 940 of the `SFCSTEP_TIME` data type. • **Literals** If you check this option button you can enter the Step supervision and delay times as literals, page 939. ◦ **Maximum** Enter the maximum supervision time for the current macro step in this text box. ◦ **Minimum** Enter the minimum supervision time for the current macro step in this text box. ◦ **Delay** Enter the delay time for the current macro step in this text box.
OK	This command button enables all entries and closes the dialog box.
Apply	This command button enables all entries without closing the properties dialog box.

Tab Comment

Representation of the **Actions** tab:



Items of the **Comment** tab:

Item	Description
Text box	Enter a comment about the step.
OK	This command button enables all entries and closes the dialog box.
Apply	This command button enables all entries without closing the properties dialog box.

Editing input steps

Overview

This section describes how to edit input steps in the SFC programming language.

Defining the properties of InSteps

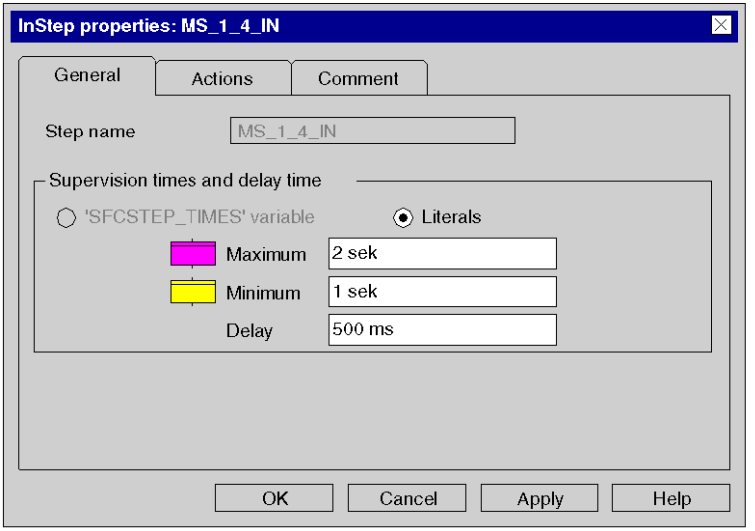
Introduction

The following can be defined for InSteps:

- the supervision times and the delay time, page 947
- the actions assigned, page 958
- a comment, page 948

Defining the supervision time/delay time

Defining the supervision time/delay time as a literal:

Step	Action
1	Open the properties dialog box, page 908 for the InStep. **Result**: The properties dialog box for the step is opened, see also InSteps properties dialog box, page 949.
2	Enter a value for the minimum supervision time, the maximum supervision time and the delay time, see also section Step times (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual. Example: 
3	Confirm the entries with **OK**. **Result:** The step times are enabled, the properties dialog box is closed, and a syntax and semantics check, page 894 is performed.

Entering a comment

Entering a comment:

Step	Action
1	Open the properties dialog box, page 908 for the InStep.
2	Select the **Comment** tab and enter a step comment.
3	Confirm the comment with **OK**. **Result:** The comment is enabled and the properties dialog box is closed.

InSteps properties dialog box

Calling the properties dialog box

see Displaying the object properties, page 908

Structure of the properties dialog box

The InStep properties dialog box consists of three tabs:

- **General**

  In this tab you can define:

  - the step supervision time and the delay time, page 947

- **Actions**

  In this tab you can define:

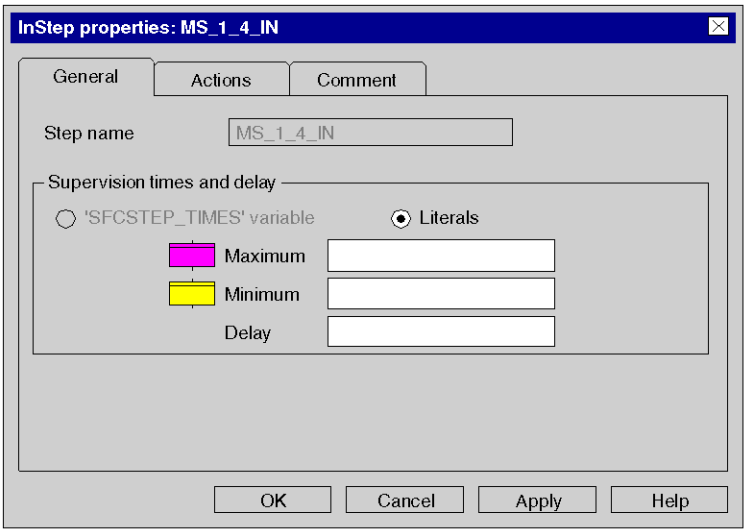
  - the actions assigned to the step as an action variable, page 961 or as an action section, page 963
  - the qualifier, page 964 allocated to the action
  - the time period for the qualifier, page 964
  - the execution order, page 966 of the actions

- **Comment**

  In this tab, comment, page 948 about the InStep can be entered.

General tab

Representation of the **General** tab:

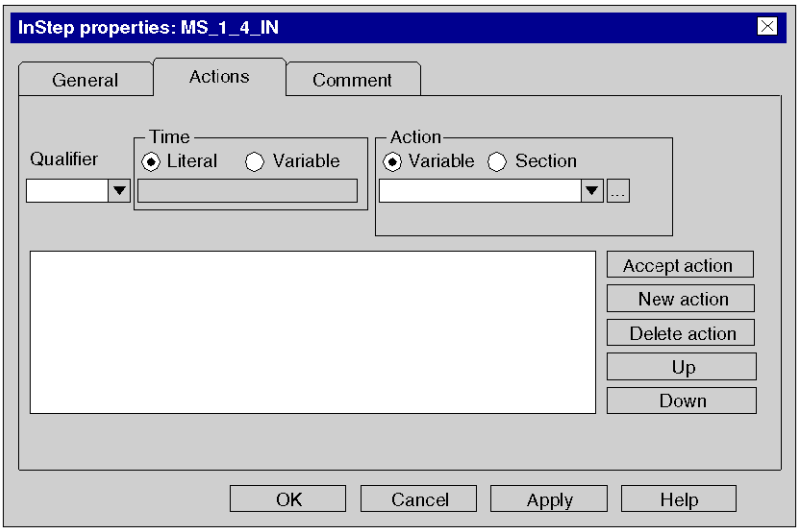


Elements of the **General** tab:

Element	Description
Step Name	The names of InSteps cannot be changed, see also section Step names (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.
Supervision times and delay time	Define the step supervision times and the delay time in this area, see also section Step times (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual. For InSteps, the steps times can be exclusively defined as literals, page 926. • **Maximum** Enter the maximum supervision time for the current InStep in this text box. • **Minimum** Enter the minimum supervision time for the current InStep in this text box. • **Delay** Enter the delay time for the current InStep in this text box.
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Actions tab

Representation of the **Actions** tab:



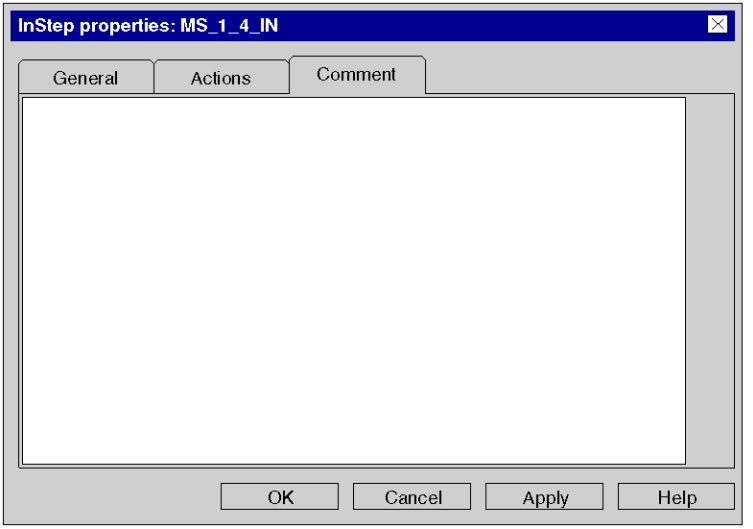
Elements of the **Actions** tab:

Element	Description
Qualifier	Select the qualifier, page 964 for the current action from this list box, see also section Qualifiers (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.
Time	Define the time period for the qualifiers, page 964 `L` (time limited), `D` (time delayed) or `DS` (delayed and saved) in this area. The contents of this area depends on the option: • **Literal** When you activate this option button, you can enter the time period as literals in the text box. • **Variable** When you activate this check box, you can enter the time period in the data selection box, page 1850 as a `TIME` data type.
Action	Define the action in this area. The contents of this area depends on the selected option: • **Variable** When you activate this check box, you can enter the action as an action variable, page 961 in the data selection box, page 1850.

Element	Description
	• **Section** When you activate this check box, you can enter the action as an action section, page 963 in the list box.
Action list box	All actions of the current step are displayed in this list box.
Edit Action section	This command button is only available when an action section is selected in the action list box. Use this command button to close the properties dialog box and open the action section selected.
Accept action	Use this command button to replace the definitions of the selected action in the action list box with the modified definitions, page 960.
New action	Use this command button to insert the new action, page 958 in the action list box.
Delete action	Use this command button to delete the selected action in the action list box., page 966 **Note:** When deleting actions with action sections you are asked to confirm whether the action section should be deleted. Note that deleted sections cannot be restored using the **Edit > Undo** menu command.
Up	Use this command button to move the selected actions in the action list box one position up, see also Modifying the execution order of actions, page 966.
Down	Use this command button to move the selected actions in the action list box one position down, see also Modifying the execution order of actions, page 966.
OK	Us this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Comment tab

Representation of the **Comment** tab:



Elements of the **Comment** tab:

Element	Description
Text box	Enter a comment about the InStep.
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Editing Output Steps

Overview

This section describes how to edit output steps in the SFC programming language.

Defining the properties of OutSteps

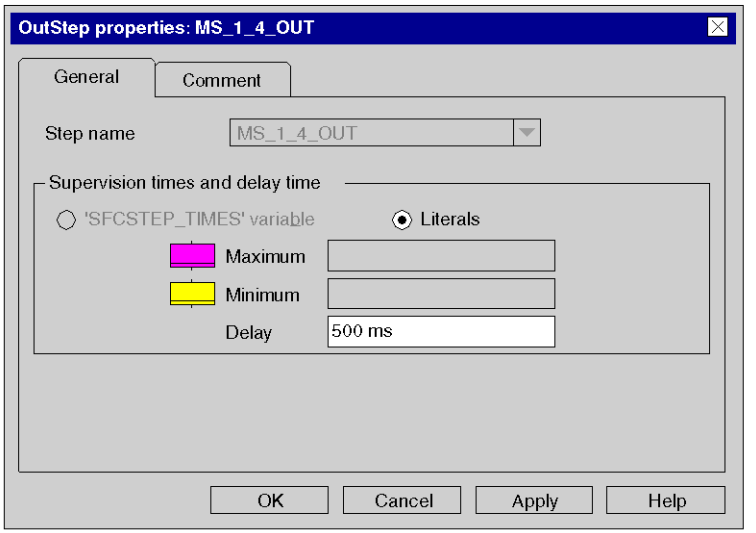
Introduction

The following can be defined for OutSteps:

- the delay time, page 954
- a comment, page 955

Defining the delay time

Defining the delay time as a literal:

Step	Action
1	Open the properties dialog box, page 908 for the OutStep. **Result**: The properties dialog box for the step is opened, see also OutStep properties dialog box, page 955.
2	Enter the value of the delay time, see also section Step times (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual. Example: 
3	Confirm the entries with **OK**. **Result:** The delay time is enabled, the properties dialog box is closed, and a syntax and semantics check, page 894 is performed.

Entering a comment

Entering a comment:

Step	Action
1	Open the properties dialog box, page 908 for the OutStep.
2	Select the **Comment** tab and enter a step comment.
3	Confirm the comment with **OK**. **Result:** The comment is enabled and the properties dialog box is closed.

OutStep properties dialog box

Calling the properties dialog box

see Displaying the object properties, page 908

Structure of the properties dialog box

The OutStep properties dialog box consists of two tabs:

- **General**

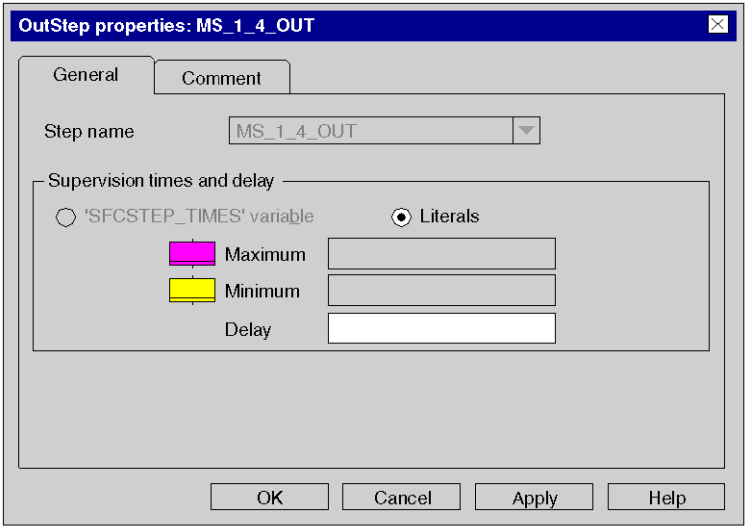
  In this tab you can define:

  - the delay time, page 947
- **Comment**

  In this tab, a comment, page 948 about the OutStep can be entered.

General tab

Representation of the **General** tab:

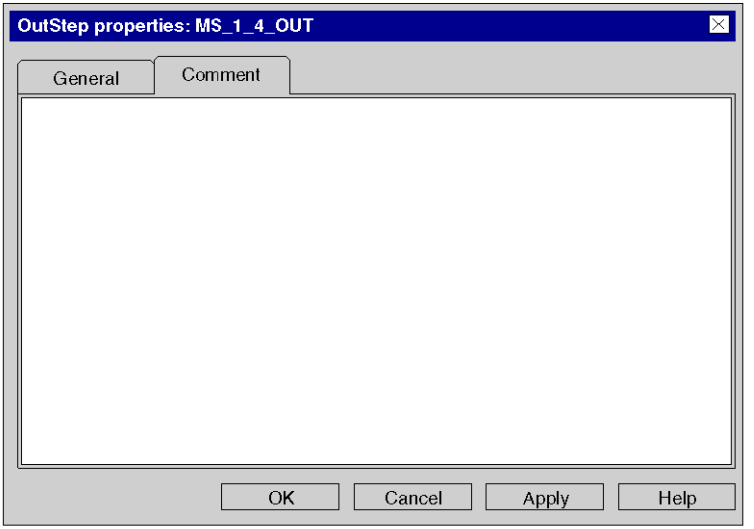


Elements of the **General** tab:

Element	Description
Step Name	The names of OutSteps cannot be changed, see also section Step names (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.
Supervision times and delay time	Define the delay time in this area, see also section Step times (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual. For OutSteps, the delay time can be exclusively defined as literal, page 954. • **Maximum** Not available • **Minimum** Not available • **Delay** Enter the delay time for the current OutStep in this text box.
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Comment tab

Representation of the **Comment** tab:



Elements of the **Comment** tab:

Element	Description
Text box	Enter a comment about the OutStep.
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Editing actions

Overview

This section describes how to edit actions in the SFC programming language.

Assigning actions to a step

Introduction

Each step contains zero or a maximum of 20 actions, see also section Actions (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.

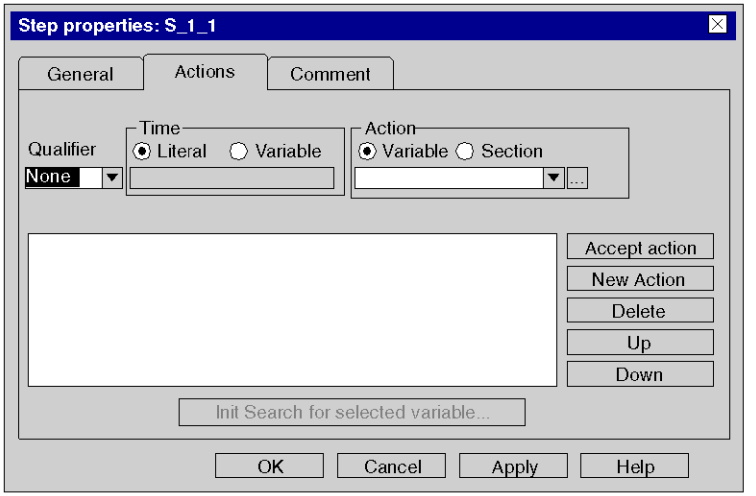
The following can be defined for actions:

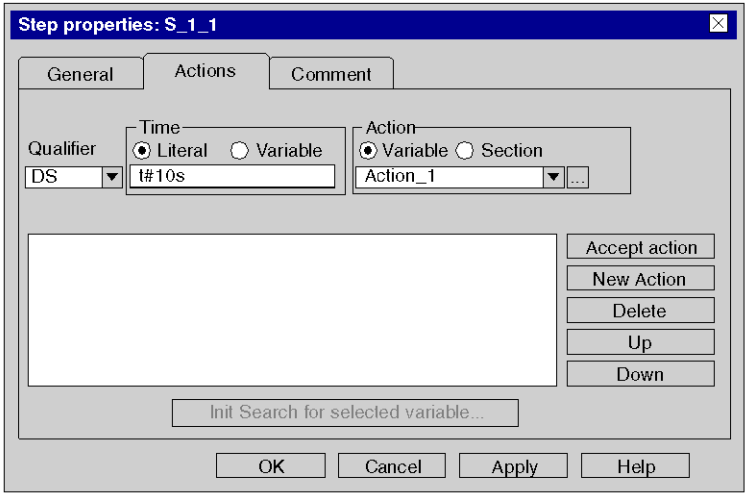
- as action variable, page 961, see also section Action variables (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual
- as action section, page 963 of the programming languages FBD, LD, IL or ST, see also section Action sections (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual
- as Qualifiers, page 964, see also section Qualifiers (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual
- the order, page 966 in which the defined actions should be processed

Defining a new action

Defining a new action:

Step	Action
1	Open the properties dialog box, page 908 for the step. **Result**: The step properties dialog box is opened, see also Steps properties dialog box, page 931.
2	Select the **Actions** tab. **Result:** The dialog box for defining actions is opened.

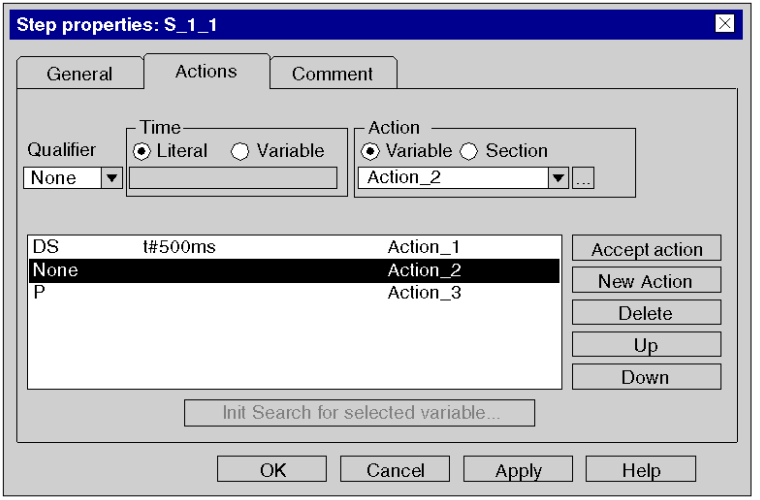
Step	Action
	Example: 
3	In the **Action** area, define an action as • action variable, page 961 or • action section, page 963
4	If necessary, define an **Qualifier** in the Qualifier symbol, page 964 list box.

Step	Action
5	Confirm the definitions for the action with the **New action** button. **Result:** The action is enabled in the action list box. Example: 
6	Confirm the entries with **OK**. **Result:** The actions are accepted, the properties dialog box is closed, and a syntax and semantics check, page 894 is performed.

Modifying an action

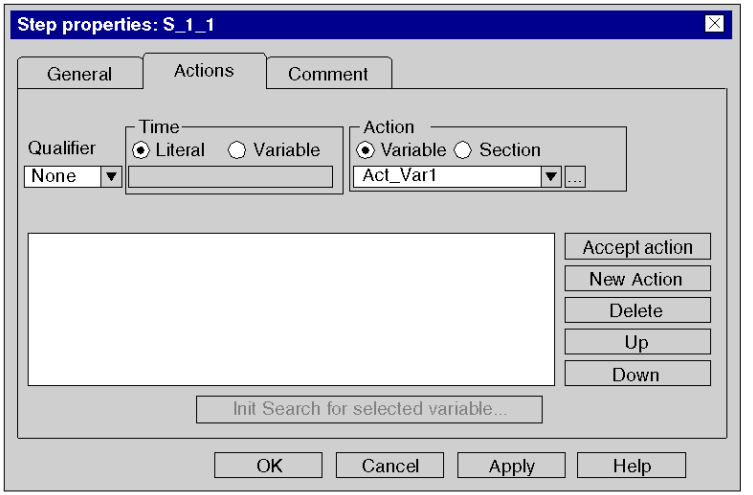
Modifying an action:

Step	Action
1	Open the properties dialog box, page 908 for the step. **Result:** The step properties dialog box is opened, see also Steps properties dialog box, page 931.
2	Select the **Actions** tab. **Result:** The dialog box for defining actions is opened.
3	Select the action to modify from the actions list box. **Result:** The properties of the selected action are enabled in the **Qualifier**, **Time** and **Action** areas.

Step	Action
	Example: 
4	If necessary, modify in the **Action** area • action variable, page 961 or • action section, page 963 **Note:** When modifying action sections you are asked to confirm whether the current action section should be deleted. Please note that deleted sections cannot be undone using**Edit > Undo** .
5	If necessary, define an **Qualifier** in the Qualifier symbol, page 964 list box.
6	Confirm the changes for the action with the **Accept action** button. **Result:** The definitions of the selected action in the action list box are replaced with the changed definitions.
7	Confirm the entries with **OK**. **Result:** The actions are accepted, the properties dialog box is closed, and a syntax and semantics check, page 894 is performed.

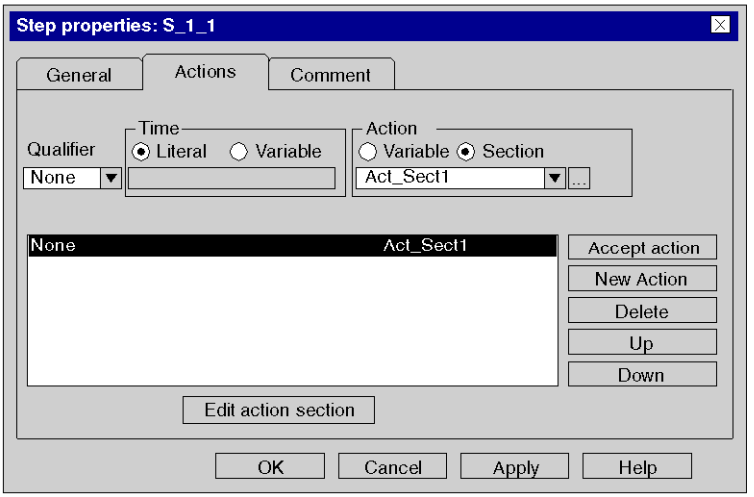
Defining an action variable

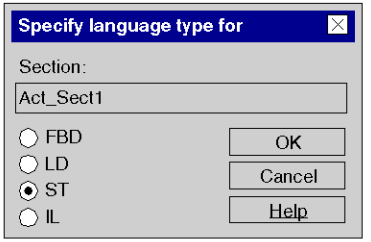
Defining an action variable:

Step	Action
1	Select the **Actions** tab from the step properties dialog box, if it isn't already activated.
2	In the **Action** area, activate the **Variable** option button, if it isn't already activated.
3	Enter a Boolean variable/address, or a Boolean element of a multi-element variable in the data selection box, page 1850, see also section Action variables (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual. You have the following options to enter variables: • You can enter the name of the variable. • You can select the name of the variable from the list of recently used names using the ▼ symbol. or • Using the **button...** you can open a variable selection dialog box, page 1181 and confirm the selected variables there with **OK**. Example: 
4	If necessary, define an qualifier, page 964 for the action. If you do not define an qualifier, the action section is automatically assigned the qualifier `N` (no qualifier).
5	Click the **New action** button to insert a new action, page 958, or the **Accept action** button to modify an existing action, page 960. **Result:** The action is enabled in the action list box.
6	If the action variable has not been declared, declare it as `BOOL` data type, see also Creation of EDT instances, page 390.

Defining a new action section

Defining a new action section:

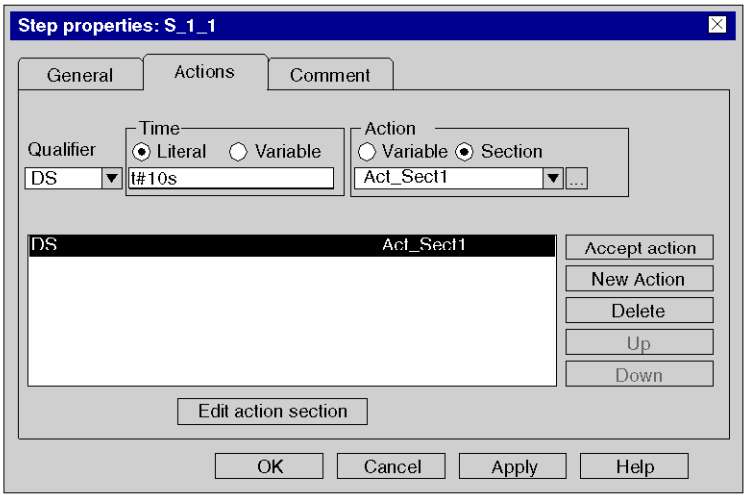
Step	Action
1	Select the **Actions** tab from the step properties dialog box, if it isn't already activated.
2	In the **Action** area, activate the **Section** option button.
3	Enter the name of the action section in the list box, see also section Action sections (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.
4	If necessary, define an qualifier, page 964 for the action section. If you do not define an qualifier, the action section is automatically assigned the qualifier N (no qualifier).
5	Click the **New action** button to insert a new action, page 958, or the **Accept action** button to modify an existing action, page 960. **Result:** The action is enabled in the action list box. Example: 
6	Click the **Edit Action section** button to create the action section logic. **Result:** A dialog box is opened for defining the programming language for the action section.

Step	Action
	Example: 
7	Select the desired programming language for the action section and confirm the selection with **OK**. **Information for creating LD sections:** An LD section contains 11-63 columns and 17-3998 lines. The number of columns is defined using the **Tools > Project Settings...dialog box, in the** Editors **tab, in the** Number of columns **text box**. These settings are only valid for new sections and not for existing sections. Always define the number of columns before you create a new LD section. (The number of lines can be increased or reduced during the creation of section contents.) **Result:** The step properties dialog box is closed and the action section is opened.
8	Create the action section logic, see also section Action sections (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual. **Note:** If the action sections subprograms are called in the **Tools > Project settings... > Language extensions** dialog box, the options **Subprograms allowed** and **Multi Token allowed** must be activated. Please note that the subroutine called from the action section do **not** affect the sequence string control, i.e. • the qualifier assigned to the action section has no effect on the subroutine, • the subroutine remains active when the calling step is deactivated.

Defining the qualifier

Defining the qualifier for an action:

Step	Action
1	Select the **Actions** tab from the step properties dialog box, if it isn't already activated.
2	Select the desired qualifier from the **Qualifier** list box, see also section Qualifiers (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.
3	If you select an qualifier of type `L` (time limited), `D` (time delayed) or `DS` (delayed and saved), you are assigning a time period for the qualifier.

Step	Action
	You can define the time period as a literal or as a variable of the `TIME` data type: • **Literal** 1. In the **Time** area, activate the **Literal** option button, if it isn't already activated. 2. Enter a time period, e.g. `500 ms`, `1 sec`. • **Variable** 1. Activate the **Variable** option button in the **Time** area. 2. You have the following options to enter variables: You can enter the name of the variable. You can select the name of the variable from the list of recently used names using the ▼ symbol. or Using the **button...** you can open a variable selection dialog box, page 1181 and confirm the selected variables there with **OK**. 3. If the variable is not already declared, declare it as a variable of the `TIME` data type and assign the variables with the desired time period, see also Creation of EDT instances, page 390.
4	Define the action variable, page 961 or the action section, page 963.
5	Click the **New action** button to insert a new action, page 958, or the **Accept action** button to modify an existing action, page 960. **Result:** The action is enabled with the qualifier in the action list box. Example: 

Modifying the execution order of actions

Modifying the execution order of actions:

Step	Action
1	Select the **Actions** tab from the step properties dialog box, if it isn't already activated.
2	Select the action from the action list box whose execution order you want to change. **Note:** If several actions are assigned to a step, they are executed in the order they are listed in the action list box. **Exception:** Regardless of their position in the action list box, actions with the qualifier `P1` are always processed first and actions with the qualifier `P0` are always processed last, see also section Qualifiers (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.
3	Move the selected action to the target position using the **Move up** and **Move down** buttons.

Deleting actions

Deleting actions:

Step	Action
1	Select the **Actions** tab from the step properties dialog box, if it isn't already activated.
2	Select the action to be deleted from the actions list box. The following selection options are available: • Selecting an action 1. Click the action. • Selecting several non-sequential actions 1. Press the **Shift** key and keep it pressed. 2. Click the first action to be selected. 3. Click the last action to be selected. • Selecting several sequential actions 1. Press the **Ctrl** key and keep it pressed. 2. Click the actions to be selected.
3	To delete the selected action(s), click the **Delete action** button. **Note:** When deleting actions with action sections you are asked to confirm whether the action section should be deleted. Please note that deleted sections CANNOT be restored using **Edit > Undo**.

Display modes for actions

Introduction

The following display modes are available:

- Input mode
- Expanded display

You have the following options to switch between modes:

- Use the **View > Expanded Display** menu command.

  or

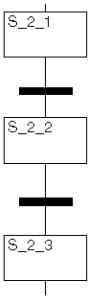
- Press the **Ctrl**+**E** key combination.

You can see which mode is active (Expanded Display) by way of a check-symbol in front of the menu command.

Input mode

Only the step symbol is displayed in this mode.

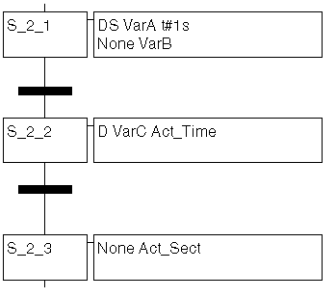
Example of a section in input mode:



Expanded display

In addition to the step symbol, this mode displays the actions assigned to the section and their properties (qualifier, name of the action variable/section and the time period defined for the qualifier, if available). If more than two actions are assigned to a step you can scroll the list.

Example of a section in Expanded Display:



Editing Transitions

Overview

This section describes how to edit transitions in the SFC programming language.

Placing transitions

Introduction

A transition provides the condition through which the checks of one or more pre-transition steps pass on one or more consecutive steps along the corresponding link.

Transitions can be placed in any free cell.

If a transition is placed in a cell that is already occupied by an object, an error message is returned.

If a transition is placed in a cell where the neighboring cell above or below already contains a transition, an error message is returned because a step must be placed between two transitions (see also section Connection rules (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual).

A transition which is placed always connects automatically with the neighboring objects above and below if there are no free cells between them.

Activating transition placement mode

You have the following options to activate the transition placement mode:

- Use the **Edit > New > Transition** menu command.
- Use the **Transition** menu command from the shortcut menu.
- Press the **Shift+F3** key combination.

  or

- Select the  symbol.

The active transition placement mode is indicated by the  cursor symbol.

Note

In the **Tools > Options** dialog box on the **Data and Languages** tab, if the **Automatically assign a variable to a new graphical object** check box is selected, then the corresponding properties dialog box is automatically opened when an object is placed. The procedures explained here are used when the check box is **cleared**.

Placing transitions

Placing transitions:

Step	Action
1	Activate the transition placement mode.
2	Click the target cell in the SFC section. or Use the **arrow keys** to move the cursor to the target position, and press **Enter**. **Result:** The transition is inserted and the selection mode is active again.
3	To insert further transitions: • Click the target cell in the SFC section. or Use the **arrow keys** to move the cursor to the target position, and press **Enter**. • To return to selection mode: Press the **Esc** key. • To insert other objects: Select the object you wish to insert.

Defining transition properties

Introduction

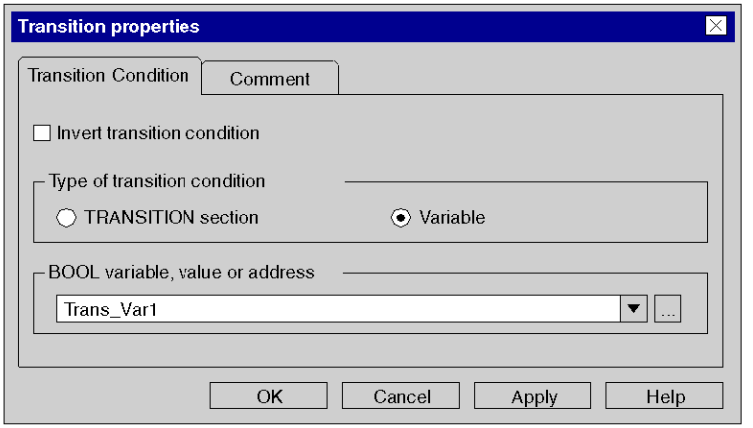
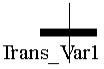
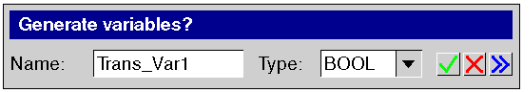
The following can be defined for transitions:

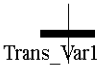
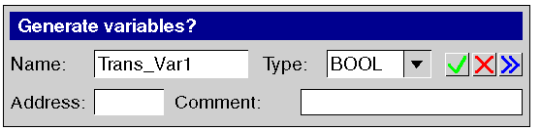
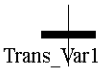
- the assigned transition condition
  - as variable, page 970
  - as transition section, page 972
- whether the transition condition should be inverted, page 974 (trigger the transition when 0)
- a comment, page 974

Defining the transition condition as a variable

Defining the transition condition as a variable:

Step	Action
1	Open the properties dialog box, page 908 for the transition. **Result**: The properties dialog box for the transition is opened, see also Properties dialog box for transitions, page 975.
2	In the **Type of transition condition** area, activate the **Variable** option button, if it isn't already activated.
3	In the **BOOL variable, value or address** data selection box, page 1850, enter a Boolean variable/address, a Boolean element for a multi-element variable or a Boolean literal, see also section Transition conditions (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual. You have the following options to enter variables: • You can enter the name of the variable. • You can select the name of the variable from the list of recently used names using the ▼ symbol. or • Using the command button **...** you can open a variable selection dialog box, page 1181 and confirm the selected variables there with **OK**.

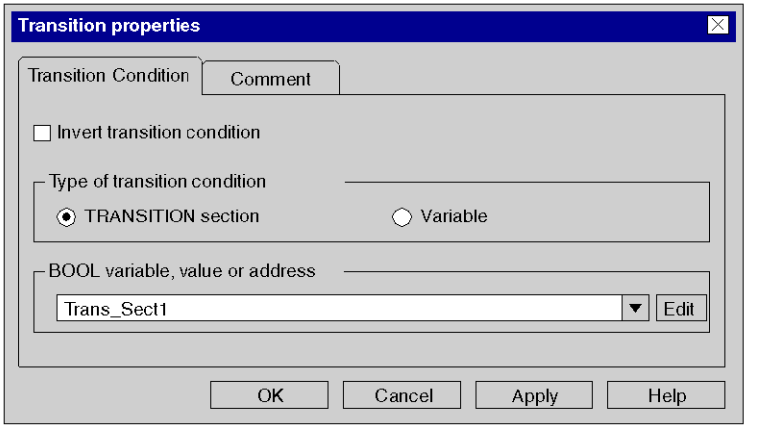
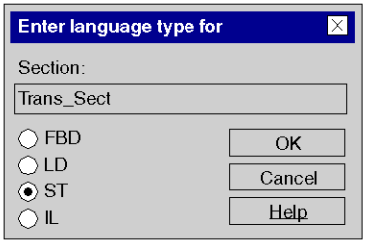
Step	Action
	Example: 
4	Confirm the entry with **OK**. **Result:** • If the selected variable is already declared, the transition condition is accepted, the properties dialog box is opened, a syntax and semantics check is, page 894 carried out and the procedure is ended.  • If the selected variable is not declared, a variable declaration dialog box opens. In this case, continue with the following steps of this procedure. Variable declaration dialog box  **Note:** If you wish to declare the variable later, close the dialog box using the symbol. In this case, the variable name is enabled in the section, but not declared. You can declare the variables later in the Data Editor, page 390 or by selecting the transition and the executing the menu command **Generate Variable** in the shortcut menu.
5	If you do not want to assign the variable an address or comment, confirm using **Enter** or the symbol. **Result:**The variable is declared and enabled in the section. A syntax and semantics check, page 894 is performed and the procedure ends here.

Step	Action
	Example: 
6	To assign an address or comment, use the  symbol. **Result:** The dialog box expands. 
7	Enter the address or comment, or both.
8	Confirm with **Enter** or the  symbol. **Result:**The variable is declared and enabled in the section. A syntax and semantics check, page 894 is performed. Example: 

Defining the transition condition as a section

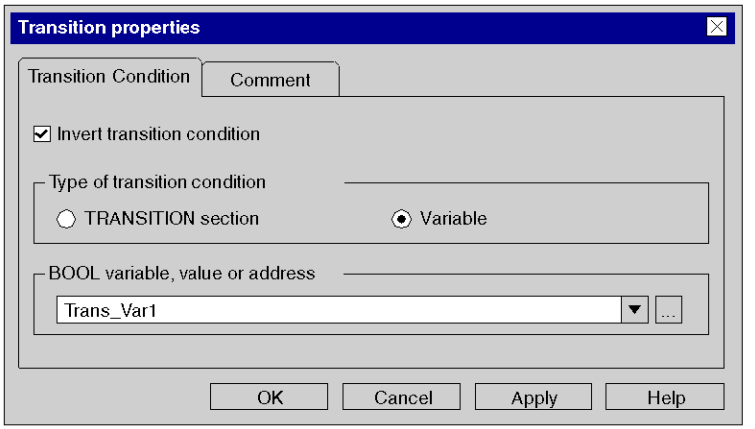
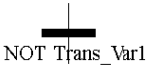
Defining the transition condition as a section:

Step	Action
1	Open the properties dialog box, page 908 for the transition. **Result**: The properties dialog box for the transition is opened, see also Properties dialog box for transitions, page 975.
2	In the **Type of transition condition** area, activate the **TRANSITIONS section** option button.
3	Enter the name of the transition section in the text box, see also section Transition section (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual. You have the following options to enter the section name: • You can enter the section name. or

Step	Action
	• You can select the name of the section from the list of transition sections which are defined but not yet used with the ▼ symbol. Example: 
4	To create the logic for the transition section, click the **Edit** command button. **Result:** A dialog box to define the programming language of the transition section is opened. Example: 
5	Select the desired programming language for the transition section and confirm the selection with **OK**. **Information for creating LD sections:** An LD section contains 11-63 columns and 17-3998 lines. The number of columns is set via the **Tools > Project Settings** dialog box, in the **Editors** tab in the **Number of columns** text box. These settings are only valid for new sections and not for existing sections. Always define the number of columns before you create a new LD section. (The number of lines can be increased or reduced during the creation of section contents.) **Result:** The properties dialog box is closed and the transition section is opened.
6	Create the logic for the transition section, see also section Transition section (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.

Inverting the transition condition

Inverting the transition condition:

Step	Action
1	Open the properties dialog box, page 908 for the transition. **Result**: The properties dialog box for the transition is opened, see also Properties dialog box for transitions, page 975.
2	To invert the transition condition, activate the **Invert transition condition** check box. Example: 
3	Confirm the entry with **OK**. **Result:** The transition condition is represented as inverted in the section, the properties dialog box is closed and a syntax and semantics check, page 894 is performed. Example: 

Enter a comment

Enter a comment:

Step	Action
1	Open the properties dialog box, page 908 for the transition.
2	Select the **Comment** tab and enter a transition comment.
3	Confirm the comment with **OK**. **Result:** The comment is accepted and the properties dialog box is closed.

Properties dialog box for transitions

Calling the properties dialog box

see Displaying the object properties, page 908

Structure of the properties dialog box

The transition properties dialog box consists of two tabs:

- **Transition Condition**

  In this tab you can define:

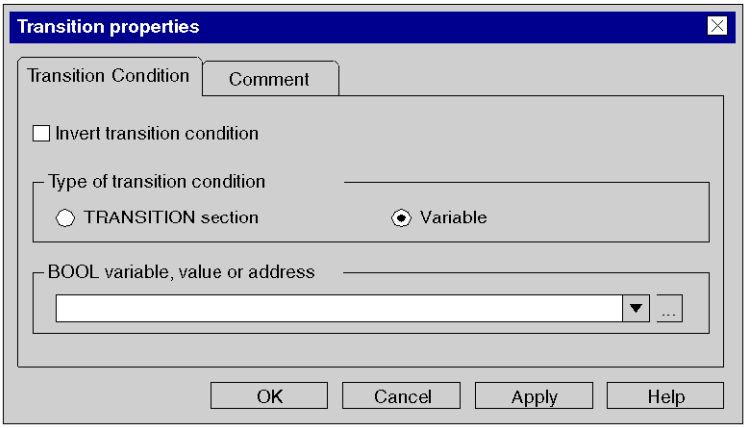
  - the transition condition assigned to the transition as a transition variable, page 970 or as a transition section, page 972
  - whether the transition condition should be inverted, page 974

- **Comment**

  In this tab, a comment, page 931 about the transition can be entered.

Transition Condition tab

Representation of the **Transition Condition** tab:

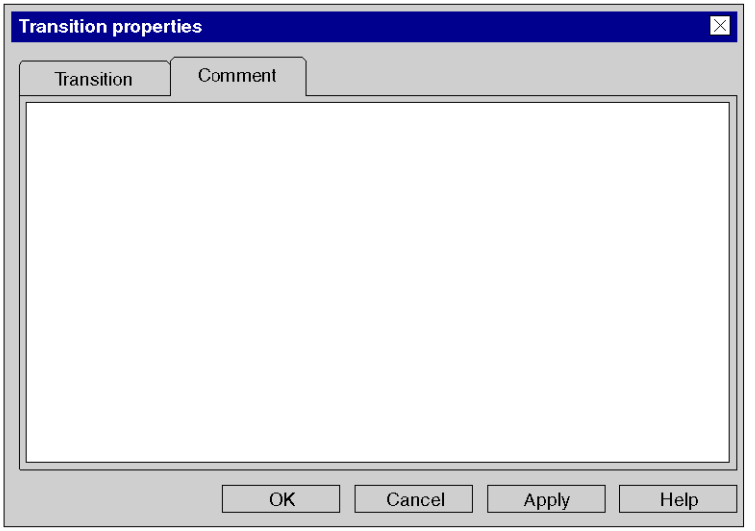


Elements of the **Transition Condition** tab:

Element	Description
Invert transition condition	If you activate this check box, the transition condition is inverted (trigger the transition when 0).
Type of transition condition	In this area, you can define the transition condition, see also section Transition conditions (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual. The contents of this area depends on the selected option: • **Variable** If you activate this option button, you can enter a Boolean variable/address, a Boolean element of a multi-element variable or a Boolean literal in the **BOOL variable, value or address** data selection box, page 1850. • **TRANSITION section** When you activate this option button, you can go to the **TRANSITION Section** text box and enter the name of the transition section. **Note:** This text box is only used for referencing transition sections. If you want to rename an existing transition section, you must execute this in the properties dialog of the transitions section.
Edit	This command button is only available when the TRANSITION section option button is activated in the **Type of transition condition** area. Use this command button to close the properties dialog box and open the respective transition section.
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Comment tab

Representation of the**Comment**tab:



Elements of the **Comment** tab:

Element	Description
Text box	Enter a comment about the transition.
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Editing Jumps

Overview

This section describes how to edit jumps in the SFC programming language.

Placing jumps

Introduction

Jumps are used to represent directional links which are not displayed in full length.

Jumps can be placed in any free cell.

If a jump is placed in a cell that is already occupied by an object, an error message is returned.

If a jump is placed in a cell where the neighboring cell above already contains a step, an error message is returned because a transition must be placed between a step and a jump (see also section Connection rules (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual).

A jump which is placed always connects automatically with the neighboring objects above if there are no free cells between them.

Use the context sensitive **Go to**, **Go to step** or **List and display jumps** dialogs to find a certain jump in the current section.

Activating the jump placement mode

You have the following options to activate the jump placement mode:

- Use the **Edit > New > Jump** menu command.
- Use the **Jump** menu command from the shortcut menu.

  or

- Choose the  symbol.

The active jump placement mode is indicated by the  cursor symbol.

Note

If in the **Tools > Options** dialog in the **Data and Languages** tab the **Automatically assign a variable to a new graphics object** checkbox is enabled, an objects property dialog will automatically open during placement. The procedures explained here are used when the check box is **cleared**.

Placing jumps

Placing jumps:

Step	Action
1	Activate the jump placement mode.
2	Click the target cell in the SFC section. or Use the **arrow keys** to move the cursor to the target position, and press **Enter**. **Result:** The jump is inserted and the selection mode is enabled again.
3	To insert further jumps: • Click the target cell in the SFC section. or Use the **arrow keys** to move the cursor to the target position, and press **Enter**. • To return to selection mode: Press the **Esc** key. • To insert other objects: Select the object you wish to insert.

Defining jump properties

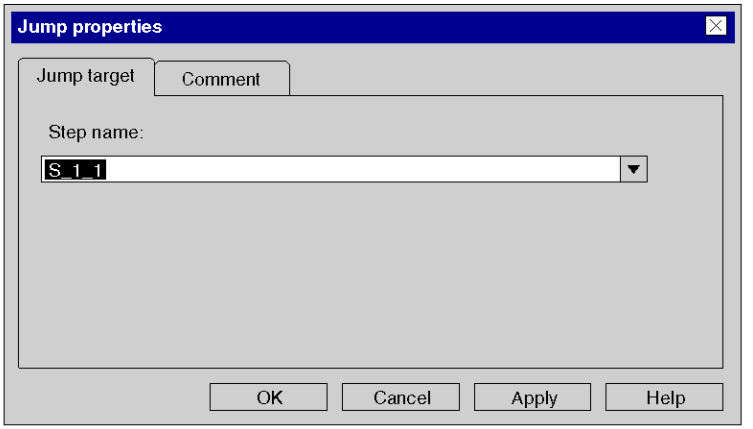
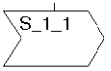
Introduction

The following can be defined for jumps:

- the jump target, page 979, see also Jump (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.
- a comment, page 981

Defining the jump target

Defining the jump target:

Step	Action
1	Open the properties dialog box, page 908 for the jump. **Result**: The jump properties dialog box is opened, see also Calling the properties dialog box, page 981.
2	Enter the name of the target step in the text box, see also Jump (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual. **Note:** With multi-token, you can enable the option to jump into a parallel sequence or out of a parallel sequence, see also the **Tools > Project Settings** dialog box, in the **Language extensions** tab, **Sequential control (SFC)** area. You have the following options to enter the target step name: • You can enter the step name. or • You can select the name of the step from the list of steps used in the current section with the ▼ symbol. Example: 
3	Confirm the selection with **OK**. **Result:** The properties dialog box is closed, the jump target is accepted and a syntax and semantics check, page 894 is performed. Example: 
4	Create the logic for the transition section, see also Transition section (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.

Entering a comment

Entering a comment:

Step	Action
1	Open the properties dialog box, page 908 for the jump.
2	Select the **Comment** tab and enter a jump comment.
3	Confirm the comment with **OK**. **Result:** The comment is enabled and the properties dialog box is closed.

Jump properties dialog box

Calling the properties dialog box

You have the following options to call the properties dialog box:

- Double-click the jump.
- Select the jump and use the **Edit > Properties...** menu command.

  or
- Select the jump and use the **Object Properties...** menu command from the shortcut menu.

Structure of the properties dialog box

The jump properties dialog box consists of two tabs:

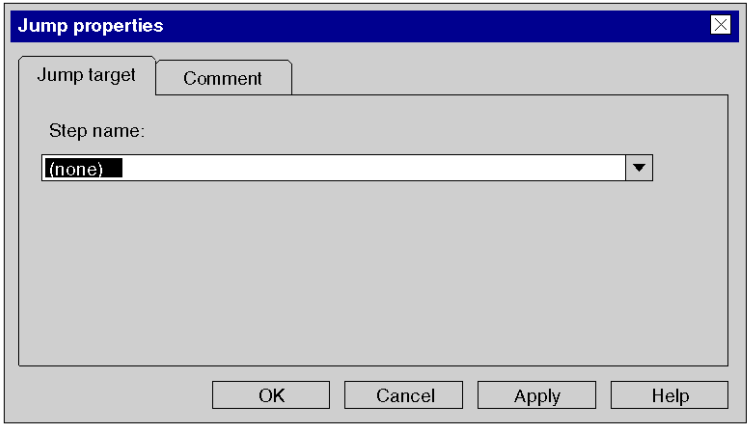
- **Jump target**

  You can define the Jump target, page 979 this tab.
- **Comment**

  In this tab, a comment, page 931 about the transition can be entered.

Jump target tab

Representation of the **Jump target** tab:

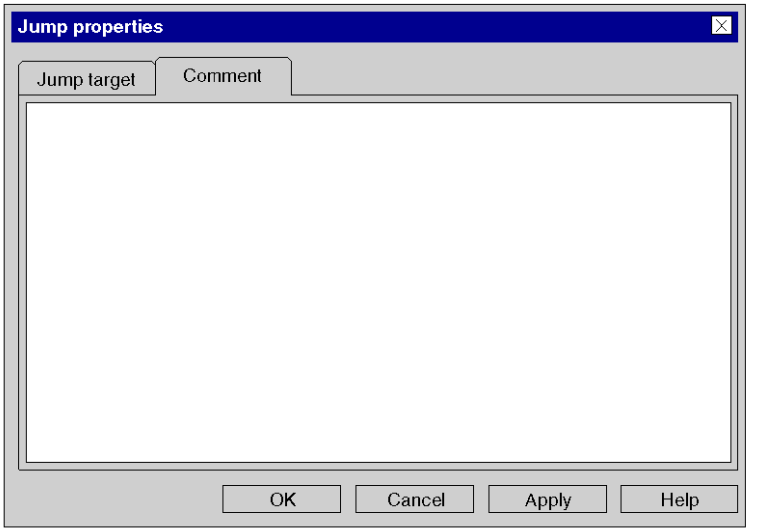


Elements of the **Jump target** tab:

Element	Description
Step Name	Enter the name of the target step, page 979 in this text box, see also section Jumps (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Comment tab

Representation of the **Comment** tab:



Elements of the **Comment** tab:

Element	Description
Text box	Enter a comment about the transition.
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Editing Branches and Joints

Overview

This section describes how to edit branches and joints in the SFC programming language.

Selecting Branches and Joints

Introduction

Alternative sequences and parallel sequences are differentiated in SFC:

- **Alternative sequence**

  The **alternative sequence** offers the option to program conditional branches in the control flow of the SFC structure, see also Alternative Branches and Alternative Joints (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.

  Alternative sequences consist of:

  - **Alternative branch**

    With alternative branches, as many transitions follow a step under the horizontal line as there are different processes.

  - **Alternative joint**

    All alternative branches are linked to a single branch using alternative joints or jumps where they are then continued to be processed.

- **Parallel sequence**

  With parallel sequences the connection of an individual transition leads to the parallel activation of several (maximum 32) steps (branches). After this activation, the individual branches are processed independently from one another, see also Parallel Branches and Parallel Joints (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.

  Parallel sequences consist of:

  - **Parallel branch**

    With parallel branches, as many transitions follow a step under the horizontal line as there are parallel processes.

  - **Parallel joint**

    In accordance to IEC 61131-1, all parallel branches are joined at their ends by a common parallel joint. The transition to a parallel connection is evaluated when all previous steps of the transition are set.

    If a parallel branch is combined using an alternative joint, this is only possible with Multi-Token-Operation, see also dialog box **Tools > Project Settings** tab **Language extensions** area **Sequential control (SFC)**.

Selecting Branches and Joints

Selecting Branches and Joints:

Type	Call via menu command	Call via symbol	Call via keyboard	Cursor symbol
Alternative branch, page 985	**Edit > New > Alternative branch** or **Branches and Joints > Alternative branch** shortcut menu		**F4**	
Alternative joint, page 985	**Edit > New > Alternative joint** or **Branches and Joints > Alternative joint** shortcut menu		**Shift+F4**	
Parallel branch, page 985	**Edit > New > Parallel branch** or **Branches and Joints > Parallel branch** shortcut menu		**F5**	
Parallel joint, page 985	**Edit > New > Parallel joint** or **Branches and Joints > Parallel joint** shortcut menu		**Shift+F5**	

Placing branches and joints

Introduction

Branches and joints do not require their own cell, but are inserted into the respective cell of the step or transition.

If a branch or joint is placed in a cell that already contains a branch or joint an error message is returned (see also section Connection rules (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual).

A branch or joint which is placed connects automatically with the neighboring objects above and below if there are no free cells between them.

Placing branches or joints

Placing branches or joints:

Step	Action
1	Select the desired branch or joint, see also Selecting Branches and Joints, page 984.
2	Click the target cell in the SFC section. or Use the **arrow keys** to move the cursor to the target position, and press **Enter**. **Result:** The selected branch or joint is inserted and the selection mode is active again.
3	To place further elements of the same type: • Click the target cell in the SFC section. or Use the **arrow keys** to move the cursor to the target position, and press **Enter**. • To return to selection mode: Press the **Esc** key. • To insert other objects: Select the object you wish to insert.

Modifying the Properties of Branches and Joints

Introduction

The following can be defined for branches:

- the number of output pins
  - by dragging the mouse, page 987
  - using the properties dialog box, page 987
- the position of the input pin
  - using the properties dialog box, page 988

The following can be defined for joints:

- the number of input pins
  - by dragging the mouse, page 987
  - using the properties dialog box, page 987
- the position of the output pins
  - using the properties dialog box, page 988

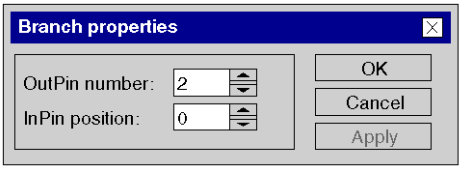
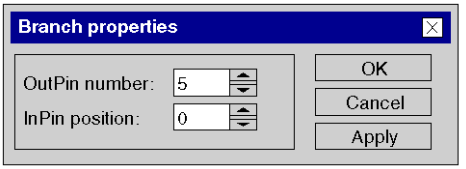
Defining the number of pins using the mouse

Defining the number of pins using the mouse:

Step	Action
1	Select the branch or joint. **Result:** A change label is placed at both ends of the branch or joint. Example: 
2	Place the mouse pointer on one of the change labels. **Result:** The mouse pointer changes its symbol to ↔.
3	Drag the branch or joint in the desired direction using the mouse. **Result:** The branch or joint is expanded to the desired number of pins. Example: 

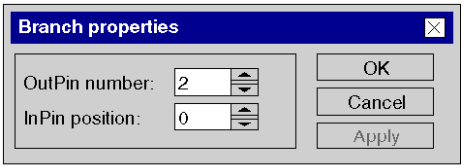
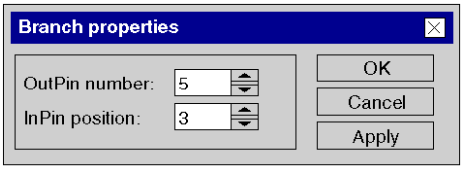
Defining the number of pins using the properties dialog box

Defining the number of pins using the properties dialog box:

Step	Action
1	Open the properties dialog box, page 908 for the branch or joint. **Result**: The properties dialog box for the branch, page 989 or joint, page 990 is opened. Example of a properties dialog box for branches: 
2	Enter the desired number of output pins (for branches) or the desired number of input pins (for joints). Example of a properties dialog box for branches: 
3	Confirm the entry with **OK**. **Result:** The number of pins is accepted. Example: 

Defining the position of the input or output pin

Defining the position of the input pin (for branches) or output pin (for joints):

Step	Action
1	Open the properties dialog box, page 908 for the branch or joint. **Result**: The properties dialog box for the branch, page 989 or joint, page 990 is opened. Example of a properties dialog box for branches: 
2	Enter the target position of the input pin (for branches) or the position of the output pin (for joints). **Note:** The position numbers start at 0. Example of a properties dialog box for branches: 
3	Confirm the entry with **OK**. **Result:** The position of the input or output pin is accepted. Example: 

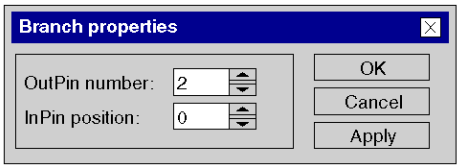
Branches properties dialog box

Calling the properties dialog box

see Displaying the object properties, page 908

Build

Representation of the dialog box:



Elements of the dialog box:

Element	Description
OutPin number	Enter the number of output pins, page 987 of the branch in this text box.
InPin position	Enter the position of the input pin, page 988 of the branch in this text box. **Note:** The position numbers start at 0.
OK	Use this command button to accept all entries and close the properties dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

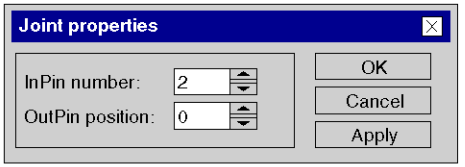
Joints properties dialog box

Calling the properties dialog box

see Displaying the object properties, page 908

Build

Representation of the dialog box:



Elements of the dialog box:

Element	Description
InPin number	Enter the number of input pins, page 987 of the branch in this text box.
OutPin position	Enter the position of the output pin, page 988 of the branch in this text box. **Note:** The position numbers start at 0.
OK	Use this command button to accept all entries and close the dialog box.
Apply	Use this command button to accept all entries without closing the properties dialog box.

Inserting Alternative Strings and Simultaneous Strings

Overview

This section describes how to insert alternative strings and simultaneous strings in the SFC programming language.

Selecting Alternative Sequences and Parallel Sequences

Introduction

You can place alternative sequence and parallel sequence objects (branches, steps, transitions, joints) individually or place entire alternative sequences or parallel sequences. You can select any number of branches or transitions (with alternative sequences) or steps (with parallel sequences).

See also section Branches and Joints (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.

Selecting Sequences

Selecting alternative sequences and parallel sequences

Sequence type	Call via menu command	Call via symbol	Cursor symbol
Alternative sequence, page 992	**Sequences** > **Alternative Sequence** shortcut menu		
Parallel Sequence, page 993	**Sequences** > **Parallel Sequence** shortcut menu		

Placing alternative sequences and parallel sequences

Introduction

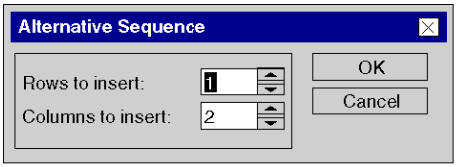
Alternative sequences and branches can only be placed in free cells. The number of free cells required depends on the size of the sequence to be inserted.

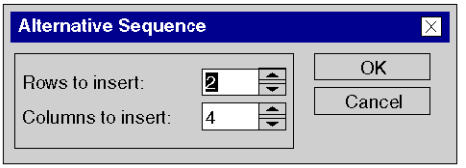
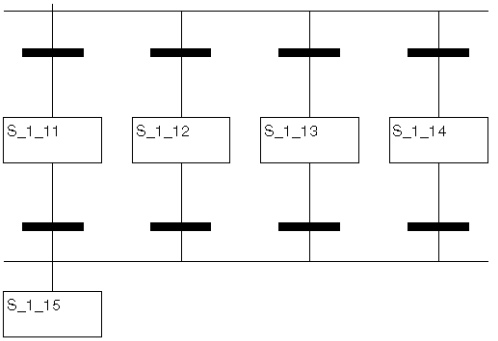
If a cell, in which an alternative sequence or parallel sequence is to be placed, is already occupied by an object, an error message is returned.

An alternative sequence or parallel sequence which is placed always connects automatically with the neighboring objects above and below if there are no free cells between them, see also section Connection rules (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.

Placing an alternative sequence

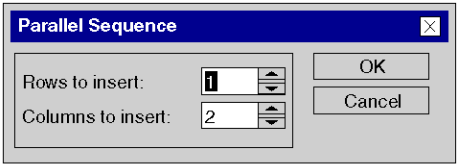
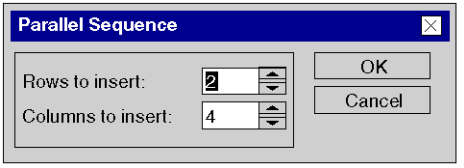
Placing an alternative sequence:

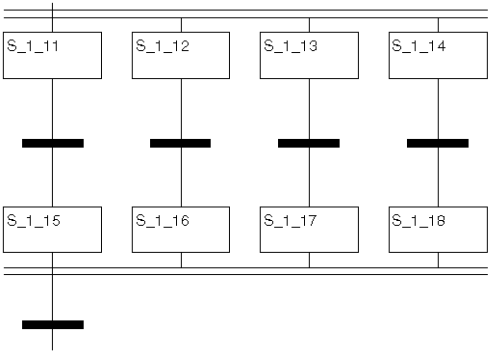
Step	Action
1	Activate the placement mode for alternative sequences, see also Selecting Alternative Sequences and Parallel Sequences, page 991. **Result:** The dialog box for defining the sequence size is opened. Example: 
2	Enter the number of transition-step sequences to be inserted per branch in the **Rows to insert** text box.

Step	Action
3	Enter the number of branches to be inserted into the **Columns to insert** text box. Example: 
4	Confirm the entries with **OK**. **Result:** The dialog box is closed and the cursor changes its symbol to  .
5	Click the cell where the input pin of the alternative sequence should be placed. or Use the **arrow keys** to move the cursor to the target position, and press **Enter**. **Result:** The alternative sequence is inserted into the section. Example: 
6	Define the properties of the placed steps, page 924 and transitions, page 970.

Placing a parallel sequence

Placing a parallel sequence:

Step	Action
1	Activate the placement mode for parallel sequences, see also Selecting Alternative Sequences and Parallel Sequences, page 991. **Result:** The dialog box for defining the sequence size is opened. Example: 
2	Enter the number of step-transition sequences to be inserted per branch in the **Rows to insert** text box.
3	Enter the number of branches to be inserted into the **Columns to insert** text box. Example: 
4	Confirm the entries with **OK**. **Result:** The dialog box is closed and the cursor changes its symbol to  .

Step	Action
5	Click the cell where the input pin of the alternative sequence should be placed. or Use the **arrow keys** to move the cursor to the target position, and press **Enter**. **Result:** The parallel sequence is inserted into the section. Example: 
6	Define the properties of the placed steps, page 924 and transitions, page 970.

Inserting Step-Transitions and Transition-Step-Sequences

Overview

This section describes how to insert step transitions and transition step sequences using the SFC programming language.

Selecting Step-Transition and Transition-Step Sequences

Introduction

You can place the steps and transitions individually or place complete Step-Transitions or Transition-Steps. Any number of steps and transitions can be selected.

Selection

Selecting Step-Transitions and Transition-Step Sequences:

Sequence type	Call via the menu command	Call via the icon	Cursor symbol
Insert Step-Transition-Sequence, page 996	**Sequence > Step-Transition Sequence** shortcut menu		
Transition-Step Sequence, page 998	**Sequence > Transition-Step Sequence** shortcut menu		

Placing step-transition and transition-step sequences

Introduction

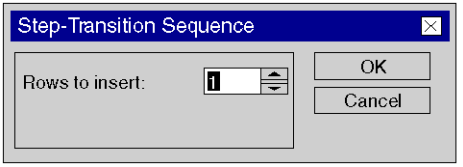
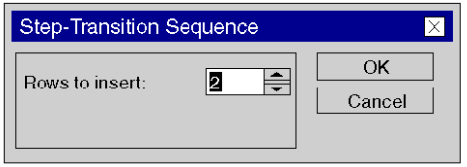
Step-transition and transition-step sequences can only be placed in free cells. The number of free cells required depends on the size of the sequence to be inserted.

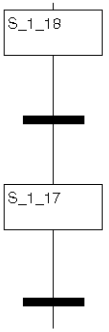
If a cell, in which the step-transition or transition-step sequence is to be placed, is already occupied by an object, an error message is returned.

A step-transition or transition-step sequence which is placed always connects automatically with the neighboring objects above and below if there are no free cells between them, see also section Connection rules (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.

Placing a step-transition sequence

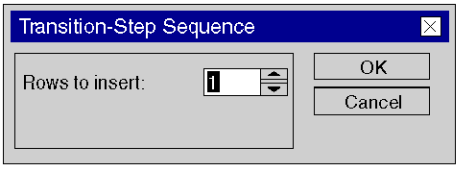
Placing a step-transition sequence:

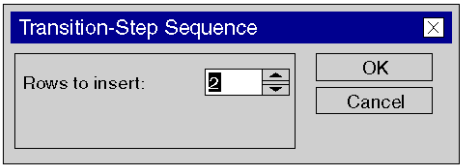
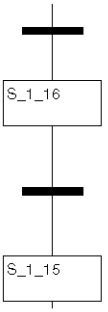
Step	Action
1	Activate the placement mode for step-transition sequences, see also Selecting Step-Transition and Transition-Step Sequences, page 995. **Result:** The dialog box for defining the number of step-transition sequences is opened. Example: 
2	Enter the number of step-transition sequences to be inserted into the **Rows to insert** text box. Example: 
3	Confirm the entry with **OK**. **Result:** The dialog box is closed and the cursor changes its symbol to .

Step	Action
4	Click the cell where the input pin of the step-transition sequence should be placed. or Use the **arrow keys** to move the cursor to the target position, and press **Enter**. **Result:** The step-transition sequences are inserted into the section. Example: 
5	Define the properties of the placed steps, page 924 and transitions, page 970.

Placing a transition-step sequence

Placing a transition-step sequence:

Step	Action
1	Activate the placement mode for transition-step sequences, see also Selecting Step-Transition and Transition-Step Sequences, page 995. **Result:** The dialog box for defining the number of transition-step sequences is opened. Example: 
2	Enter the number of transition-step sequences to be inserted into the **Rows to insert** text box.

Step	Action
	Example: 
3	Confirm the entry with **OK**. **Result:** The dialog box is closed and the cursor changes its symbol to .
4	Click the cell where the input pin of the transition-step sequence should be placed. or Use the **arrow keys** to move the cursor to the target position, and press **Enter**. **Result:** The transition-step sequences are inserted into the section. Example: 
5	Define the properties of the placed steps, page 924 and transitions, page 970.

Editing Links

Overview

Editing links in the SFC programming language.

Links

Introduction

Links connect steps and transitions, transitions and steps, etc.

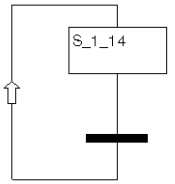
Attention should be paid to the following programming information:

- Links between objects of the same type (step with step, transition with transition, etc.) are not possible.
- Links are possible between:
  - Unused object outputs
  - Used or unused step inputs

    (i.e. step inputs can be connected using links)
- Overlapping links and other SFC objects (step, transition, jump, etc.) is not possible.
- Overlapping links and links is possible.
- Crossing links and links is possible. Crossed links are indicated by a "broken" link.



- The signal flow is generally from top to bottom. In exceptional cases (e.g. return) it can also be from bottom to top. This special case is indicated using an arrow symbol.



NOTE: A detailed description about links can be found in section Links (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual.

Selecting links

You have the following options to activate the placement mode:

- Use the **Edit > New > Link** menu command.
- Use the **Link** menu command from the shortcut menu.
- Press the **F6** key.

  or

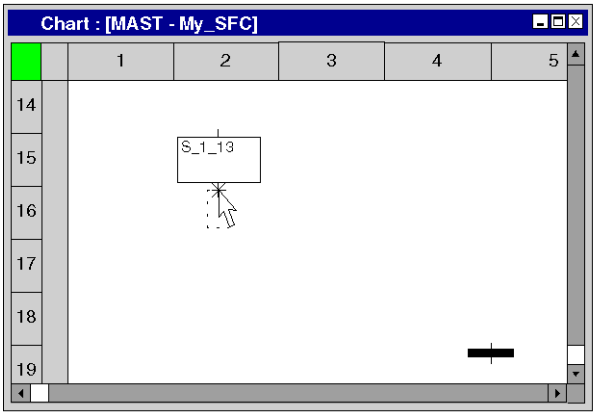
- Select the symbol.

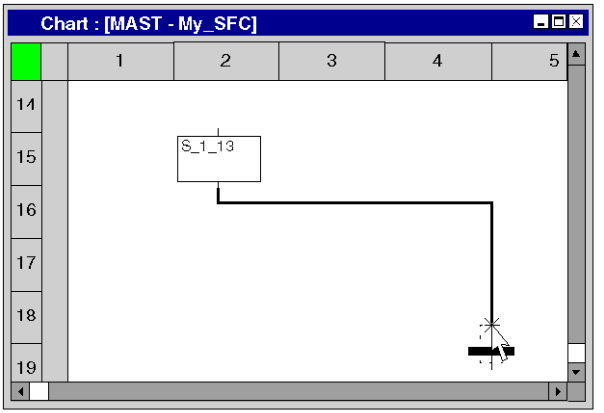
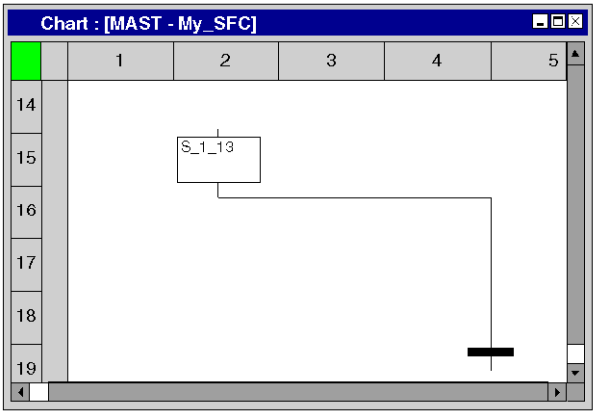
The + and cursor symbols indicate which mode is active.

Placing a Link

Placing a link using the mouse

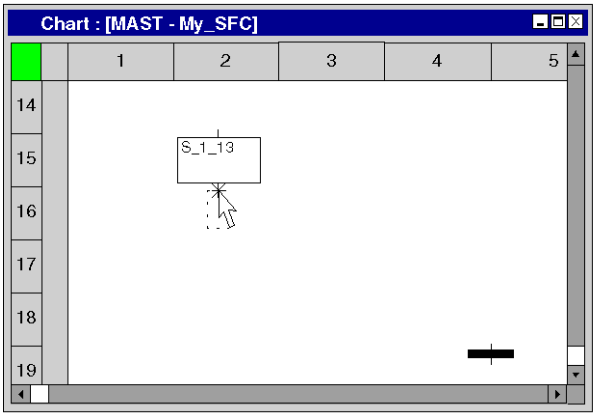
Carry out the following steps to place a link using the mouse:

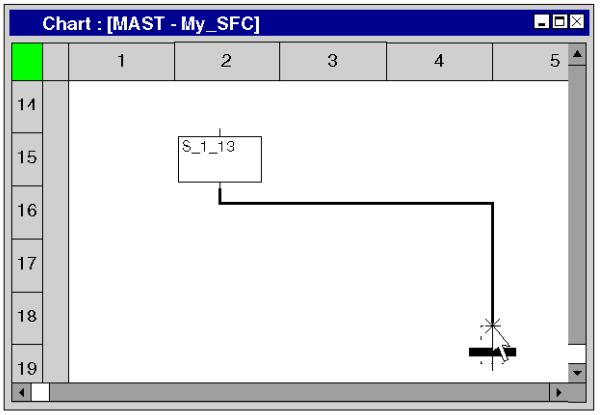
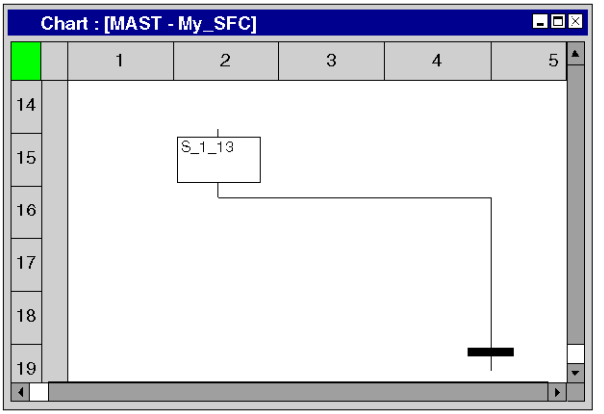
Step	Action
1	Activate the placement mode for links, see also Links, page 1000.
2	Place the mouse pointer on the start point of the link. **Note:** The correct position is shown by the mouse pointer symbol. Example: 

Step	Action
3	Click the left mouse button on the start point and drag mouse pointer to the target point of the link. Example:  **Tips:** • By clicking between the start point and target position you can place intermediate points; for example, to avoid crossing other objects. • You can leave this mode at any time by pressing **Esc**.
4	Click on the target position with the left mouse button. **Result:** The link is inserted. Example:  See also Editing links, page 1005

Placing a link using the keyboard

Carry out the following steps to place a link using the keyboard:

Step	Action
1	Activate the placement mode for links, see also Links, page 1000.
2	Use the **cursor keys** to move the cursor to the start point of the link, and press **Enter**. **Note:** The correct position is shown by the cursor symbol. Example: 

Step	Action
3	Use the **cursor keys** to move the cursor to the target position of the link. Example:  **Tips:** • By pressing the **Enter** key between the start point and target position you can place intermediate points; for example, to avoid crossing other objects. • You can leave this mode at any time by pressing **Esc**.
4	Press **Enter**. **Result:** The link is inserted. Example:  See also Editing links, page 1005

Editing links

Editing links

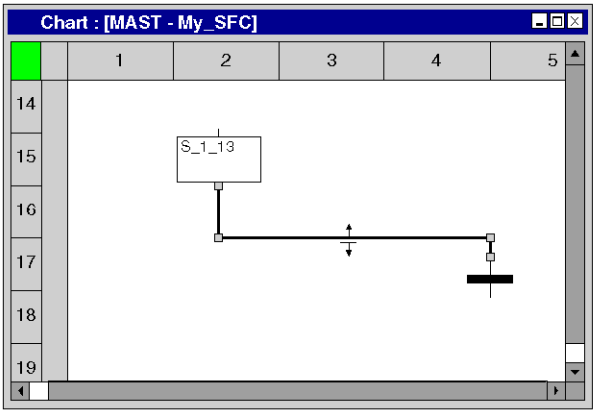
Links are comprised of one or more segments connected to one another.

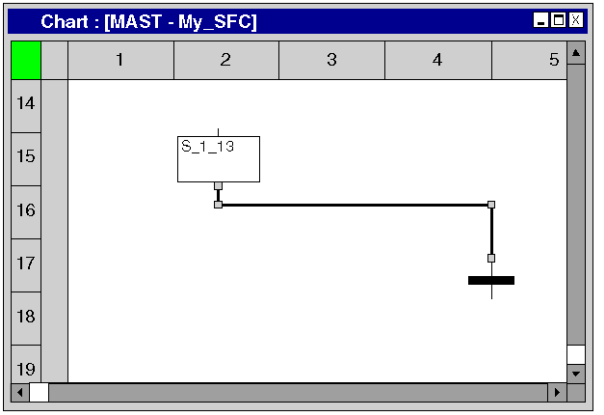
When cutting, page 901, deleting, page 900, copying, page 902 and moving, page 904 links, the action is executed for the entire link.

If a link is selected, the target points are displayed at their beginnings, ends and at every direction change

Moving links vertically/horizontally

Process the following procedures to move a link segment vertically or horizontally:

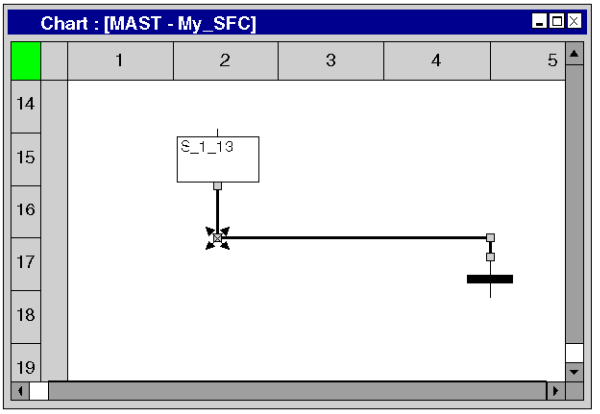
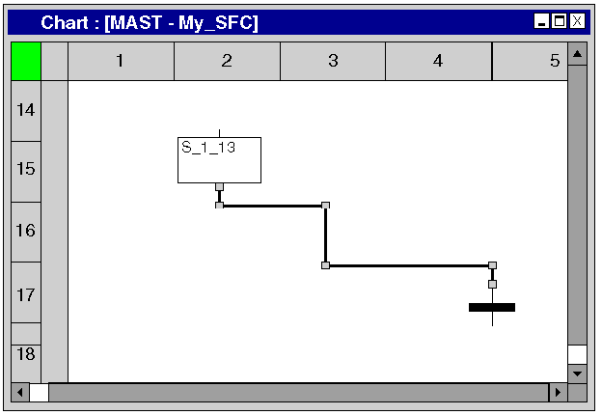
Step	Action
1	Select the link.
2	Place the mouse pointer on the item to be moved. **Note:** The correct position is indicated by the mouse pointer symbol. Example: 

Step	Action
3	Click the left mouse button and drag the segment to the target position.
4	Release the mouse button. **Result:** The change is made. 

Creating angles in FFB links

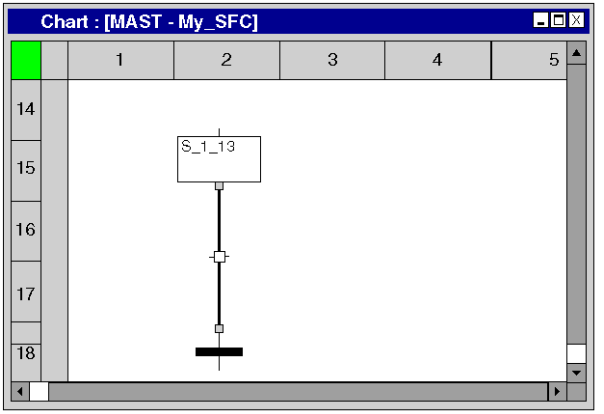
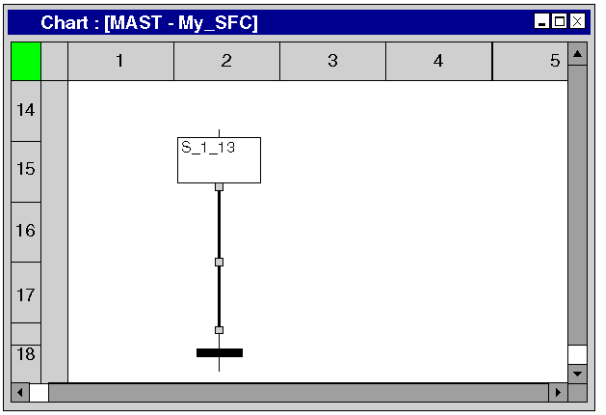
Process the following steps to create angles in links:

Step	Action
1	Select the link.
2	Place the mouse pointer on a change label. **Note:** The correct position is indicated by the mouse pointer symbol.

Step	Action
	Example: 
3	Click the left mouse button and drag the change label to the target position.
4	Release the mouse button. **Result:** The change is made. 

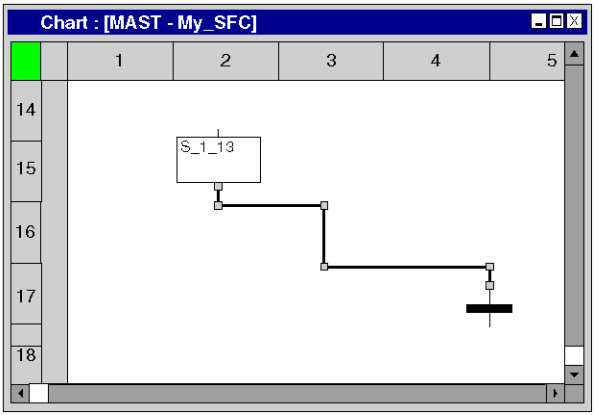
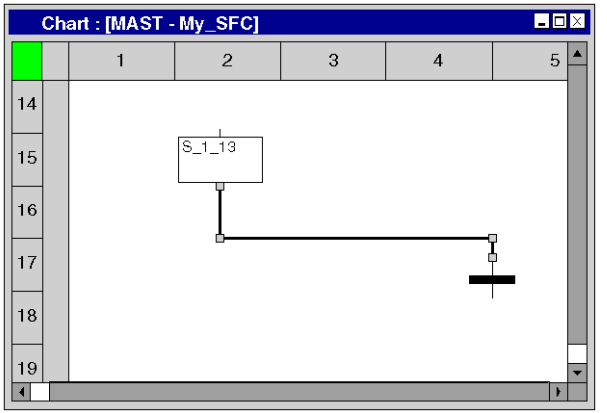
Creating a sizing handle

The creation of angles in links can only be made using sizing handles. If no sizing handles are available they can be created. Process the following steps to create sizing handles in links:

Step	Action
1	Select the link.
2	Press the **Ctrl** key and place the mouse pointer at the target position. **Note:** The correct position is indicated by the mouse pointer symbol. Example: 
3	Left-click the target position. **Result:** The sizing handle is created. 

Determining the optimum link route

Determining the optimum link route:

Step	Action
1	Select the link. Example: 
2	Use the **Edit > Link > Route** menu command. or Use the **Route** menu command from the shortcut menu. **Result:** The current link route is changed to the optimum route. Example: 

Entering Comments

Overview

This section describes how to enter comments in the SFC programming language.

Entering comments

Introduction

Comments can be placed in SFC in the form of text objects.

Text objects can overlap other objects.

The size of the object, depending on the size of the text, can be extended vertically and horizontally to fill additional grid units.

The same rules apply for entering text and navigating within text objects as for editing ASCII text in standard text editors. (To create a line break in text objects you must press the **Ctrl** +**Enter** key combination).

NOTE: Keep in mind that each change to a comment (e.g. change to a comment text, change to the size of the text object) makes it necessary to recreate the section involved (**Generation > Generate project**).

Displaying or hiding the comments

Use the  button in the toolbar to show or hide the comments.

The background color of the button in blue indicates that the comments are hidden.

NOTE: To select or place text objects, check that the  button in the toolbar is not activated.

Selecting the text object

You have the following options to select text objects:

- Use the **Edit > New > Comment** menu command.
- Use the **Comment** menu command from the shortcut menu.

- Press the **F8** key.

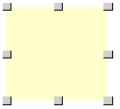
  or

- Select the  symbol.

The active placement mode for text objects is indicated by the  cursor symbol.

Placing text objects:

Placing text objects:

Step	Action
1	Select the text object.
2	Click the target position in the SFC section. or Use the **arrow keys** to move the cursor to the target position, and press **Enter**. **Result:** The selected text object is inserted and the selection mode is active again. Example: 
3	Enter the comment.
4	Confirm the text entered by: • Clicking with the mouse outside of the text object or • Pressing the **Enter** key.
5	To insert additional text objects: • Click the target position in the SFC section. or Use the **arrow keys** to move the cursor to the target position, and press **Enter**. • To return to selection mode: Press the **Esc** key. • To insert other objects: Select the object you wish to insert.

Online functions

Overview

This section describes the online functions of the SFC programming language.

Online functions

Online functions

The description of online functions is found in chapter Debugging in Sequential Function Chart Language (SFC), page 1443.

Export/Import

Overview

This section describes the Import/Export functions of the SFC programming language.

Export/Import SFC Sections

Export/Import

The description for exporting/importing sections is found in the chapter Import / Export, page 1686.

Customize SFC Editor Colors

Overview

This section describes how to customize the SFC editor colors.

Customize SFC Editor Colors

Overview

To customize SFC editor colors, open the **Tools > Colors Settings...** dialog box.

The **Color Settings** dialog box allows to:

- Modify the color of the selected item in the list.
- Reset the colors with initial values.
- Import an *.ini file that define all the colors associated to the SFC editor.
- Export the definition colors associated to the SFC editor.

Colors Setting File

Depending on the operating system of the PC, the colors settings file SFCColors.ini is located in one of the following folder:

- `c:\Program Files\Schneider Electric\Control Expert ••.•\`
- `c:\Program Files (x86)\Schneider Electric\Control Expert ••.•\`

For more detail on destination folder when Control Expert is installed, refer to *EcoStruxure™ Control Expert, Installation Manual*.

NOTE: Make a copy of this file before modifying it.

Description

This table describes the default color values for each parameters in the `SFCColors.ini` file:

Parameter	Default value (R, G, B)	Description
BkAnimation	`192,192,192`	Editor background color when animation is ON and you are connected in programming mode.
StatementError	`255,0,0`	Element in error is underlined with this color.
InspectBk	`0,255,255`	Inspect window background color
InspectText	`0,0,0`	Inspect window text color
InspectBkMin	`255,255,0`	Inspect window background color when the value is less than the minimum value defined in the inspect window settings.

Parameter	Default value (R, G, B)	Description
InspectBkMax	255,0,255	Inspect window background color when the value is greater than the maximum value defined in the inspect window settings.
BkAnimMonitoring	255,211,211	Editor background color when animation is ON and you are connected in monitoring mode.
AnalyzeError	0,0,255	Drawing color when an element generates an analyze error.
GraphAnimBoolTrue	0,150,0	Color of true BOOL in animation mode.
GraphAnimBoolFalse	255,0,0	Color of false BOOL in animation mode.
BkComment	255,255,204	Background color of comment area
BkCommentEdit	225,225,225	Background color of comment area when you are editing it.
BkAnimNumeric	255,255,0	Background color of numeric variables
BlockBreakpoint	128,0,0	Bullet color in the element where the breakpoint is set.
BlockCurrentStep	255,255,0	Color of the current element in debug mode
AnchorLink	128,0,0	Color of anchor link
BkAnimFrozen	128,255,255	Editor background color when animation is frozen (using SFC debug mode).
SFCStepTMax	255,0,255	Step color when the maximum time defined in the step is reached.
SFCStepTMin	255,255,0	Step color when the minimum time defined in the step is reached.
Background	255,255,255	Editor background color.

Common Function for Textual Language

What's in This Chapter

Selecting Text 1016
Deleting, cutting, copying, pasting and moving text 1019
Undoing and redoing changes 1021
Match case 1022
Zoom (font size) 1023
Using Bookmarks 1024
Go to 1026
Go to dialog box 1028
Searching text 1030
Searching and replacing text 1032
Call a subroutine 1034
Refine DFBs and subroutines 1035
Inspect windows 1037
Enter comments 1041
Printing the Current Section 1042
Initialize search 1043
Initialize Animation table 1044
Displaying the section properties 1046
Fullscreen 1046
Customize IL and ST Editor Colors 1047

Overview

This chapter describes the common menus and dialog boxes of the IL and ST editor.

Selecting Text

Selecting one or more characters

Selecting one or more characters:

Using the mouse	Using the keyboard
1. Left click with mouse and keep it pressed. 2. Drag the mouse across the characters you want to select. or 1. Place the insertion point in front of the first character you want to select. 2. Press the **Shift** key and keep it pressed. 3. Click the left mouse button behind the last character you want to select.	1. Press the **Shift** key and keep it pressed. 2. Use the arrow keys to move the insertion point over the characters to be selected.

Selecting a word

Selecting a word:

Using the mouse	Using the keyboard
Double click on the word to be selected.	1. Press the **Shift** key and keep it pressed. 2. Use the arrow keys to move the insertion point over the word to be selected.

Selecting one or more lines

Selecting one or more lines:

Using the mouse	Using the keyboard
1. Click the left mouse button at the height of the line to be selected in the left margin. or 1. Click the left mouse button at the height of the first line to be selected in the left margin. 2. Press and hold the mouse button down and drag the cursor down to the last line to be selected. or 1. Place the insertion point at the start of the first line you want to select. 2. Left click the mouse and keep it pressed. 3. Drag the cursor to the end of the last line to be selected. 4. Release the mouse key. or 1. Place the insertion point at the start of the first line you want to select. 2. Press the **Shift** key and keep it pressed. 3. Click the left mouse button behind the last line you want to select.	1. Press the **Shift** key and keep it pressed. 2. Use the arrow keys to move the insertion point over the characters to be selected.

Select all (entire contents of the section)

Select all (entire contents of the section):

Using the mouse	Using the keyboard
Use the menu command **Edit > Select all**. or Press the **Ctrl** key, and left mouse click in the left margin.	Press the key combination **Ctrl+A**.

Deleting, cutting, copying, pasting and moving text

Deleting Text

Deleting text:

Using the mouse	Using the keyboard
1. Select, page 1016 the text to be deleted. 2. Use the **Edit** > **Delete** menu command.	1. Select, page 1016 the text to be deleted. 2. Press the **Delete** key.

Cutting Text

Cutting text:

Using the mouse	Using the keyboard
1. Select, page 1016 the text to be cut. 2. Use the **Edit** > **Cut** menu command. or Use the **Cut** menu command from the shortcut menu (right-click). or Click the  symbol.	1. Select, page 1016 the text to be cut. 2. Press the **Ctrl**+**X** key combination.

The text which is cut can be pasted, page 1020 at any location (also in another IL or ST section).

Copying Text into the Clipboard

Copying text into the clipboard:

Using the mouse	Using the keyboard
1. Select, page 1016 the text to be copied. 2. Use the **Edit > Copy** menu command. or Use the **Copy** menu command from the shortcut menu (right-click). or Click the  symbol.	1. Select, page 1016 the text to be copied. 2. Press the **Ctrl+C** key combination.

The text which is copied can be pasted, page 1020 at any location (also in another IL or ST section).

Pasting Text from the Clipboard

Pasting text from the clipboard:

Using the mouse	Using the keyboard
1. Place the cursor at the target position. 2. Use the menu command **Edit > Insert**. or Use the menu command **Paste** in the shortcut menu (right mouse button). or Click the  symbol.	1. Place the cursor at the target position. 2. Press the **Ctrl+V** key combination.

Moving Text

Moving text:

Using the mouse	Using the keyboard
1. Select, page 1016 the text to be moved. 2. Position the mouse pointer on the selected text. 3. Click the left mouse button and keep it pressed. 4. Drag the text to the new position. **Note:** This can also be done between sections in various open IL and ST sections. 5. Release the mouse button. **Result:** The selected text is moved from the original position to the target position.	-

Copying text

Copying text:

Using the mouse	Using the keyboard
1. Select, page 1016 the text to be moved. 2. Position the mouse pointer on the selected text. 3. Click the left mouse button and keep it pressed. 4. Press the **Ctrl** key and keep it pressed. 5. Drag the text to the new position. **Note:** This can also be done between sections in various open IL and ST sections. 6. Release the mouse button first and then the **Ctrl** key. **Result:** A copy of the selected text is pasted at the target position.	-

Undoing and redoing changes

Undo changes

You have the following options to undo the recent changes:

- Use the **Edit > Undo** menu command.
- Press the **Ctrl**+**Z** key combination.

  or

- Select the  symbol.

One change is undone each time the function is executed.

The function can be executed a maximum of 10 times.

Redo changes

Changes that have been undone using **Undo** can be restored:

- Use the **Edit > Redo** menu command.
- Press the **Ctrl**+**Y** key combination.

  or

- Select the  symbol.

One undone change is restored each time the function is executed.

The function can be executed a maximum of 10 times.

Match case

All caps text

Process the following steps to display text in All Caps:

Step	Action
1	Select the text to display in All Caps.
2	You have the following options to convert the selected text: • Use the **Edit > All caps** menu command. • Press the **Shift**+**Alt**+**U** key combination. • Select the  symbol.

Lower case text

Process the following steps to display text in lower case:

Step	Action
1	Select the text to display in lower case.
2	You have the following options to convert the selected text: • Use the **Edit > Lower case** menu command. • Press the **Alt+U** key combination. • Select the  symbol.

Zoom (font size)

Introduction

Several font sizes are available to provide an optimal display of the text in the current section (overview of the entire contents or display of details).

Zoom out (reducing the font size)

To reduce current display (font size) by 25 %:

- Select the menu **View > Zoom > Zoom out**.
- Right click in the section and select **Zoom > Zoom out** for the section.
- Press the key combination **Ctrl+Minus (-)**.
- Click the  symbol.
- Click the  symbol.
- Using the mouse: press **Ctrl+Rotate the mouse wheel**.

Zoom in (increasing the font size)

To increase current display (font size) by 25 %:

- Select the menu **View > Zoom > Zoom in**.
- Right click in the section and select **Zoom > Zoom in** for the section.
- Press the key combination **Ctrl+Plus (+)**.
- Click the  symbol.

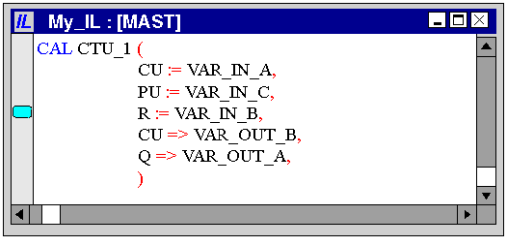
- Click the  symbol.
- Using the mouse: press **Ctrl**+**Rotate the mouse wheel**.

Using Bookmarks

Introduction

Bookmarks are used to mark text lines and allow these marked lines to be easily found in different sections.

An example of a line marked with a bookmark:



Setting a Single Bookmark

You have the following options to set a bookmark:

- Place the cursor in the line to be marked and carry out the menu command **Edit > Bookmarks> > Toggle Bookmark**.
- Place the cursor in the line to be marked and press the key combination **Ctrl+F2**.

  or
- Place the mouse pointer in the margin (gray column on the left edge of the section) for the line to be marked and use **Toggle Bookmark** from the context menu of the margin.

Setting Multiple Bookmarks

Setting a bookmark in each line which contains a defined search text can be done in the following ways:

- Use the menu command **Edit > Find...** to open the search dialog box, enter the search text and press the command button **Mark all**.

  or

- Press the key combination **Ctrl+F** to open the search dialog box, enter the search text and press the command button **Mark all**.

Also see Searching text, page 1030

Removing Single Bookmarks

You have the following options to remove a bookmark:

- Place the cursor in the line with the bookmark to be removed and carry out the menu command **Edit > Bookmark> > Toggle Bookmark**.
- Place the cursor in the line with the bookmark to be removed and press the key combination **Ctrl+F2**.

  or

- Place the mouse pointer on the bookmark to be removed and use **Toggle Bookmark** from the context menu of the bookmark.

Removing All Bookmarks

You have the following options to remove all bookmarks in the current section:

- Use the **Edit > Bookmarks > Clear all Bookmarks** menu command.

  or

- Press the **Ctrl+Shift+F2** key combination.

Searching for Bookmarks (Search Down)

You have the following options to search for bookmarks (search down):

- Use the **Edit > Bookmarks > Next Bookmark** menu command.
- Use the **Edit > Go to > Bookmark > Next** menu command.

  or

- Press the **F2** key.

The bookmarks are traversed according to their order of appearance in the **Bookmarks** dialog.

Searching for Bookmarks (Search Up)

You have the following options to search for bookmarks (search up):

- Use the **Edit > Bookmarks > Previous Bookmark** menu command.
- Use the **Edit > Go to > Bookmark > Previous** menu command.

  or

- Press the **Shift+F2** key combination.

The bookmarks are traversed according to their order of appearance in the **Bookmarks** dialog.

Bookmarks Dialog

Please also refer to Bookmarks Dialog, page 618.

Go to

Introduction

The **Go to** function is used to jump to:

- a specific line in the current section,
- a specific bookmark in the current section or
- a specific mark in the current section,

Open the dialog

The following options are available to open the**Go to**dialog box.

- Use the menu command **Edit > Go to**,
- select the menu command **Go to** from the sections shortcut menu.
- Press the key combination **Ctrl**+**G**.

  or

- Choose the  symbol.

Go to a line

Go to a certain line in the current section by carrying out the following steps:

Step	Action
1	Open, page 1026 the **Go to**dialog box.
2	Select the **Line** tab.
3	Enter the line number.
4	Confirm the entry using the command button **Go to**. **Result:** The desired line is displayed.

Go to a bookmark

Go to a certain bookmark in the current section by carrying out the following steps:

Step	Action
1	Open, page 1026 the **Go to**dialog box.
2	Choose the **Bookmark** tab.
3	Select a bookmark in the list.
4	Double click on the selected bookmark or press the **Go to**command button. **Result:** The desired bookmark is displayed.

Go to a location

Go to a certain location in the current section by carrying out the following steps:

Step	Action
1	Open, page 1026 the **Go to**dialog box.
2	Select the **Location** tab.
3	Select a location in the list.
4	Double click on the selected location or press the **Go to**command button. **Result:** The desired location is displayed.

Go to dialog box

Call the dialog box:

see Open the dialog, page 1026

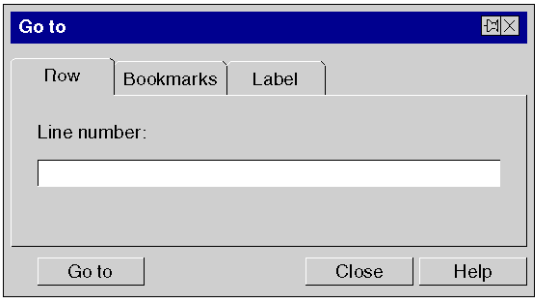
Call the dialog box

The **Go to**dialog box is made up of three tabs:

- **Row**
  Go to a certain location in the current line using this tab.
- **Bookmarks**
  Go to a certain bookmark in the current section using this tab:
- **Label**
  Go to a certain location in the current section using this tab:

tab Line

Representation of the**Line** tab:



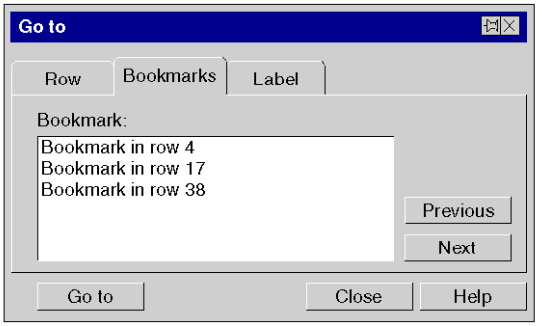
Elements of the tab **Line**:

Element	Description
	For symbols that are not printed, the dialog box is closed after every search. For printed symbols the dialog box remains open until you close it.
Line number	The number of the row to be displayed can be entered in this text box.

Element	Description
Go to	Display the current position using the button desired line, page 1027.
Close	Use this button to close the dialog.
Help	Use this button to call up the dialog help.

Register tab Bookmarks

Representation of the**Bookmark** tab:

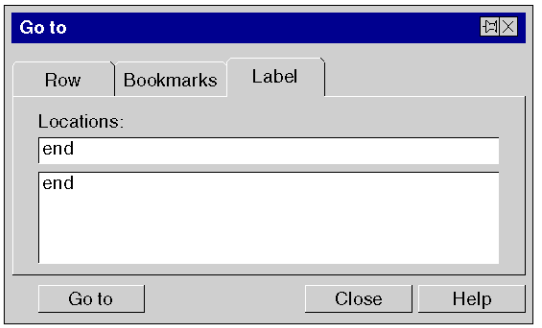


Elements of the tab **Bookmark**:

Element	Description
	For symbols that are not printed, the dialog box is closed after every search. For printed symbols the dialog box remains open until you close it.
Bookmark:	Select the bookmark to be displayed from this list box.
Prev	Display the current position using this button. When the start of the section is reached, the search starts again from the bottom.
Next	Display the next bookmark using this button. When the end of the section is reached, the search starts again from the top.
Go to	Display the current position using the button desired bookmark, page 1027.
Close	Use this button to close the dialog.
Help	Use this button to call up the dialog help.

Register tab Location

Representation of the tab **Label**:



Elements of the tab **Bookmark**:

Element	Description
 	For symbols that are not printed, the dialog box is closed after every search. For printed symbols the dialog box remains open until you close it.
Locations:	Select the location to be displayed from this text box. You can enter the name of the location in the text box with the keyboard or using the drop-down list by clicking with the mouse.
Go to	Display the current position using the button desired location, page 1027.
Close	Use this button to close the dialog.
Help	Use this button to call up the dialog help.

Searching text

Searching text

When searching text, you can define the search text and various options for the search.

You have the following options to search for text:

- Use the **Edit > Find** menu command.
- Press the key combination **Ctrl**+**F**.

  or

- click the symbol.

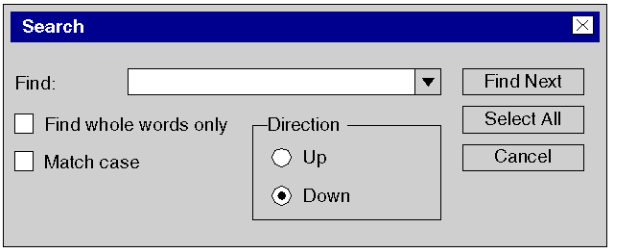
This function is only used for searching and in some cases marking search terms.

To search and replace text, use the **Edit > Replace** menu command, see also Searching and replacing text, page 1032.

For intelligent search and replace of variables and FFBs you can use the **Tools > Search/ Replace** menu command.

Representation

Search dialog display



Elements of the dialog box

Element	Description
Find:	Enter the text to be found. You can insert text from the clipboard or from the list of recently used terms.
Find whole words only	If you select this option, only the whole term is searched for and not parts of the term.
Match case	If you select this option, only terms are searched whose capitalization matches the text in the **Find what** text box.
Direction	Select the desired direction here.
Find Next	Use this command button to search for the next instance of the search target in the section.
Select All	Use this command button to mark all lines in the section that contain the search term with a bookmark, see also Using Bookmarks, page 1024.

Find Previous

You have the following options to find earlier instances of the term defined in the Find dialog box:

- Use the **Edit > Previous** menu command.

  or

- press the **Shift**+**F3** key combination.

Find Next

You have the following options to find later instances of the term defined in the Find dialog box:

- Use the **Edit > Find Next** menu command.

  or

- press the **F3** key.

Searching and replacing text

Searching and replacing text

You have the following options to search and replace text:

- Use the **Edit > Replace** menu command.

  or

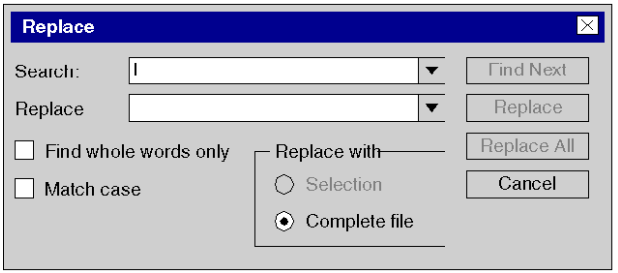
- Press the **Ctrl**+**H** key combination.

This function is used to search and replace simple text.

For intelligent search and replace of variables and FFBs you can use the **Tools > Search/ Replace** menu command.

Representation

Representation of the Find dialog box



Elements of the dialog box

Element	Description
Find what	Enter the text to be found. You can insert text from the clipboard or from the list of recently used terms.
Replace with	Enter the new text. You can insert text from the clipboard or from the list of recently used terms. In order to delete the text in the **Find what** text box, leave the **Replace with** text box empty.
Find whole words only	If you select this option, only the whole term is searched for and not parts of the term.
Match case	If you select this option, only terms are searched whose capitalization matches the text in the **Find what** text box.
Replace with	Select the area where the replace operation should be performed: • **Selection** The replace procedure is performed in the text area selected in the section. • **Complete file** The replace procedure is performed in the entire section.
Find Next	Use this command button to search for the next instance of the search target in the section. (No replacement is made.)
Replace	Use this command button to replace the selected occurrence of the search term and select the next occurrence.
Replace All	Use this command button to replace all occurrences of the search term in the area defined in **Replace with** with the replacement term.

Call a subroutine

Introduction

The subroutine to be called must be located in the same task as the section calling.

Subroutines can also be called from within subroutines.

Subroutine calls are expansions to IEC 61131-3 and must be explicitly enabled via the **Tools > Project Settings** dialog box, in the **Language extensions** tab by activating the **Allow subroutines** check box.

In IL, a subroutine call consists of the `CAL` operator, followed by the name of the subroutine section, followed by an empty parameter list (optional), e.g.

```
CAL SR_Name
```

or

```
CAL SR_Name ()
```

In ST, a subroutine call consists of the name of the subroutine section, followed by an empty parameter list, followed by a semicolon, e.g.

```
SR_Name () ;
```

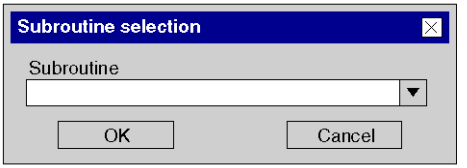
The name of subroutine is shown in boldface to identify that these subroutines can be refined, page 1035.

You can enter the name of the subroutine section just like any other text, or use the input assistant.

Entering the name of the subroutine section with input assistant

Process the following steps to enter a subroutine section name with the input assistant:

Step	Action
1	Place the cursor at the target position.
2	Use the **Edit > New > Select subroutine** menu command. or Select the  symbol. **Result:**

Step	Action
	A selection dialog box for selecting the subroutine to be called is opened. 
3	Click the ▾ symbol, select the subroutine to be called from the list and confirm pressing **Enter**. **Note:** The subroutine to be called must be located in the same task as the section calling. **Result:** The name of the subroutine section is enabled in the section.

4	Complete the subroutine call.
	IL example:
	`CAL MY_SR`
	ST example:
	`MY_SR () ;`

Refine DFBs and subroutines

Introduction

This function makes it possible to "look into" a DFB or subroutine. This makes it possible to see the internal state of the DFB/subroutine when animation is used, or to make changes to the logic very quickly.

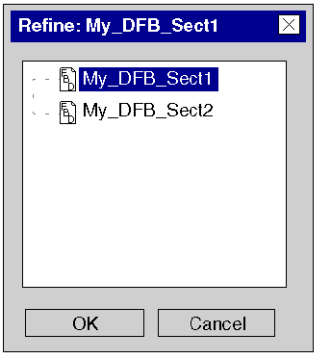
Refinable objects are labeled in boldface, e.g.
`CAL SR_Name`

or
`MY_DFB (IN1 := var1, IN2 := var12, OUT => var3);`

NOTE: The only limitation occurs when the displayed variable is in an ARRAY with at least 2 dimensions and when the index is not a constant. In this case an error message is displayed and the variable value can't be displayed.

Refining DFBs

Process the following steps to refine DFBs:

Step	Action
1	Place the cursor on the DFB instance name.
2	You have the following options to refine DFBs: • Use the **Services > Refine** menu command. • Use the **Refine** menu command from the shortcut menu. or • Press the **Ctrl+Q** key combination. **Result:** If the DFB contains only one section, this will be opened automatically. If the DFB contains several sections, a selection dialog is opened. Example: 
3	Select the desired DFB section. **Result:** The DFB section is opened. **Note:** With nested DFBs several **refine** processes are possible, but do not open simultaneously the refine editors or it could lead to an incorrect data display.

Refining subroutines

Process the following steps to refine subroutines:

Step	Action
1	Position the cursor on the subroutine call.
2	You have the following options to refine subroutines: • Use the **Services > Refine** menu command. • Use the **Refine** menu command from the shortcut menu. or • Press the **Ctrl**+**Q** key combination. **Result:** The subroutine is opened. **Note:** With nested subroutines several **refine** processes are possible.

Inspect windows

Introduction

In offline mode, the name, address (if available) and the comment (if available) of the associated variable are displayed in inspect windows.

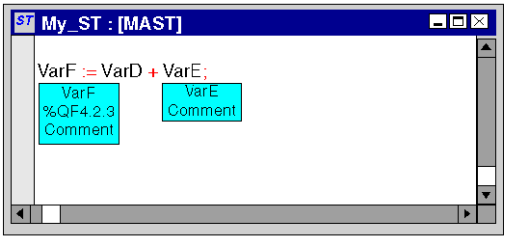
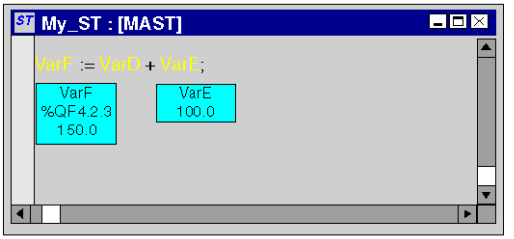
In online mode, the name, address (if available) and the current value of the variable are displayed in inspect windows.

The current values can be displayed:

- Decimal, e.g. `15`
- Binary, e.g. `00001111`
- Hexadecimal, e.g. `F`
- Floating point, e.g. `15.4`
- Exponent, e.g. `1.3e+003`
- Literal, e.g. `'otto'`

Generating an inspect window

Generating an inspect window:

Step	Action
1	Position the cursor in front of or in the target variable.
2	You have the following options to generate inspect windows: • Use the **Edit > New > Inspect window** menu command. • Use the **Place Inspect** menu command from the shortcut menu. • Press the **F9** key. or • Select the  symbol. **Result:** The inspect window is generated. Example (offline):  Example (online): 

Resizing

Position the mouse pointer on the right or left edge of an inspect window (the correct position is indicated by the ↙ or ↘ mouse pointer symbol), click the left mouse button and drag the mouse right or left.

The size selected applies to all inspect windows within the current section.

Changing the position

Position the mouse pointer on the inspect window, click the left mouse button and move the inspect window to the target position.

Anchoring the position

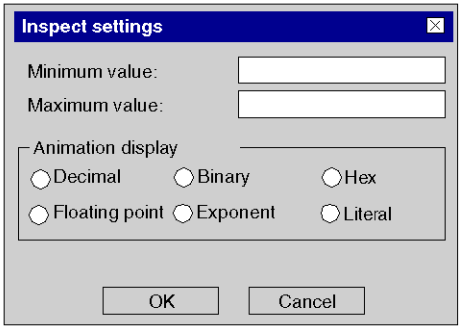
You can anchor all inspect windows of a section to their relative positions within the section (the inspect window stays in its onscreen position even if the page is scrolled), or to their absolute position (the inspect window is scrolled along with the rest of the section contents).

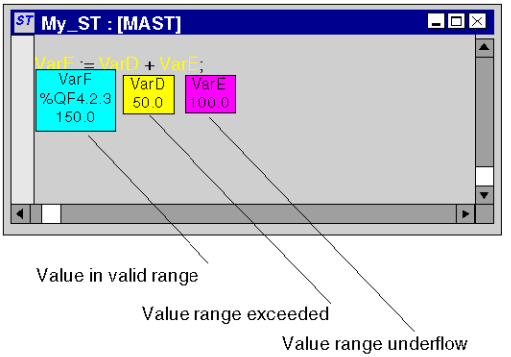
To switch between modes, select the **View > Floating Inspect Windows** menu command.

You can see which mode is active by way of a check-symbol in front of the menu command.

Setting a zone monitor

Setting a zone monitor:

Step	Action
1	Position the mouse pointer on the inspect window.
2	Use the **Settings** menu command from the shortcut menu. **Result:** The properties dialog box for the inspect window is opened. Example: 
3	Enter a minimum and a maximum value for the variable to be monitored.

Step	Action
4	Enter the format of the variable displayed in the section.
5	Confirm the entries with **OK**. **Result:** • If the current value is within the defined range, the inspect window is displayed in cyan. • If the current value is outside the defined range, the inspect window is displayed in yellow. • If the current value is above the defined range, the inspect window is displayed in magenta. Example: 

Hiding inspect windows

To hide and show the inspect windows select the **View > Hide Inspect Windows** menu command.

You can see which mode is active (hide inspect windows) by way of a check-symbol in front of the menu command.

Deleting inspect windows

Position the cursor on the inspect window to be deleted and select the **Delete Inspect** menu command from the shortcut menu.

Enter comments

Introduction

In the IL and ST editors, comments start with the string `(*` and end with the string `*)`. Any comment can be entered between these two strings.

NOTE: Keep in mind that each change to a comment (e.g. change to a comment text, change to the size of the text object) makes it necessary to recreate the section involved (**Generation > Generate project**).

In accordance with IEC 61131-3, nesting of comments is not permitted. If it is necessary to nest comments you must enable them via the **Tools > Project Settings** dialog box, in the **Language extensions** tab by activating the **Allow nested comments** check box.

Placing a new comment

Placing a comment:

Step	Action
1	Place the cursor at the target position.
2	You have the following options for placing a comment: • Use the **Edit > New > Comment** menu command. • Use the **Comment** menu command from the shortcut menu. • Press the **F8** key. or • Select the  symbol. **Result:** The strings for the start and end of the comments are inserted. Example: `(*  *)`
3	Enter the comment between the two strings. Example: `(* Comment *)`

Executing Text Comments

Placing a comment:

Step	Action

1	Select the command text to be executed.
	Example:
	`CAL My_Count`
	`(CU:=Var1,R:=Var2,`
	`PV:=Var3,Q=>Var4)`

2	You have the following options for placing a comment:
	• Use the **Edit > New > Comment** menu command.
	• Use the **Comment** menu command from the shortcut menu.
	• Press the **F8** key.
	or
	• Select the  symbol.
	Result: The strings for the start and end of the comments are inserted.
	Example:
	`(*CAL My_Count`
	`(CU:=Var1,R:=Var2,`
	`PV:=Var3,Q=>Var4)*)`

Printing the Current Section

Introduction

There are the following ways to print the current section:

- Print the section using the menu command
- Print the section using the print documentation

Print the section using the menu command

Carry out the following steps to print the current section using the menu command:

Step	Action
1	Open the print dialog using: • the menu command **File > Print**, • or the key combination **Ctrl+P** or • the symbol  .
2	Press **OK** to start printing the current section.

Print the section using the print documentation

Carry out the following steps to print the current section using the print documentation:

Step	Action
1	Double-click on **Documentation** in the project browser. **Result:** An overview of your system is open, see also Documentation, page 1660
2	Select **Program > Tasks > MAST/FAST/AUX > ...** to print the section.
3	Select the menu command **Print** from the sections shortcut menu.
4	Press **OK** to start printing the current section.

Initialize search

Introduction

The **Initialize Search** function transfers the name of the selected element (search text) to the **Element** text box in the **Cross-References** window.

Selecting the search text

Process the following steps to select the search text:

Step	Action
1	Select the element to search for (search text). The following elements can be searched for: • Variable (To search for a variable, select the variable.) • EFB/DFB Instance (To search for an EFB/DFB instance, select the name of the EFB/DFB instance.)
2	Use the **Services > Initialize Search** menu command. or Select the menu command **Initialize search** from the sections shortcut menu. or Press the **Ctrl+U** key combination. **Result:** The search text is transferred to the **Element** text box in the **Cross-References** window.

Initialize Animation table

Introduction

The **Initialize Animation Table** transfers the selected variables to the Animation table, page 1481.

Selecting the variables

Process the following steps to select variables:

Step	Action
1	Select the variables to transfer to the Animation table. The following items can be selected: • single variable (To accept a variable, select the variable.) • several variables (To accept several variables, select the entire text area in which the variables are located.)
2	Use the **Services > Initialize Animation table** menu command. or Select the menu command **Animation Table** from the sections context menu. or Press the **Ctrl**+**T** key combination. **Result:** The variable(s) is/are transferred to the Animation table, page 1481.

Initialize Animation Table

Using **Initialize Animation Table** (**Ctrl**+**T**) you create an animation table with a default name e.g. `Table[ST Editor - Mixer : [MAST]]`. You can rename the table.

All the variables you selected are listed in the animation table.

Selecting a new set of variables and using **Initialize Animation Table** again, all new variables are added to this animation table (if you did not rename it).

Initialize New Animation Table

Using **Initialize New Animation Table** (**Ctrl**+**Shift**+**T**) you create a new animation table with an indexed default name e.g. `Table[ST Editor - Mixer : [MAST]1]`. You can rename the table.

All the variables you selected are listed in the table.

But if you click anywhere in the section and do not select any variable, you create a new empty animation table.

Every time you use **Initialize New Animation Table** you create a new animation table.

Displaying the section properties

View section properties

You have the following options to display the section properties dialog box, page 505:

- Use the **Edit > Data Selection...menu command**.
- Use the **Properties** from the shortcut menu.
- Press the **Alt**+**Enter** key combination.

View Data Properties

You have the following options to display the section properties dialog box, page 1202:

- using the data editor, page 348

1. Selecting one or several lines in a list editor
2. Using the shortcut menu select the **Properties** command.

- via the IL/ST section.

1. Selecting one or several element(s) in the IL/ST section.
2. From the context menu select the command **Data properties** or press **Ctrl** + **Enter**.

Fullscreen

Fullscreen

You have the following options to display the section application window in full screen mode:

- Select the **View > Full screen** menu.
- Press the key combination **Ctrl**-**F8**.
- Click the  symbol.

To end the full screen mode press **Esc** or use the menu.

Customize IL and ST Editor Colors

Overview

Each of IL and ST editors have its own color setting definition file.

To customize IL or ST editor colors, open the **Tools > Colors Settings...** dialog box.

The **Color Settings** dialog box allows to:

- Modify the color of the selected item in the list.
- Reset the colors with initial values.
- Import an *.ini file that defines the colors associated to the IL or ST editor.
- Export the definition colors associated to the IL or ST editor.

Color Setting Files

Depending on the operating system of the PC, the color setting files ILColors.ini and STColors.ini are located in one of the following folders:

- `c:\Program Files\Schneider Electric\Control Expert ••.•\`
- `c:\Program Files (x86)\Schneider Electric\Control Expert ••.•\`

For more detail on destination folder when Control Expert is installed, refer to *EcoStruxure™ Control Expert, Installation Manual*.

NOTE: Make a copy of this file before modifying it.

Description

This table describes the default color values for each parameter in both `ILColors.ini` and STColors.ini files:

Group	Parameter	Name	Default value (R, G, B)	Description
`Text`	`Foreground`	**Text Color**	0,0,0	Color of text
	`Background`	**Background Color Text**	255,255,255	Color of the background text
`Text Selection`	`Foreground`	**Text Selection Color**	255,255,255	Color of selected text

Group	Parameter	Name	Default value (R, G, B)	Description
	Background	**Text Selection Background Color**	0,0,128	Background color of selected text
Operator	Foreground	**Operator Color**	255,0,0	Color of operator
	Background	**Operator background color**	255,255,255	Background color of operator
Comment	Foreground	**Comment color**	0,128,0	Color for comment text
	Background	**Background comment color**	255,255,255	Background color for comment text
Keyword	Foreground	**Keyword color**	0,0,255	Color for keyword text
	Background	**Keyword background color**	255,255,255	Background color for keyword
Identifier	Foreground	**Identifier color**	0,0,0	Text color for identifier
	Background	**Identifier background color**	255,255,255	Background color for identifier text
Variable	Foreground	**Variable color**	0,0,0	Color for variable text
	Background	**Variable background color**	255,255,255	Background color for variable text
FFB	Foreground	**FFB color**	0,0,0	Color for FFB text
	Background	**FFB background color**	255,255,255	Background color for FFB text
Literal	Foreground	**Literal color**	0,0,0	Color for literal text
	Background	**Literal background color**	255,255,255	Background color for literal text
AnimBool-True	Foreground	**Animation bool true color**	0,140,0	Color of true BOOL in animation mode.
	Background	**Animation bool true background color**	192,192,192	Background color of true BOOL in animation mode.
AnimBool-False	Foreground	**Animation bool false color**	255,0,0	Color of false BOOL in animation mode.
	Background	**Animation bool false background color**	192,192,192	Background color of false BOOL in animation mode.

Group	Parameter	Name	Default value (R, G, B)	Description
AnimNoValue	Foreground	**Animation No Value color**	0,0,0	Color of any variable without value in animation mode
	Background	**Animation No Value background color**	255,255,0	Background color of any variable without value in animation mode
AnimForced	Foreground	**Animation forced color**	0,0,0	Color of any forced variable in animation mode
	Background	**Animation forced background color**	192,192,192	Background color of any forced variable in animation mode
AnimCyclic-Set	Foreground	**Animation Cycle Set color**	255,255,255	Color of cyclic set in animation mode
	Background	**Animation Cycle Set background color**	80,0,80	Background color of cyclic set in animation mode
BKAnimation	Foreground	**Background animation color**	255,255,255	Foreground color of background in animation mode
	Background	**Background animation background color**	192,192,192	Background color in animation mode
TokenError	Foreground	**Token error color**	255,0,0	Color of token error
	Background	**Token error background color**	255,255,255	Background color of token error
Statemen-tError	Foreground	**Statement error color**	255,0,0	Color of statement error marking
	Background	**Statement error background color**	255,255,255	Background color of statement error marking
AnimError	Foreground	**Animation error color**	255,255,0	Color of error in animation mode
	Background	**Animation error background color**	192,192,192	Background color of error in animation mode

IL Editor

What's in This Chapter

Structure of an IL program 1051
Creating an IL Program 1053
Syntax and Semantics Check during Programming 1054
Navigating with the keyboard 1059
Data input 1060
Calling an FFB 1066
Using Public Variables 1077
Online functions 1079
Reference Data Type in IL 1079
Export/Import IL Sections 1080

Overview

This chapter describes the specific menus and dialog boxes of the IL editor.

For a description of the syntax for the IL programming language, please refer to the Instruction List IL (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) chapter in the Reference manual.

Structure of an IL program

Introduction

An IL program (Instruction List) consists of a list of statements that are executed in sequence by the controller. Using these statements you can call function blocks, functions and procedures conditionally or unconditionally, process assignments and jumps conditionally or unconditionally within a section.

Properties of an IL program

Properties of an IL program:

- Statements consist of:
  - an operator,
  - with modifier if necessary,
  - an operand if required and
  - a comment if required.
  - In addition, every statement can be provided with a label.
- Every statement begins on a new line.
- Each line is limited to 300 characters.
- Line breaks are possible in statements (multi-line statements).
- Labels, symbols and comments can be placed anywhere in a section. (Comments can be entered anywhere where empty spaces are permitted.)
- A syntax and semantics check, page 1054 is performed immediately after the statement is entered. The result of this check is displayed in colored text.
- Syntactically or semantically incorrect sections can be saved.

Edit and view functions

Edit and view functions of the IL editor:

- Entering text using the insert or overwrite mode, page 1053
- Deleting text, page 1019
- Selecting text, page 1016 (characters, words, lines, all)
- Cutting, page 1019, copying, page 1019 and pasting, page 1020 text
- Moving text, page 1020 (also between different ST/IL sections)
- Undo, page 1021 and Redo, page 1022
- Go To function, page 1026
- Using bookmarks, page 1024
- Searching for text and marking the result, page 1030
- Searching and replacing text, page 1032
- Searching and replacing variables and function blocks and DFB instances
- Displaying text in selectable fonts and sizes

  The text display type can be selected from the **View > Select Font** menu command. This configuration applies to all sections of the IL and ST programming languages.
- Intelligent text indentation

  The text indentation of a new line is taken from the previous one.

- Display the position (line, column) of the cursor in the status bar.

Input assistants

Input assistant for the IL editor:

- Input assistant for variables, page 1060
- Input assistant for functions, function blocks and procedures, page 1066

Online functions

Online functions of the IL editor:

- Displaying actual values, page 1037.
- Setting breakpoints, page 1422
- Setting watchpoints, page 1427
- Step by Step, page 1424

Creating an IL Program

Writing in insert/overwrite mode

Text can be entered in insert or overwrite modes

- Insert mode

  The characters entered are inserted in the current position of the insert marker in addition to the existing characters.

- Overwrite mode

  The characters entered are inserted in the current position of the insert marker and overwrite the existing characters.

Switching between the modes is done using the **Insert** key.

The current mode is displayed in the status bar using **INS** for insert mode and **OVR** for overwrite mode.

Creating an IL program

Carry out the following procedures to create an IL program:

Step	Action
1	Creating an IL Section, page 502.
2	Enter the first operator (e.g. `LD`). **Note:** As soon as the text is entered several checks are made, such as syntax/semantic error checks, correct spelling of keywords and variable names etc. A detailed description of syntax is found in chapter Instruction List IL (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual. The results of the checks are indicated by a color folder, see also Syntax and Semantics Check during Programming, page 1054.
3	Use the **TAB** key or spaces to separate operands from operators.
4	Enter the operands (e.g. `il`).
5	Confirm the line with the **ENTER** key.
6	Repeat these steps until all instructions are entered. Example: LD i1 INT_TO_REAL ADD r4 ST r3

Syntax and Semantics Check during Programming

Introduction

Syntax and semantics check is performed directly when creating the program.

The result of this check is displayed in three formats:

- directly in the program section with colored text
- as tooltip if the cursor is placed on incorrect text.
- In the output window, if **Generate > Analyze** is selected.

Representation

Representation of colors and labels:

Labeling	Description	Example
blue	Keyword (e.g. LD, AND, ST,...) **Note:** Function block input operators (see also section Input operator requirements (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual) are not treated as keywords.	`LD i1`

red	Operators (that are not keywords)	`CAL My_Count`
		`(CU:=Var1,R:=Var2,`
		`PV:=Var3,Q=>Var4)`

green	Comment	`LD i1 (* Comment *)`
black	standard text (e.g. variable, address, label, ...)	`start: LD B`

bold	The name of DFBs and subroutines are shown in boldface to identify that these objects can be refined, page 1035.	`CAL My_DFB`
		`(IN1:=Var1,`
		`IN2:=Var2,`
		`OUT=>Var3)`
		
		`CAL MY_SUBROUTINE`

underlining	In DFB sections, the formal parameters of the DFBs are shown underlined.	`LD IN1`
		`AND IN2`
		`ST OUT`

red wavy line	Erroneous text:
	• Syntax error
	e.g. spelling errors in keywords, non-declared variables or FB instances, incorrect block call, unavailable block call (in the current controller plateform or in the library types), EFB type call instead of the instance,...
	• Semantic error
	e.g. division by zero, value range exceeded/underflow (recognized by the system when entered)
		`LDs i1`
		`CAL TON`
		`(IN:=Var1,`
		`PT:=Var2,`
		`Q=>Var3`
		`ET:=Var4)`

Inverted text	Text is selected	`LD i1`

Tooltips

If the cursor is placed over incorrect text, the tooltip is displayed with a brief description of the cause of the error. It will also be displayed in the output window after the analysis.

Error messages in the output window

Error message: `Accessed object is not a function`.

Cause of error	Error Correction	Example

Write error is present during accessing of a function.	Correct the write error.	Incorrect example:
		`LEO (IN1 := VarE, IN2 := VarF)`
		`ST VarG`
		
		Correct example:
		`LE (IN1 := VarE, IN2 := VarF)`
		`ST VarG`

Name of the function was already used for: • one variable • an FB instance • a DFB Type • an SR Section • a procedure or function in a user library	Rename the already used object.	-

Incorrect syntax used during access
• an FB
	Correct the syntax.	Incorrect example:
		`My_CTD (`
		`  CD := VarH,`
		`  LD := VarI,`
		`  PV := VarJ,`
		`  Q  => VarK,`
		`  CV => VarL)`
		
		Correct example:
		`CAL My_CTD (`
		`  CD := VarH,`
		`  LD := VarI,`
		`  PV := VarJ,`
		`  Q  => VarK,`
		`  CV => VarL)`
		
		Incorrect example:
		`CAL TON`
		`  (IN:=Var1,`
		`  PT:=Var2,`
		`  Q=>Var3`
		`  ET:=Var4)`
		
		Correct example:

Cause of error	Error Correction	Example
		CAL My_TON (IN:=Var1, PT:=Var2, Q=>Var3 ET:=Var4)

Error message: `Accessed object is not a function block.`

Cause of error	Error Correction	Example
Write error is present during accessing. • an FB	Correct the write error.	Incorrect example: CAL MyCTD (CD := VarH, LD := VarI, PV := VarJ, Q => VarK, CV => VarL) Correct example: CAL My_CTD (CD := VarH, LD := VarI, PV := VarJ, Q => VarK, CV => VarL)
Name of the function was already used for: • one variable • an FB instance • a DFB Type • an SR Section • a procedure or function in a user library	Rename the already used object.	-
Function block instances that are not defined in the data editor.	Define the FB on the data editor.	-
EFB that is not available for the current controller platform.	Change the EFB.	-
An EFB that is not in the libset of the current project.	Update the libset.	-

Navigating with the keyboard

Navigating with the keyboard

The following keys and key combinations are provided for navigation:

Key combinations	Motion
Cursor left	Moves the cursor to the left
Cursor right	Moves the cursor to the right
Cursor up	Moves the cursor up
Cursor down	Moves the cursor down
Ctrl+Cursor left	Moves the cursor one string left
Ctrl+Cursor right	Moves the cursor one string right
Ctrl+Cursor up	Scrolls one line up (the position of the cursor is not affected by this function)
Ctrl+Cursor down	Scrolls one line down (the position of the cursor is not affected by this function)
Home	Moves the cursor to the beginning of the current line
End	Moves the cursor to the end of the current line
Ctrl+Home	Moves the cursor to the beginning of the current section
Ctrl+End	Moves the cursor to the end of the current section
Page Up	Scrolls one page up (the cursor remains in its position on the screen)
Page Down	Scrolls one page down (the cursor remains in its position on the screen)
Ctrl+Alt+Page Up	Displays the previous section (including via the **View > Previous Section**menu).
Ctrl+Alt+Page Down	Displays the next section (including via the **View > Next Section**menu).
Alt+Enter	Opens the properties dialog box for the selected section.

Data input

Introduction

You have different options for using data in your IL section.

- Using already declared variables, page 1060
  - you can use a data selection dialog box,

    or
  - you can enter the variable name just like any other text.
- Use undeclared variables and declare them later, page 1061

  You can define variable names when creating your IL section and then declare all variables used.
- Use undeclared variables and declare them immediately, page 1062

  You can define variable names when creating your IL section and declare them immediately.
- Declaration of variables directly in the IL section., page 1063

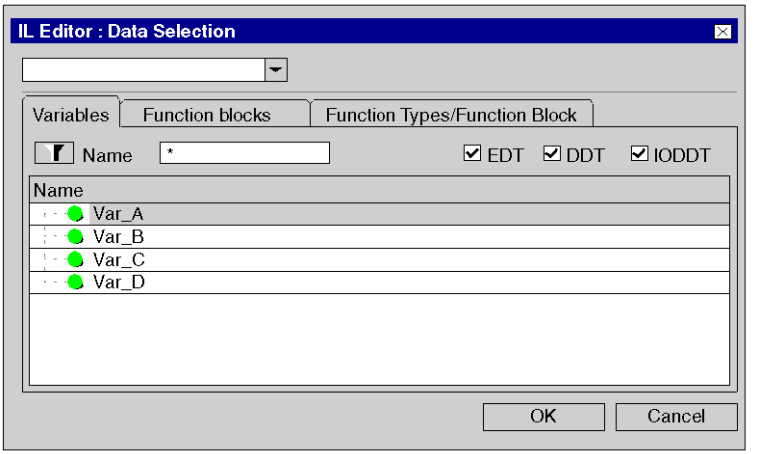
  You can declare variables in the Data Editor, page 390, but also directly in the IL section.
- Using formal parameters from function blocks (inputs/outputs, public variables), page 1064
  - You can use a data selection dialog box.

    or
  - You can enter the formal parameter name just like any other text.

Entering declared variables using data selection

Carry out the following steps to enter a declared variable using data selection:

Step	Action
1	Declare the required variables, see also Creation of EDT instances, page 390.
2	Place the cursor at the target position.

Step	Action
3	Open data selection by: • The **Edit > Data Selection menu command.**. • The **Data Selection... menu command**from the shortcut menu for the section. or • Press the **Ctrl+D** key combination. **Result:** A data selection box, page 1850 is opened. 
4	You have the following options to enter variables: • You can now enter the variable name and confirm using **Enter** or the button ✓. • You can select the variable names from the list of recently used names using the ▼ symbol and confirm using **Enter** or the button ✓. • You can delete your entry by pressing **Esc** or the button ✗. • Use the button **...**to open a variable selection dialog box, page 1192 and in the **Variables** tab confirm the selected variable using the **OK** button. Example of a data selection dialog box:  **Result:** The selected variable is inserted into the section and a syntax and semantics check, page 1054 is performed.

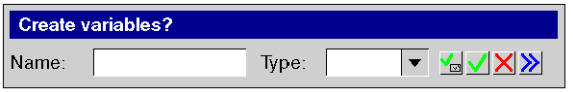
Entering undeclared variables and declaring them later

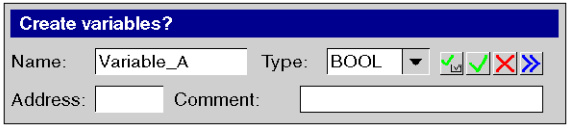
Process the following steps to enter undeclared variables and declare them later:

Step	Action
1	Place the cursor at the target position.
2	Enter the variable name. **Result:** The variable selected is accepted. The syntax and semantics check, page 1054 marks the variable name with a red wavy line to identify it as being undeclared, e.g. VarA
3	Enter all variables in this way.
4	Declare all variables in the Data Editor, page 390 or directly in the IL section, page 1063. **Result:** The red wavy line under the variable name disappears.

Entering undeclared variables and declaring them immediately

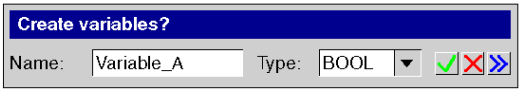
Process the following steps to enter undeclared variables and declare them immediately:

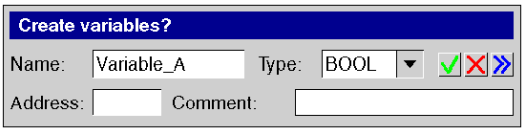
Step	Action
1	Place the cursor at the target position.
2	Click **Create Variable** in the shortcut menu, or press the **Shift**+**Enter** key combination **Result:** The variable declaration dialog box opens. 
3	Enter the variable name and data type.
4	If you do not want to assign an address or comment, confirm using **Enter** or the  symbol. **Result:** The variable is declared and inserted at the target position in the section.

Step	Action
5	If you do want to assign the variable an address and/or comment, use the  symbol for obtaining an advanced dialog and enter the address and/or comment. 
6	Confirm with **Enter** or the  symbol. **Result:** The variable is declared and inserted at the target position in the section.

Declaration of variables directly in the IL section.

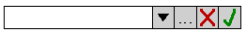
Process the following steps to declare variables directly in the IL section.

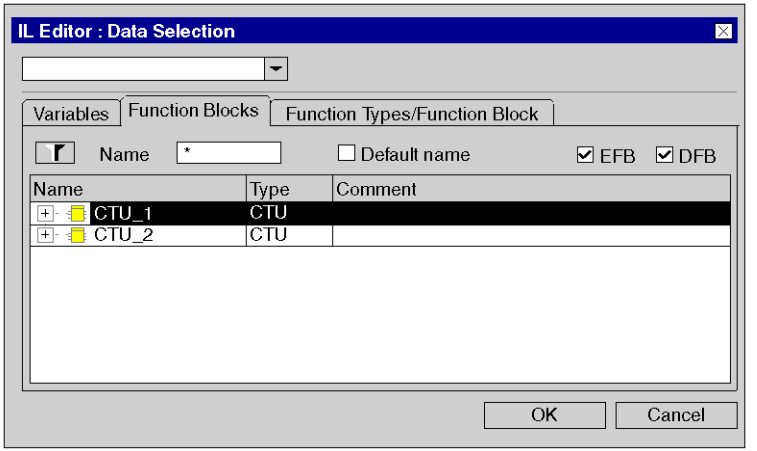
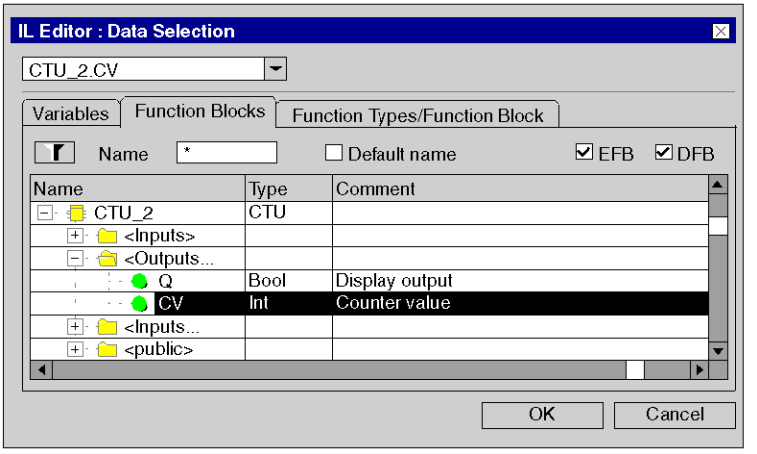
Step	Action
1	Place the cursor in an undeclared variable (undeclared variables are identified by a red wavy line).
2	Click **Create Variable** in the shortcut menu, or press the **Shift**+**Enter** key combination **Result:** The variable declaration dialog box opens.  **Note:** The variable name automatically appears in the **Name** field, if editor recognizes the text entered in the section as a variable and its syntax is correct.
3	If you do not want to assign an address or comment, confirm using **Enter** or the  symbol. **Result:** The variable is declared and the red wavy line under the variable name disappears.

Step	Action
4	If you do want to assign the variable an address and/or comment, use the » symbol for obtaining an advanced dialog and enter the address and/or comment. 
5	Confirm with **Enter** or the ✓ symbol. **Result:** The variable is declared and the red wavy line under the variable name disappears.

Using formal parameters

Process the following steps to use formal parameters with data selection:

Step	Action
1	Call the function block whose formal parameters you want to use, see also Calling a function block, page 1066.
2	To use the formal parameter, place the cursor at the target position.
3	Open data selection by: • The **Edit > Data Selection menu command.**. • The **Data Selection... menu command**from the shortcut menu for the section. or • Press the **Ctrl+D** key combination. **Result:** A data selection box, page 1850 is opened. 
4	You have the following options to enter the formal parameter: • You can enter the formal parameter name and confirm using **Enter** or the ✓ button. In this case the procedure ends here. • You can select the name from the list of recently used names using the ▼ symbol and confirm using **Enter** or the button ✓. In this case the procedure ends here. or • Use the button **...** to open a data selection dialog box, page 1192.
5	Select the **Function Blocks** tab.

Step	Action
	Result: All declared block instances are displayed. 
6	Expand the display of the inputs, outputs, and any inputs/outputs and public variables by clicking their **+** symbols. **Result:** You can now see all available inputs/outputs of the function block. 

Step	Action
7	Select the desired formal parameter and confirm with **OK**. **Result:** The selected formal parameter is inserted into the section and a syntax and semantics check, page 1054 is performed. For example: `LD CTU_2.CV`

Calling an FFB

Introduction

You have the following options to open FFBs (Elementary Function, Elementary Function Block, Derived Function Block, Procedure):

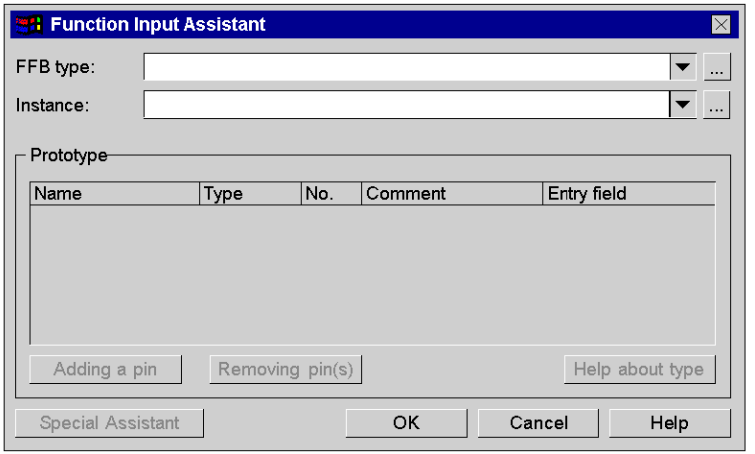
- You can enter the call like you enter any text (for syntax, see Calling FFB sections (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual),
- you can use the function input assistants,

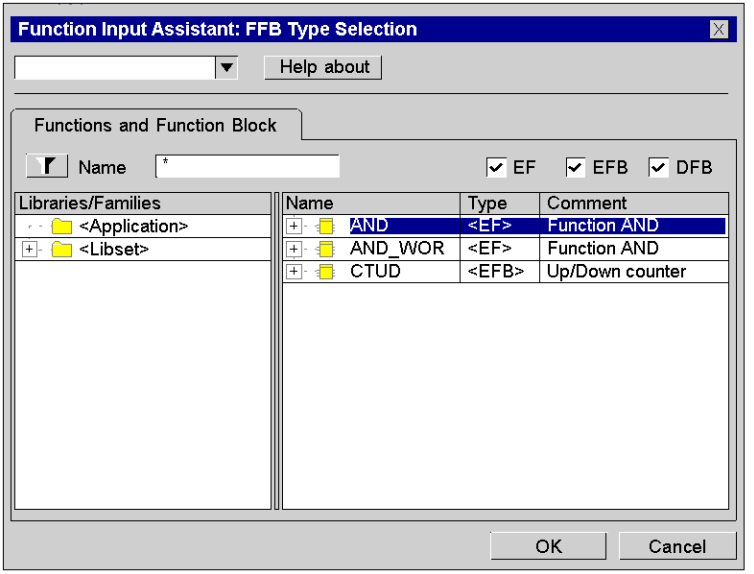
  or
- you can enter the FFB to the section using the drag & drop method.

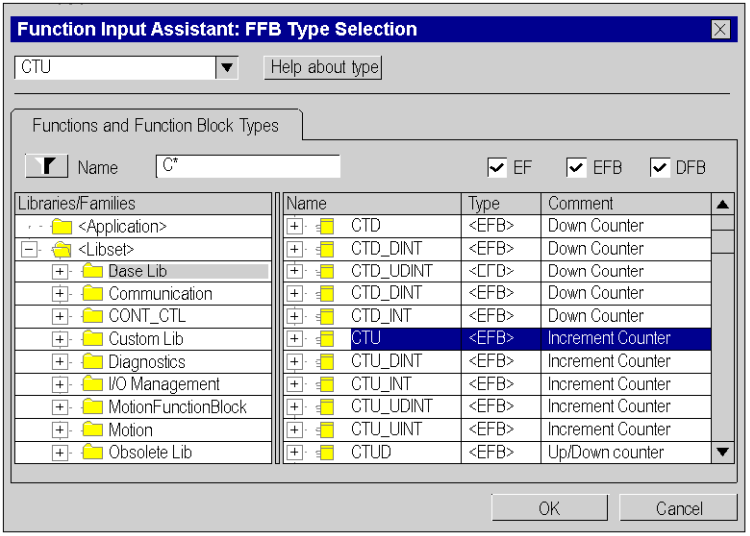
  NOTE: Procedures are expansions to IEC 61131-3 and must be explicitly enabled via the **Tools > Project Settings** dialog box, in the **Language extensions** tab by activating the **Allow procedures** check box.

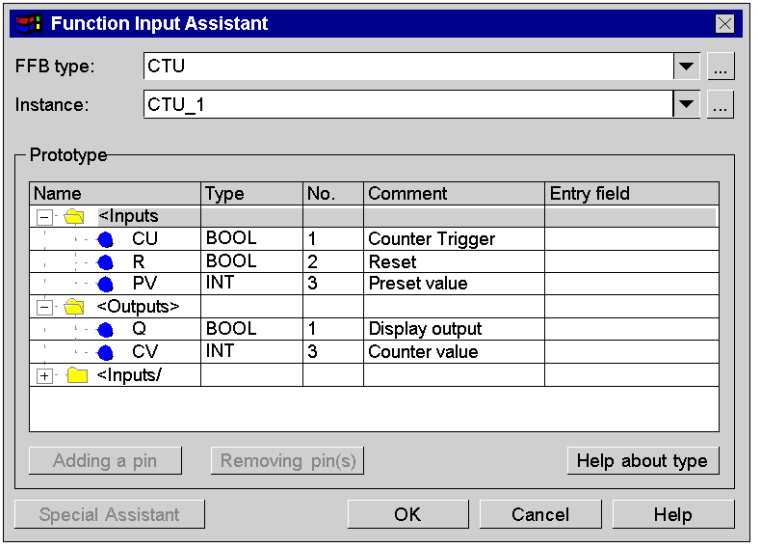
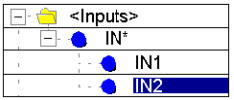
Calling a function block using the function input assistant

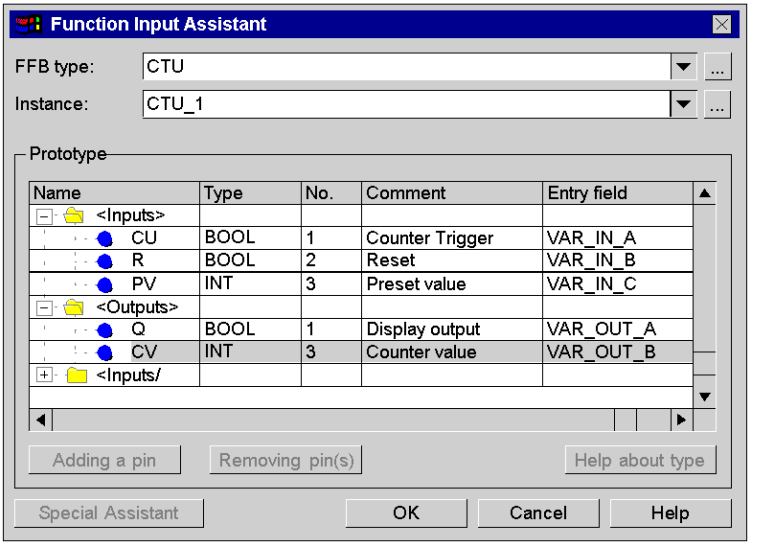
Process the following steps to call a function block (elementary or derived) with the function input assistant:

Step	Action
1	Place the cursor at the target position. Note: No text may be selected in the section. The following positions are possible: • On a free position outside of an existing FFB call. • On a space within an existing FFB call. (In this case, an FFB call is created within an FFB call.)
2	Open the function input assistant, page 1199 using: • the **Edit > FFB Input Assistant... menu command.**. • The **FFB Input Assistant...menu command**from the shortcut menu (right-click) for the section. • Pressing the **Ctrl+I** key combination. or • Select the  symbol. **Result:** The function input assistant will be opened. 
3	In the **FFB type** line click the button **....**, to open an FFB selection window, page 1196. **Result:**

Step	Action
	The FFB selection dialog box is opened. 
4	Select the library and family from the **Libraries/Families** column which contains the desired function block. **Tip:** If you do not know where to find the FFB you are looking for select the **Libset**, entry to see a list of all the FFBs available independently of their libraries.
5	Select the desired function block from the **Name** column.

Step	Action
	Example: 
6	Confirm the selection with **OK**. **Result:** • The function block is added to the function input assistant dialog box. • An instance name is automatically suggested. You can keep this or change it as you please. • You can see all available parameters (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) of the function block, see also Information about programming (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).

Step	Action
	
7	The number of inputs can be increased with some elementary functions. (You find out which functions can be expanded by referring to the descriptions for the individual functions.) Expandable pins can be recognized from the following display:  To add further pins select the last pin in the structure and press **Add pin**.

Step	Action
8	Double-click the **Entry field** cell of the first parameter and enter the name of the variable/address or an expression to be used. You have the following options to enter the variable/address: • You can now enter the variable name and confirm using **Enter**. • You can select the variable/address from the list of recently used variables/addresses using the ▼ symbol. or • Using the button **...** to open a variable selection dialog box, page 1193. Assign all parameters of the function block a variable/address in this way. Example: 

9	Confirm the entries using the **OK** button.
	
	Result: The function block call is inserted into the IL section and a syntax and semantics check, page 1054 is performed.
	
	Example:
	
	`CAL CTU_1 (`
	`CU := VAR_IN_A,`
	`R := VAR_IN_B,`
	`PV := VAR_IN_C,`
	`CV => VAR_OUT_B,`
	`Q => VAR_OUT_A`
	`)`

Calling a function or procedure using the function input assistant

Calling a function or procedure using the function input assistant is identical in principle to calling a function block. The only difference is the absence of an instance name (step 6 of the procedure Calling a function block using the function input assistant, page 1066).

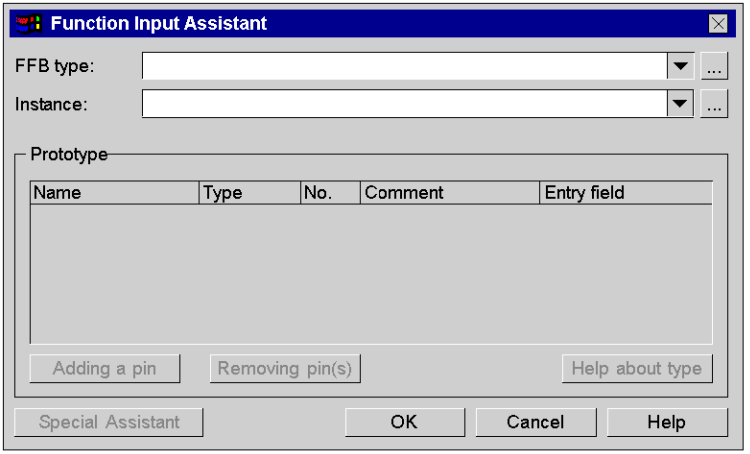
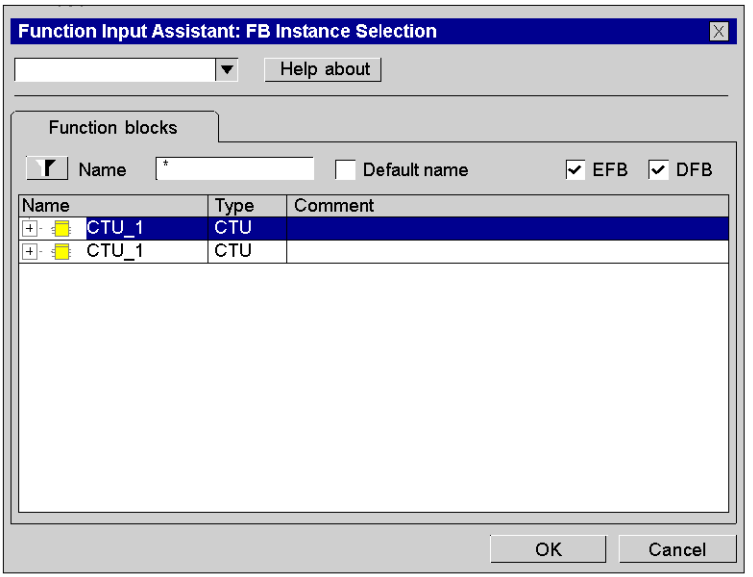
Remarks

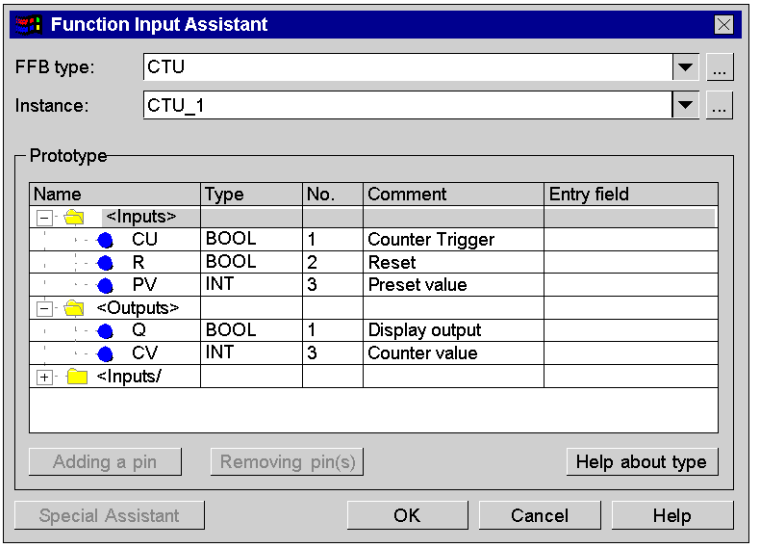
Some complex FFBs provide additional windows for entering parameters. Access to this window is gained using the **Special Assistant** button that appears at the bottom of the window.

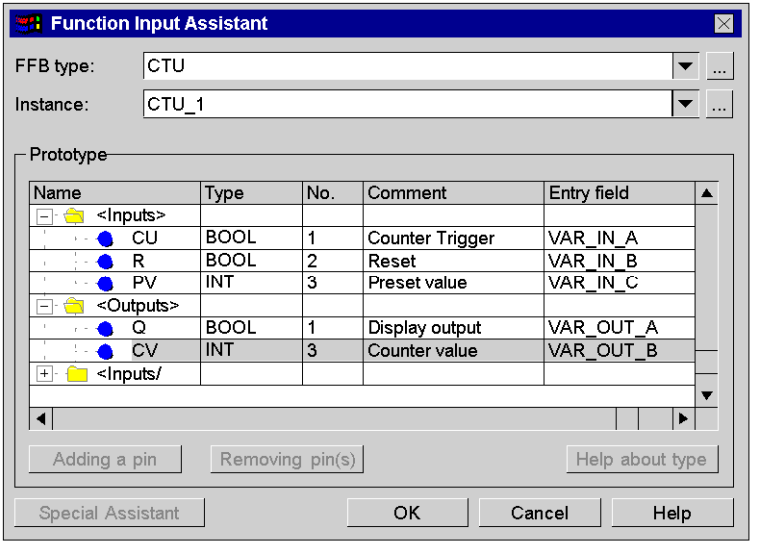
Calling a function block instance using the function input assistant

Process the following steps to select a block instance that has already been declared:

Step	Action
1	Place the cursor at the target position.
2	Open the function input assistant, page 1199 using: • The **Edit > FFB Input Assistant** menu command. • The **FFB Input Assistant...menu command**from the shortcut menu (right-click) for the section. • Pressing the **Ctrl+I** key combination. or • Select the  symbol. **Result:**

Step	Action
	The function input assistant will be opened. 
3	In the **Instance line** click the button **....**, to open an FFB selection window, page 1187. **Result:** The dialog box for selecting function block instances is opened. 
4	Select the desired block instance from the **Name** column and confirm the selection with **OK**. **Result:**

Step	Action
	• The block instance is added to the function input assistant dialog box. • You can see all available parameters (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) of the function block, see also Information about programming (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual). 
5	Double-click the **Entry field** cell of the first parameter and enter the name of the variable/address to be used. You have the following options to enter the variable/address: • You can now enter the variable name and confirm using **Enter**. • You can select the variable/address from the list of recently used variables/addresses using the ▼ symbol. or • Using the button **...** to open a variable selection dialog box, page 1193. Assign all parameters of the function block a variable/address in this way.

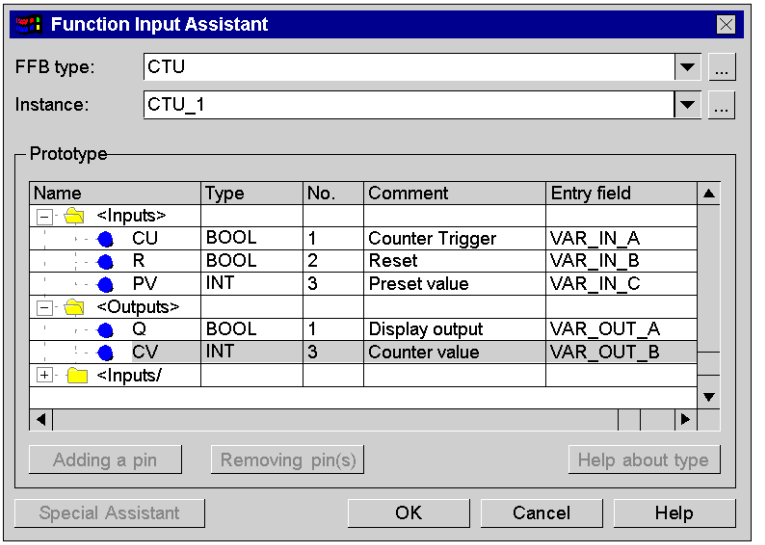
Step	Action
	Example: 

6	Confirm the entries using the **OK** button.
	
	Result: The function block call is inserted into the IL section and a syntax and semantics check, page 1054 is performed. If the variables are not declared yet a red wavy line under the name is used to identify them.
	
	Example:
	
	`CAL CTU_1 (`
	`CU := VAR_IN_A,`
	`R := VAR_IN_B,`
	`PV := VAR_IN_C,`
	`CV => VAR_OUT_B,`
	`Q => VAR_OUT_A`
	`)`

7	Declare all variables in the Data Editor, page 390 or directly in the IL section, page 1063. **Result:** The red wavy line under the variable name disappears.

Changing the actual parameter using the function input assistant

Process the following steps to change actual parameters with the function input assistant:

Step	Action
1	Place the cursor inside a word for the FFB to be changed. Note: No text may be selected and the FFB call to be changed must be syntactically correct.
2	Open the function input assistant, page 1199 using: • The **Edit > FFB Input Assistant... menu command.**. • The **FFB Input Assistant...menu command**from the shortcut menu (right-click) for the section. • Pressing the **Ctrl+I** key combination. or • Select the  symbol. **Result:** The function input assistant is opened with the current actual parameters. 
3	Make the desired changes and confirm using **OK**.

Inserting FFBs via drag & drop

FFBs can also be inserted into the section via drag & drop using the **Types Library Browser**.

You have the following options to activate the **Types Library Manager**:

- Use the **Tools > Types Library Manager** menu command.

  or

- Pressing the **Alt+3** key combination.

Using Public Variables

Introduction

In addition to inputs/outputs, some function blocks also provide public variables.

These variables transfer statistical values (values that are not influenced by the process) to the function block. They are used for setting parameters for the function block.

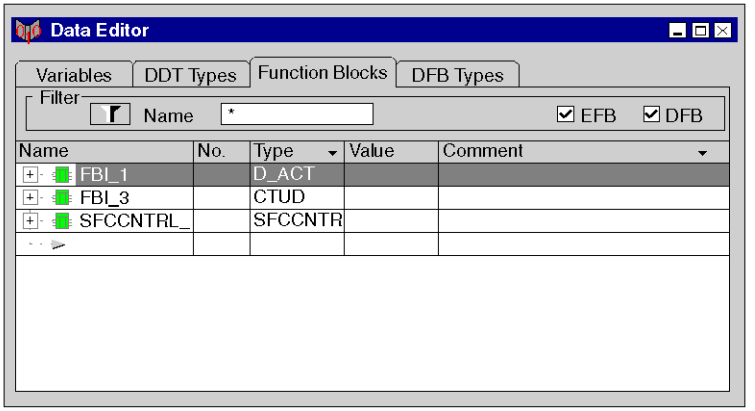
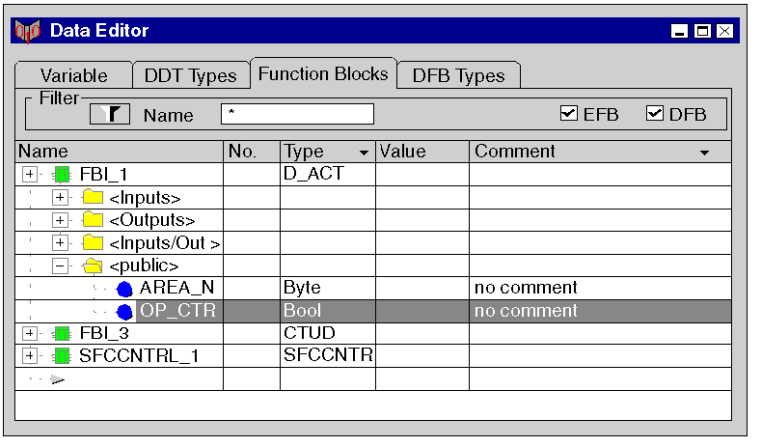
The assignment of values to public variables is made via their initial values or via the load and save instructions.

Public variables are read via the instance name of the function block and the names of the public variables.

Assigning Values using the Initial Value

Process the following steps to assign values using the initial value:

Step	Action
1	Call the function block for whose public variable you want to assign a value, see also Calling an FFB, page 1066.
2	Open the data editor, see also Data Editor Access, page 349.
3	Select the **Function Block** tab. **Result:**

Step	Action
	The function block instances are displayed. 
4	Expand the display of the public variables by clicking the respective **+** symbols. **Result:** You can now see all available public variables of the function block. 
5	Enter the desired value for the public variable in the **Value** box.

Assigning Values using the Load and Save Instructions

Carry out the following steps to assign values using the load and save instructions:

Step	Action
1	Call the function block for whose public variable you want to assign a value, see also Calling an FFB, page 1066.
2	Load the value to be assigned to the buffer. For example: `LD 1`
3	Save the buffer contents in a public variable, see also Using formal parameters, page 1064. For example: `ST D_ACT1.OP_CTRL`

Reading public variables

See Using formal parameters, page 1064

Online functions

Online functions

The description of online functions is found in chapter Debugging in Textual Languages (Structured Text, Instruction List), page 1421.

Reference Data Type in IL

Introduction

References to variables can be assigned in IL application programs.

A reference can be de-referenced in IL application programs.

For more information, refer to Reference Data Type (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).

IL Examples using the Reference Data Type:

myIL : [MAST]

```
REF (IN := myVarInt(*ANY*))
ST ptrToMyInt(*REF_ANY*)

MOVE (IN := REF(myVarBool)(*ANY*))
ST ptrToMyBool(*ANY*)

CAL TON_3  (IN := ptrToMyBool(*BOOL*),
            PT := t#5s(*TIME*),
            Q => output(*BOOL),
            ET => internalTime(*TIME*))
```

The EF, `REF`, takes a variable as an input parameter (type *ANY*) and provides its address as output parameter (type *REF_ANY*).

The EF, `MOVE`, can assign a reference to a variable to another reference. If the input parameter is (*REF(MyVariable)*), `MOVE` has the same effect as the `REF` EF.

The EF, `TON`, takes *ptrToMyBool^* as an input parameter to dereference the reference ptrToMyBool.

Export/Import IL Sections

Export/Import

The description for exporting/importing sections is found in the chapter Import / Export, page 1686.

ST Editor

What's in This Chapter

Structure of an ST program 1082
Creating an ST Program 1084
Syntax and Semantics Check during Programming 1085
Navigating with the keyboard 1089
Data input 1090
Calling an FFB 1097
Using public variables 1108
Calling a statement 1110
Online functions 1117
Reference Data Type in ST 1118
Export/Import IL Sections 1118

Overview

This chapter describes the specific menus and dialog boxes of the ST editor.

For a description of the syntax for the ST programming language, please refer to the Structured Text ST (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) chapter in the Reference manual.

Structure of an ST program

Introduction

An ST program (Structured Text) consists of a list of statements that are executed in sequence by the controller. Using these statements you can call function blocks, functions and procedures conditionally or unconditionally, make assignments, execute statements explicitly, repeat statements and execute jumps conditionally or unconditionally within a section.

Properties of an ST program

Properties of an ST program:

- Statements consist of:
  - an operator,
  - an operand,
  - an expression if required and
  - a comment if required.
  - In addition, every statement can be provided with a label.
- Statements must be concluded with semicolons.
- Several statements (separated by semicolons) may be present in one line.
- A single semi-colon denotes an empty statements (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).
- Each line is limited to 300 characters.
- Line breaks are possible in statements (multi-line statements).
- Labels, symbols and comments can be placed anywhere in a section. (Comments can be entered anywhere where empty spaces are permitted.)
- A syntax and semantics check, page 1085 is performed immediately after the statement is entered. The result of this check is displayed in colored text.
- Syntactically or semantically incorrect sections can be saved.

Edit and view functions

Edit and view functions of the ST editor:

- Entering text using the insert or overwrite mode, page 1084
- Selecting text, page 1016 (characters, words, lines, all)
- Deleting text, page 1019
- Cutting, page 1019, copying, page 1019 and pasting, page 1020
- Moving text, page 1020 (also between different ST/IL sections)
- Undo, page 1021 and Redo, page 1022
- Go To function, page 1026
- Using bookmarks, page 1024
- Searching for text and marking the result, page 1030
- Searching and replacing text, page 1032
- Searching and replacing variables and function blocks and DFB instances

- Displaying text in selectable fonts and sizes

  The text display type can be selected from the **View > Select Font** menu command. This configuration applies to all sections of the IL and ST programming languages.
- Intelligent text indentation

  The text indentation of a new line is taken from the previous one.
- Display the position (line, column) of the cursor in the status bar.

Input assistants

Input assistant for the ST editor:

- Input assistant for variables, page 1090
- Input assistant for functions, function blocks and procedures, page 1097
- Input assistant for statements, page 1110

Online functions

ST editor online functions:

- Displaying actual values, page 1037.
- Setting breakpoints, page 1422
- Setting watchpoints, page 1427
- Step by Step, page 1424

Creating an ST Program

Writing in insert/overwrite mode

Text can be entered in insert or overwrite modes

- Insert mode

  The characters entered are inserted in the current position of the insert marker in addition to the existing characters.
- Overwrite mode

  The characters entered are inserted in the current position of the insert marker and overwrite the existing characters.

Switching between the modes is done using the **Insert** key.

The current mode is displayed by **INS** for insert mode and **OVR** for overwrite mode.

Creating an ST program

Carry out the following procedures to create an ST program:

Step	Action
1	Creating an ST section, page 502.
2	Enter the first instruction (e.g. `VarA := VarB`). **Note:** As soon as the text is entered several checks are made, such as syntax/semantic error checks, correct spelling of keywords and variable names etc. A detailed description of syntax is found in chapter Structured Text (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual. The results of the checks are indicated by a color folder, see also Syntax and Semantics Check during Programming, page 1085.
3	Terminate the instruction with the closing operator (`;`).
4	Confirm the line with the **ENTER** key.

5	Repeat these steps until all instructions are entered.
	
	Example:
	
	`VarA := VarB*VarB-4*VarC*VarD ;`
	`IF VarA < 0.0 THEN VarE := 0 ;`
	`ELSIF VarA = 0.0 THEN`
	`VarE := 1 ;`
	`X1 := -X2/(2.0*VarC) ;`
	`ELSE`
	`VarE := 2;`
	`X1 := (-B+SQRT(D)) / (2.0*A) ;`
	`X2 := (-B-SQRT(D)) / (2.0*A) ;`
	`END_IF;`

Syntax and Semantics Check during Programming

Introduction

Syntax and semantics check is performed directly when creating the program.

The result of this check is displayed in three formats:

- directly in the program section with colored text
- as tooltip if the cursor is placed on incorrect text.
- In the output window, if **Generate > Analyze** is selected.

Illustration in the program section

Representation of colors and labels:

Labeling	Description	Example
blue	Keyword (e.g. `IF`, `FOR`, `WHILE`,...) **Note:** IL specific keywords (e.g. `LD`, `CAL`, `ST`) are not keywords in ST, but are still labeled as such for technical reasons.	`IF VarA`
red	Operators (that are not keywords)	`My_Count (CU:=Var1,R:=Var2, PV:=Var3,Q=>Var4);`
green	Comment	`IF VarA (* Comment *)`
black	standard text (e.g. variable, address, label, ...)	`start: IF VarB`
bold	The name of DFBs and subroutines are shown in boldface to identify that these objects can be refined, page 1035.	**`My_DFB`** `(IN1:=Var1, IN2:=Var2, OUT=>Var3);` **`MY_SUBROUTINE`** `();`
underlining	In DFB sections, the formal parameters of the DFBs are shown underlined.	`OUT := IN1 AND IN2;`
red wavy line	Erroneous text: • Syntax error e.g. spelling errors in keywords, non-declared variables or FB instances, incorrect block call, unvailable block call (in the current controller platform or in the library types), EFB type call instead of the instance,... • Semantic error e.g. division by zero, value range exceeded/underflow (recognized by the system when entered)	`IFs i1` `TON();`
Inverted text	Text is selected	`IF VarA`

Tooltips

If the cursor is placed over incorrect text, the tooltip is displayed with a brief description of the cause of the error. It will also be displayed in the output window after the analysis.

Error messages in the output window

Error message: `Accessed object is not a function.`

Cause of error	Error Correction	Example

Write error is present during accessing of a function.	Correct the write error.	Incorrect example:
		`VarG := LEQ (VarE, VarF);`
		Correct example:
		`VarG := LE (VarE, VarF);`

Name of the function was already used for: • one variable • an FB instance • a DFB Type • an SR Section • a procedure or function in a user library	Rename the already used object.	-

Incorrect syntax used during access
• an FB
• a procedure
• an SR Section
	Correct the syntax.	Incorrect example:
		`VarK := My_CTD (VarH, VarI, VarJ, VarL);`
		Correct example:
		`My_CTD (VarH, VarI, VarJ, VarK, VarL);`
		Incorrect example:
		`VarQ := DEC (VarQ);`
		Correct example:
		`DEC (VarQ);`
		Incorrect example:
		`VarA := My_SR ();`
		Correct example:
		`My_SR ();`

Error message: `Accessed object is not a function block.`

Cause of error	Error Correction	Example

Write error is present during accessing.
• an FB
• a procedure
• an SR Section
	Correct the write error.	Incorrect example:
		`MyCTD (VarH, VarI, VarJ, VarK, VarL);`
		Correct example:
		`My_CTD (VarH, VarI, VarJ, VarK, VarL);`
		Incorrect example:
		`DEZ (VarQ);`
		Correct example:
		`DEC (VarQ);`
		Incorrect example:
		`MySR ();`
		Correct example:
		`My_SR ();`

Name of the function was already used for: • one variable • an FB instance • a DFB Type • an SR Section • a procedure or function in a user library	Rename the already used object.	-
Function block instances that are not defined in the data editor.	Define the FB on the data editor.	-
EFB that is not available for the current controller platform.	Change the EFB.	-

Cause of error	Error Correction	Example
An EFB that is not in the libset of the current project.	Update the libset.	-

Incorrect syntax used during access.
• a function
• a function block
	Correct the syntax.	Incorrect example:
		`DIVTIME (VarR, VarS);`
		`DIVTIME (VarR, VarS, VarT);`
		Correct example:
		`VarT := DIVTIME (VarR, VarS);`
		Incorrect example:
		`CTD (VarH, VarI, VarJ, VarK, VarL);`
		Correct example:
		`My_CTD (VarH, VarI, VarJ, VarK, VarL);`
		Incorrect example:
		`TON ();`
		Correct example:
		`My_TON ();`

Navigating with the keyboard

Navigating with the keyboard

The following keys and key combinations are provided for navigation:

Key combinations	Motion
Cursor left	Moves the cursor to the left
Cursor right	Moves the cursor to the right
Cursor up	Moves the cursor up
Cursor down	Moves the cursor down
Ctrl+Cursor left	Moves the cursor one string left
Ctrl+Cursor right	Moves the cursor one string right

Key combinations	Motion
Home	Moves the cursor to the beginning of the current line
End	Moves the cursor to the end of the current line
Ctrl+Home	Moves the cursor to the beginning of the current section
Ctrl+End	Moves the cursor to the end of the current section
Page Up	Scrolls one page up (the cursor remains in its position on the screen)
Page Down	Scrolls one page down (the cursor remains in its position on the screen)
Ctrl+Alt+Page Up	Displays the previous section (including via the **View > Previous Section**menu).
Ctrl+Alt+Page Down	Displays the next section (including via the **View > Next Section**menu).
Alt+Enter	Opens the properties dialog box for the selected section.

Data input

Introduction

You have different options for using data in your ST section.

- Using already declared variables, page 1091
  - You can use a data selection dialog box.

    or
  - you can enter the variable name just like any other text.
- Use undeclared variables and declare them later, page 1092

  You can define the variable name when creating your ST section and then declare all variables used.
- Use undeclared variables and declare them immediately, page 1093

  You can define variable names when creating your ST section and declare them immediately.
- Declaration of variables directly in the ST section., page 1094

  You can declare variables in the Data Editor, page 390, but also directly in the ST section.

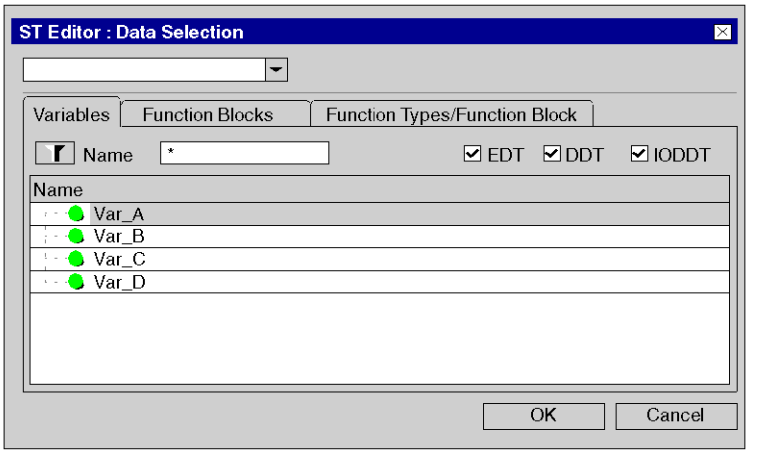
- Using formal parameters from function blocks (inputs/outputs, public variables), page 1095
  - You can use a data selection dialog box.

    or
  - You can enter the formal parameter name just like any other text.

Entering declared variables using data selection

Carry out the following steps to enter a declared variable using data selection:

Step	Action
1	Declare the required variables, see also Creation of EDT instances, page 390.
2	Place the cursor at the target position.

Step	Action
3	Open data selection by: • The **Edit > Data Selection menu command.**. • The **Data Selection... menu command**from the shortcut menu for the section. or • Press the **Ctrl**+**D** key combination. **Result:** A data selection box, page 1850 is opened. 
4	You have the following options to enter variables: • You can now enter the variable name and confirm using **Enter** or the button ✓. • You can select the variable names from the list of recently used names using the ▼ symbol and confirm using **Enter** or the button ✓. • You can delete your entry by pressing **Esc** or the button ✗. • Use the button **...**to open a variable selection dialog box, page 1192 and in the **Variables** tab confirm the selected variable using the **OK** button. Example of a variable selection dialog box:  **Result:** The selected variable is enabled in the section.

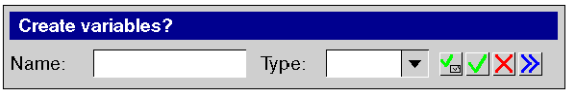
Entering undeclared variables and declaring them later

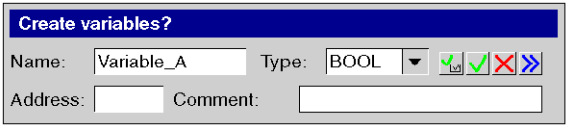
Process the following steps to enter undeclared variables and declare them later:

Step	Action
1	Place the cursor at the target position.
2	Enter the variable name. **Result:** The variable selected is accepted. The syntax and semantics check, page 1085 marks the variable name with a red wavy line to identify it as being undeclared, e.g. VarA
3	Enter all variables in this way.
4	Declare all variables in the Data Editor, page 390 or directly in the ST section, page 1094. **Result:** The red wavy line under the variable name disappears.

Entering undeclared variables and declaring them immediately

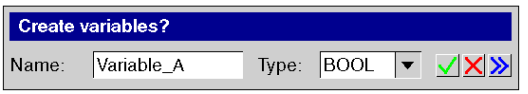
Process the following steps to enter undeclared variables and declare them immediately:

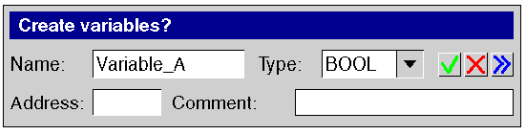
Step	Action
1	Place the cursor at the target position.
2	Click **Create Variable** in the shortcut menu, or press the **Shift**+**Enter** key combination **Result:** The variable declaration dialog box opens. 
3	Enter the variable name and data type.
4	If you do not want to assign an address or comment, confirm using **Enter** or the  symbol. **Result:** The variable is declared and inserted at the target position in the section.

Step	Action
5	If you do want to assign the variable an address and/or comment, use the symbol for obtaining an advanced dialog and enter the address and/or comment. 
6	Confirm with **Enter** or the symbol. **Result:** The variable is declared and inserted at the target position in the section.

Declaration of variables directly in the ST section.

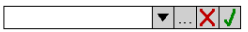
Process the following steps to declare variables directly in the ST section.

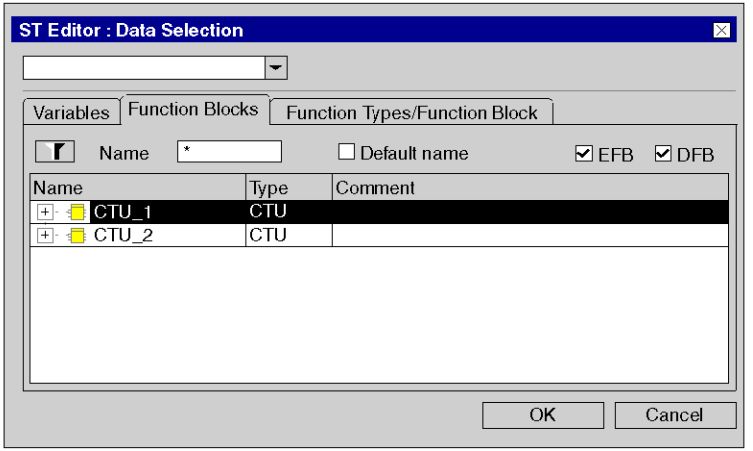
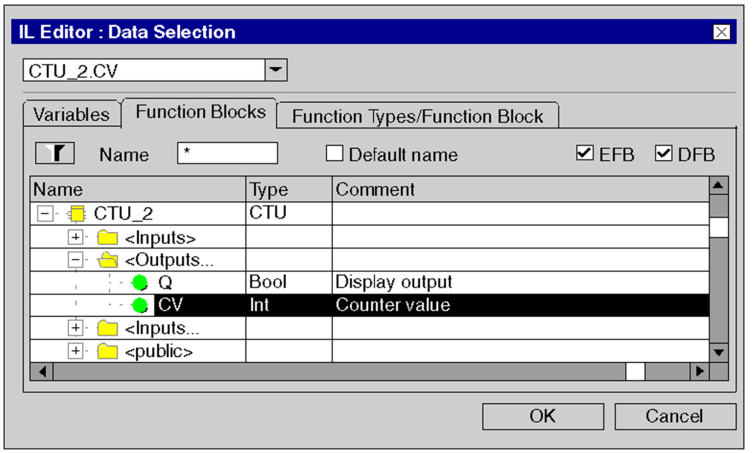
Step	Action
1	Place the cursor in an undeclared variable (undeclared variables are identified by a red wavy line).
2	Click **Create Variable** in the shortcut menu, or press the **Shift+Enter** key combination **Result:** The variable declaration dialog box opens.  **Note:** The variable name automatically appears in the **Name** field, if editor recognizes the text entered in the section as a variable and its syntax is correct.
3	If you do not want to assign an address or comment, confirm using **Enter** or the symbol. **Result:** The variable is declared and the red wavy line under the variable name disappears.

Step	Action
4	If you do want to assign the variable an address and/or comment, use the » symbol for obtaining an advanced dialog and enter the address and/or comment. 
5	Confirm with **Enter** or the ✓ symbol. **Result:** The variable is declared and the red wavy line under the variable name disappears.

Using formal parameters

Process the following steps to use formal parameters with data selection:

Step	Action
1	Call the function block whose formal parameters you want to use, see also Calling a function block, page 1097.
2	To use the formal parameter, place the cursor at the target position.
3	Open data selection by: • The **Edit > Data Selection menu command.**. • The **Data Selection... menu command**from the shortcut menu for the section. or • Press the **Ctrl**+**D** key combination. **Result:** A data selection box, page 1850 is opened. 
4	You have the following options to enter the formal parameter: • You can enter the formal parameter name and confirm using **Enter** or the ✓ button. In this case the procedure ends here. • You can select the name from the list of recently used names using the ▼ symbol and confirm using **Enter** or the button ✓. In this case the procedure ends here. or • Using the button **...** to open a data selection dialog box, page 1192.
5	Select the **Function Blocks** tab.

Step	Action
	Result: All declared block instances are displayed. 
6	Expand the display of the inputs, outputs, and any inputs/outputs and public variables by clicking their **+** symbols. **Result:** All available formal parameters are displayed now. 

Step	Action
7	Select the desired formal parameter and confirm with **OK**. **Result:** The selected formal parameter is inserted into the section and a syntax and semantics check, page 1085 is performed. For example: `IF CTU_2.CV = 35 THEN`

Calling an FFB

Introduction

You have the following options to open FFBs (Elementary Function, Elementary Function Block, Derived Function Block, Procedure):

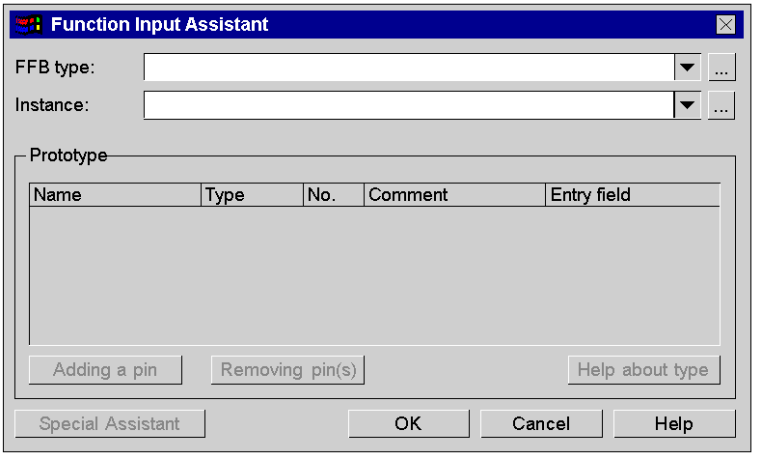
- You can enter the call like you enter any text (for syntax, see the section Calling FFBs (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual),
- You can use the function input assistants,

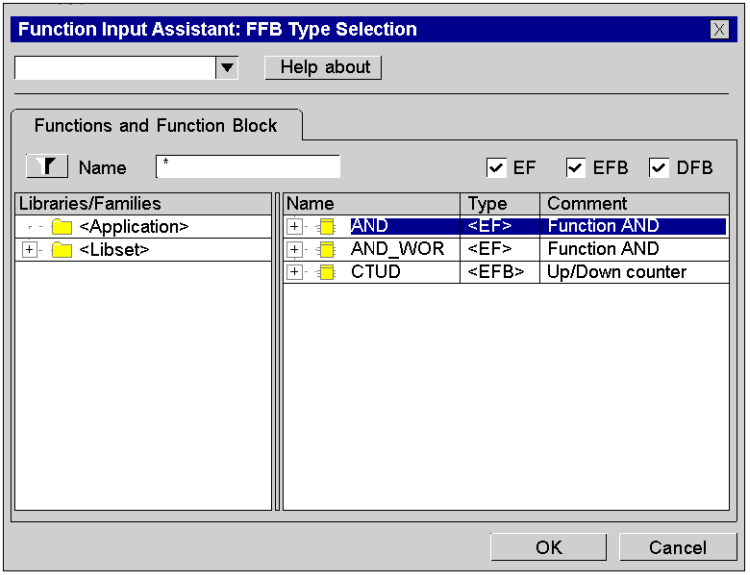
  or
- You can enter the FFB to the section using the drag & drop method.

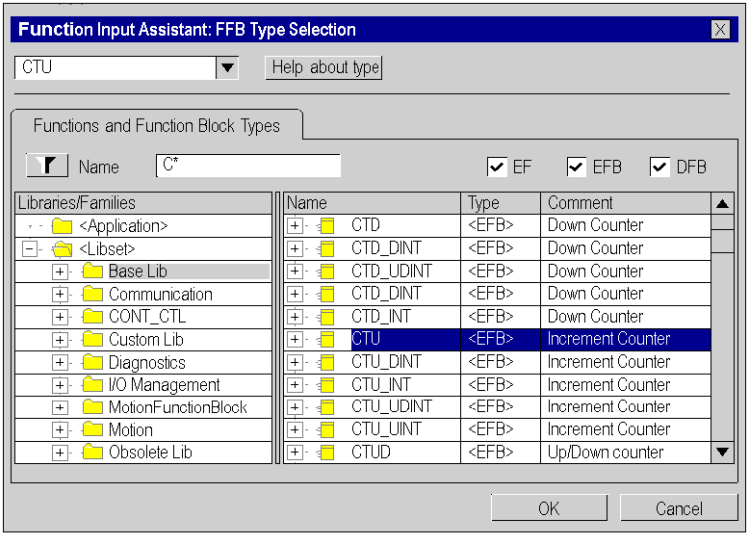
  NOTE: Procedures are expansions to IEC 61131-3 and must be explicitly enabled via the **Tools > Project Settings** dialog box, in the **Language extensions** tab by activating the **Allow procedures** check box.

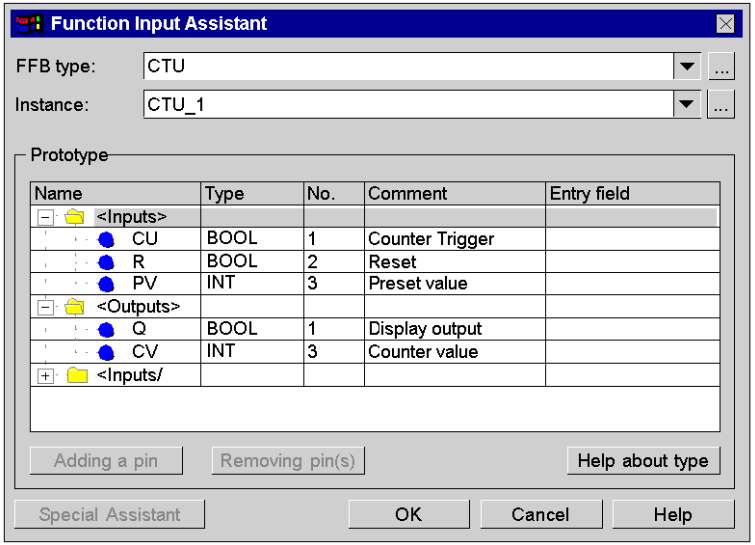
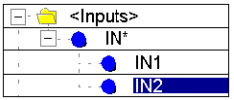
Calling a function block using the function input assistant

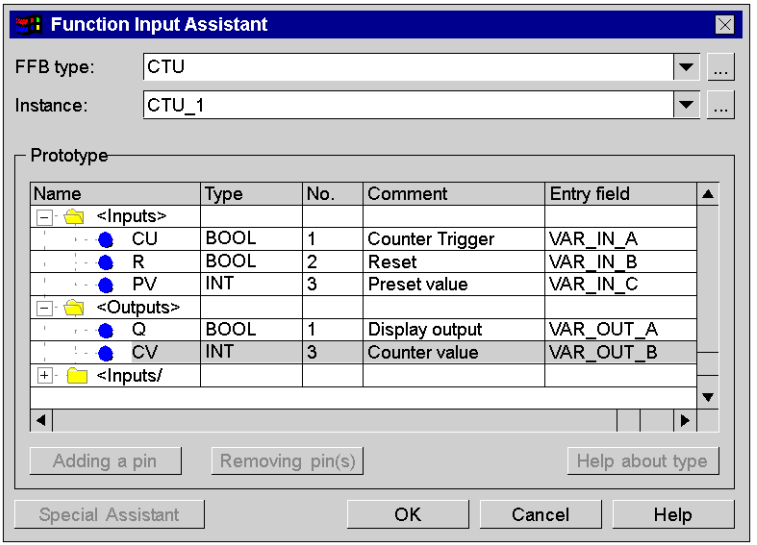
Process the following steps to call a function block (elementary or derived) with the function input assistant:

Step	Action
1	Place the cursor at the target position. Note: No text may be selected in the section. The following positions are possible: • On a free position outside of an existing FFB call. • On a space within an existing FFB call. (In this case, an FFB call is created within an FFB call.)
2	Open the function input assistant, page 1199: • The **Edit > FFB Input Assistant... menu command.**. • The **FFB Input Assistant...menu command**from the shortcut menu (right-click) for the section. • Press the **Ctrl+I**key combination. or • Select the symbol. **Result:** The function input assistant will be opened. 
3	Click the **FFB-Typ** button in the**....column**, to open an FFB selection window, page 1196. **Result:**

Step	Action
	The FFB selection dialog box is opened. 
4	Select the library and family from the **Libraries/Families** column which contains the desired function block. **Tip:** If you do not know where to find the FFB you are looking for select the **Libset**, entry to see a list of all the FFBs available independently of their libraries.
5	Select the desired function block from the **Name** column.

Step	Action
	Example: 
6	Confirm the selection with **OK**. **Result:** • The function block is added to the function input assistant dialog box. • An instance name is automatically suggested. You can keep this or change it as you please. • You can see all available parameters (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) of the function block, see also Information about programming (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).

Step	Action
	
7	The number of inputs can be increased with some elementary functions. (You find out which functions can be expanded by referring to the descriptions for the individual functions.) Expandable pins can be recognized from the following display:  To add further pins select the last pin in the structure and press **Add pin**.

Step	Action
8	Double-click the **Entry field** cell of the first parameter and enter the name of the variable/address or an expression to be used. You have the following options to enter the variable/address: • You can now enter the variable name and confirm using **Enter**. • You can select the variable/address from the list of recently used variables/addresses using the ▼ symbol. or • Using the button **...** to open a variable selection dialog box, page 1193. Assign all parameters of the function block a variable/address in this way. Example: 

9	Confirm the entries using the **OK** button.
	
	Result: The function block call is inserted into the ST section and a syntax and semantics check, page 1085 is performed.
	
	Example:
	
	`CTU_1 (CU := VAR_IN_A,`
	`R := VAR_IN_B,`
	`PV := VAR_IN_C,`
	`CV => VAR_OUT_B,`
	`Q => VAR_OUT_A);`

Calling a function or procedure using the function input assistant

Calling a function or procedure using the function input assistant is identical in principle to calling a function block. The only difference is the absence of an instance name (step 5 of the procedure Calling a function block using the function input assistant, page 1097).

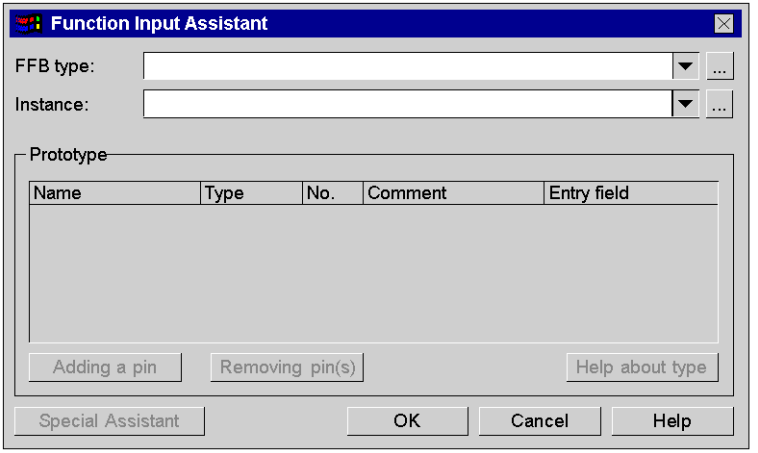
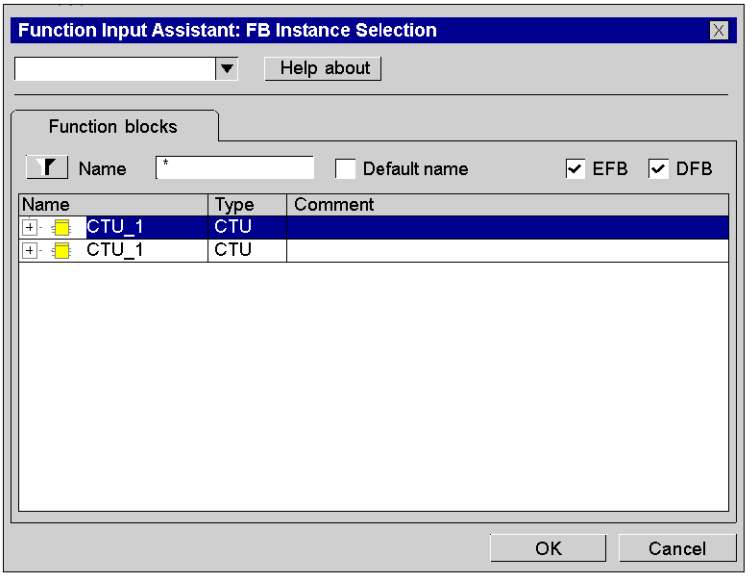
Remarks

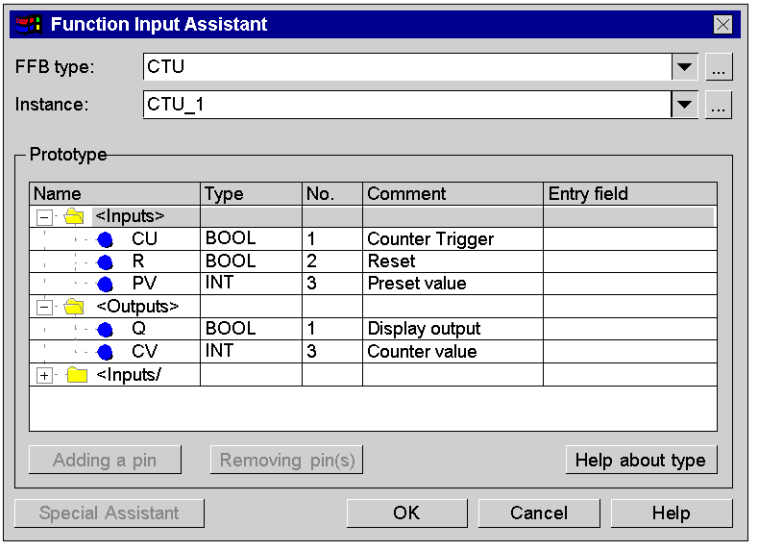
Some complex FFBs provide additional windows for entering parameters. Access to this window is gained using the **Special Assistant** button that appears at the bottom of the window.

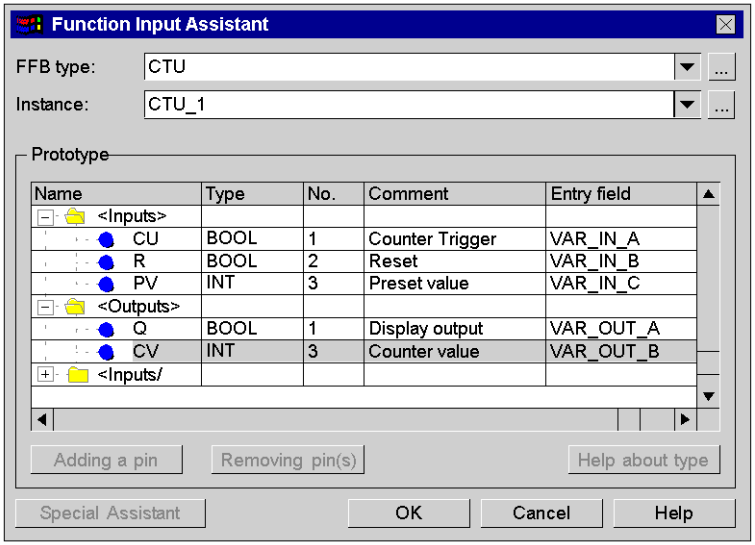
Calling a function block instance using the function input assistant

Process the following steps to select a block instance that has already been declared:

Step	Action
1	Place the cursor at the target position.
2	Open the function input assistant, page 1199: • The **Edit > FFB Input Assistant... menu command.**. • The **FFB Input Assistant...menu command**from the shortcut menu (right-click) for the section. • Press the **Ctrl**+**I**key combination. or • Select the  symbol. **Result:**

Step	Action
	The function input assistant will be opened. 
3	Click the **Instance** button in the**....column**, to open an FFB selection window, page 1187. **Result:** The dialog box for selecting function block instances is opened. 
4	Select the desired block instance from the **Name** column and confirm the selection with **OK**. **Result:**

Step	Action
	• The block instance is added to the function input assistant dialog box. • You can see all available parameters (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) of the function block, see also Information about programming (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual). 
5	Double-click the **Entry field** cell of the first parameter and enter the name of the variable/address to be used. You have the following options to enter the variable/address: • You can now enter the variable name and confirm using **Enter**. • You can select the variable/address from the list of recently used variables/addresses using the ▼ symbol. or • Using the button **...** to open a variable selection dialog box, page 1193. Assign all parameters of the function block a variable/address in this way.

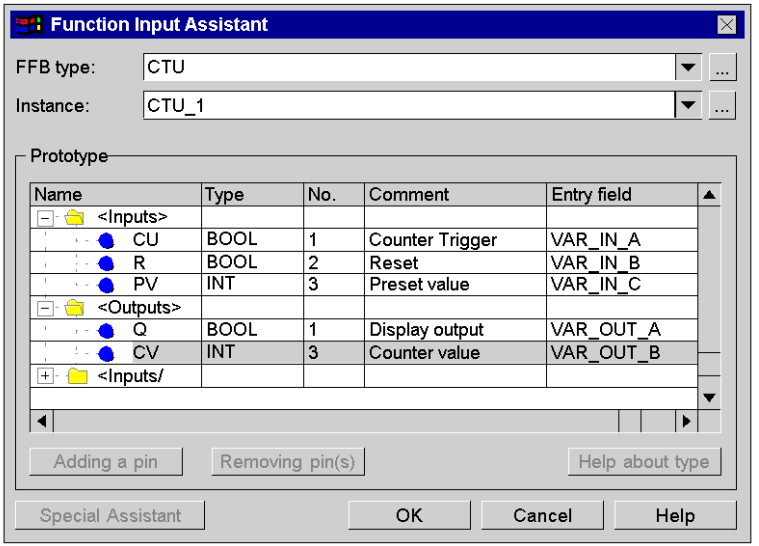
Step	Action
	Example: 

6	Confirm the entries using the **OK** button.
	
	Result: The function block call is inserted into the ST section and a syntax and semantics check, page 1085 is performed. If the variables are not declared yet a red wavy line under the name is used to identify them.
	
	Example:
	
	`CTU_1 (CU := VAR_IN_A,`
	`R := VAR_IN_B,`
	`PV := VAR_IN_C,`
	`CV => VAR_OUT_B,`
	`Q => VAR_OUT_A);`

7	Declare all variables in the Data Editor, page 390 or directly in the ST section, page 1094. **Result:** The red wavy line under the variable name disappears.

Changing the actual parameter using the function input assistant

Process the following steps to change actual parameters with the function input assistant:

Step	Action
1	Place the cursor inside a word of the FFB to be changed. Note: No text may be selected and the FFB call to be changed must be syntactically correct.
2	Open the function input assistant, page 1199: • The **Edit > FFB Input Assistant... menu command.**. • The **FFB Input Assistant...menu command**from the shortcut menu (right-click) for the section. • Press the **Ctrl+I**key combination. or • Select the symbol. **Result:** The function input assistant is opened with the current actual parameters. 
3	Make the desired changes and confirm using **OK**.

Inserting FFBs via drag & drop

FFBs can also be inserted into the section via drag & drop using the **Types Library Browser**.

You have the following options to activate the **Types Library Manager**:

- Use the **Tools > Types Library Manager** menu command.

  or

- Press the **Alt**+**3** key combination.

Using public variables

Introduction

In addition to inputs/outputs, some function blocks also provide public variables.

These variables transfer statistical values (values that are not influenced by the process) to the function block. They are used for setting parameters for the function block.

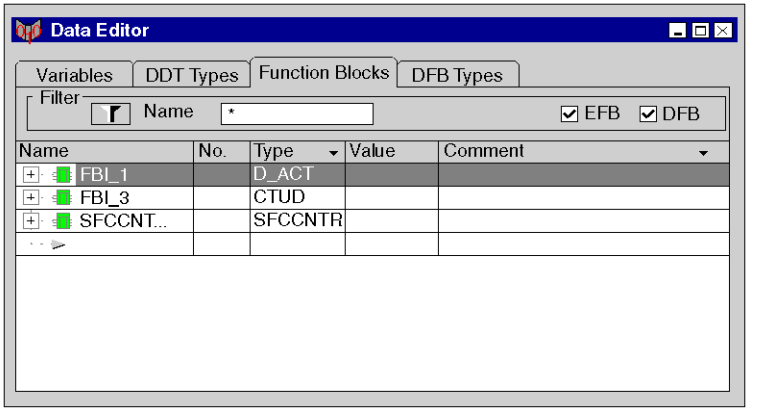
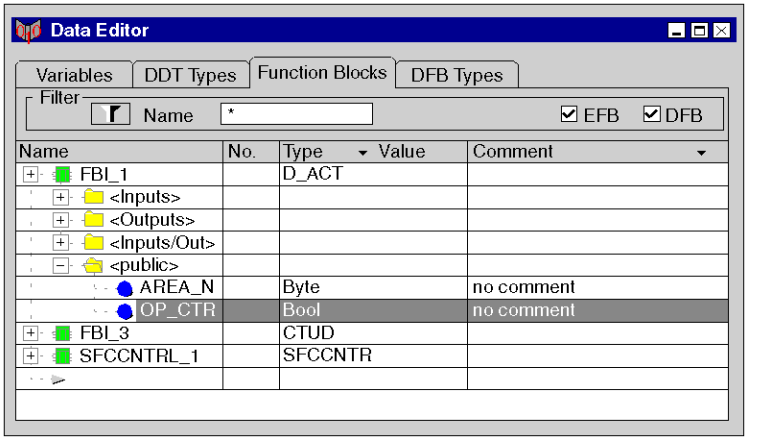
The assignment of values to public variables is made via their initial values or assignments.

Public variables are read via the instance name of the function block and the names of the public variables.

Assigning values using the initial value

Process the following steps to assign values using the initial value:

Step	Action
1	Call the function block for whose public variable you want to assign a value, see also Calling an FFB, page 1097.
2	Open the data editor, see also Data Editor Access, page 349.
3	Select the **Function Block** tab. **Result:**

Step	Action
	The function block instances are displayed. 
4	Expand the display of the public variables by clicking the respective **+** symbols. **Result:** You can now see all available public variables of the function block. 
5	Enter the desired value for the public variable in the **Value** box.

Assigning values using the assignment operator

Process the following steps to assign values using the assignment operator:

Step	Action
1	Call the function block for whose public variable you want to assign a value, see also Calling an FFB, page 1097.
2	Call the public variable, see also Using formal parameters, page 1095. For example: `D_ACT1.OP_CTRL`
3	Assign a value to the public variable. For example: `D_ACT1.OP_CTRL := 1 ;`

Reading public variables

See Using formal parameters, page 1095

Calling a statement

Introduction

You can call a statement (`IF`, `FOR`, `WHILE`, `REPEAT`, `CASE`) in the same way that you enter text (syntax, see section Statement (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in the Reference manual) or using an input assistant.

Selecting statements

Selecting statements:

State-ment	Description	Call via menu command	Call via symbol	Call via key combination
IF	The IF statement determines that a statement or a group of statements will only be executed if its related Boolean expression has the value 1 (true). If the condition is 0, the statement or the statement group will not be executed. See also in the Reference manual: IF...THEN...END_IF (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual), ELSE (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) and ELSIF...THEN (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual)	**Edit > New > IF Statement**		**F4**
FOR,	The FOR statement repeats a statement sequence until the END_FOR statement. The number of repetitions is determined by the start value, the end value and the control variable. See also in the Reference manual: FOR...TO...BY...DO... END_FOR (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual),	**Edit > New > FOR Statement**		**F5**
WHILE	The WHILE statement causes a sequence of statements to be executed repeatedly until its related Boolean expression is 0. If the expression is false right from the start, the group of statements will not be executed at all. See also in the Reference manual:	**Edit > New > WHILE Statement**		**F6**

State-ment	Description	Call via menu command	Call via symbol	Call via key combination
	WHILE...DO...END_WHILE (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual),			
REPEAT	The REPEAT statement causes a sequence of statements to be executed repeatedly (at least once) until its related Boolean condition is 1. See also in the Reference manual: REPEAT...UNTIL...END_REPEAT (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual),	**Edit > New > REPEAT Statement**		**Shift + F6**
CASE	The CASE statement consists of an INT data type expression (the "selector") and a list of statement groups. Each group is provided with a label which consists of one or several integer numbers (INT, DINT, UINT, UDINT) or areas of integer values. The first group of statements is executed, whose label contains the calculated value of the selector. Otherwise none of the statements will be executed. See also in the Reference manual: CASE...OF...END_CASE (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual),	**Edit > New > CASE Statement**		**Shift + F4**

Calling a IF statement using the input assistant

Process the following steps to call a IF statement with the input assistant:

Step	Action
1	Place the cursor at the beginning of an empty line.

2	Select the input assistant for `IF` statements, see also Selecting statements, page 1110.
	Result:
	An empty `IF` statement is inserted into the section.
	`IF THEN`
	`ELSEIF THEN`
	`ELSE`
	`END_IF;`

3	Complete the `IF` statement with your data.
	Result: A syntax and semantics check, page 1085 is performed immediately after the statement is entered.
	Note: An input assistant is also provided for entering variables, page 1090 and calling FFBs, page 1097 (functions, function blocks and procedures).
	For example:
	`IF A>B THEN`
	`C:=SIN(A) * COS(B);`
	`ELSIF A=B THEN`
	`C:=ADD(A,B);`
	`ELSE`
	`B:=MUL(C,A);`
	`END_IF;`

Calling a `FOR` statement using the input assistant

Process the following steps to call a `FOR` statement with the input assistant:

Step	Action
1	Place the cursor at the beginning of an empty line.

2	Select the input assistant for `FOR` statements, see also Selecting statements, page 1110.
	Result:
	An empty `FOR` statement is inserted into the section.
	`FOR TO BY DO`
	`END_FOR;`

3	Complete the `FOR` statement with your data.
	Result: A syntax and semantics check, page 1085 is performed immediately after the statement is entered.
	Note: An input assistant is also provided for entering variables, page 1090 and calling FFBs, page 1097 (functions, function blocks and procedures).
	For example:
	`FOR i:= 10 TO 1 BY -1 DO`
	`C:=SIN(A) * COS(B);`
	`END_FOR;`

Calling a `WHILE` statement using the input assistant

Process the following steps to call a `WHILE` statement with the input assistant:

Step	Action
1	Place the cursor at the beginning of an empty line.

2	Select the input assistant for `WHILE` statements, see also Selecting statements, page 1110.
	Result:
	An empty `WHILE` statement is inserted into the section.
	`WHILE DO`
	`END_WHILE;`

3	Complete the `WHILE` statement with your data.
	Result: A syntax and semantics check, page 1085 is performed immediately after the statement is entered.
	Note: An input assistant is also provided for entering variables, page 1090 and calling FFBs, page 1097 (functions, function blocks and procedures).
	For example:
	`WHILE X <= 100 DO`
	`  x := x + 4;`
	`END_WHILE;`

Calling a `REPEAT` statement using the input assistant

Process the following steps to call a `REPEAT` statement with the input assistant:

Step	Action
1	Place the cursor at the beginning of an empty line.

2	Select the input assistant for `REPEAT` statements, see also Selecting statements, page 1110.
	Result:
	An empty `REPEAT` statement is inserted into the section.
	`REPEAT`
	`UNTIL`
	`END_REPEAT;`

3	Complete the `REPEAT` statement with your data.
	Result: A syntax and semantics check, page 1085 is performed immediately after the statement is entered.
	Note: An input assistant is also provided for entering variables, page 1090 and calling FFBs, page 1097 (functions, function blocks and procedures).
	For example:
	`REPEAT`
	`  x := x + 2;`
	`UNITL x >= 101`
	`END_REPEAT;`

Calling a `CASE` statement using the input assistant

Process the following steps to call a `CASE` statement with the input assistant:

Step	Action
1	Place the cursor at the beginning of an empty line.

2	Select the input assistant for `CASE` statements, see also Selecting statements, page 1110.
	Result:
	An empty `CASE` statement is inserted into the section.
	`CASE OF`
	`ELSE`
	`END_CASE;`

3	Complete the `CASE` statement with your data.
	Result: A syntax and semantics check, page 1085 is performed immediately after the statement is entered.
	Note: An input assistant is also provided for entering variables, page 1090 and calling FFBs, page 1097 (functions, function blocks and procedures).
	For example:
	`CASE X OF`
	`1,5: C:=SIN(A) * COS(B);`
	`2: B:=C - A;`
	`6..10: C:=C * A;`
	`ELSE`
	`B:=C * A;`
	`C:=A / B;`
	`END_CASE;`

Online functions

Online functions

The description of online functions is found in chapter Debugging in Textual Languages (Structured Text, Instruction List), page 1421.

Reference Data Type in ST

Introduction

References to variables can be assigned in ST application programs.

A reference can be de-referenced in ST application programs.

For more information, refer to Reference Data Type (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).

ST Examples using the Reference Data Type:

myST : [MAST]

```
(*REF_ANY*) ptrToMyInt := REF(IN := myVarInt(*ANY*));

(*ANY*) ptrToMyBool := MOVE (IN := REF(myVarBool)(*ANY*));

TON_2  (IN := ptrToMyBool^(*BOOL*),
        PT := t#5s(*TIME*),
        Q => output(*BOOL*),
        ET => internalTime(*TIME*));
```

The EF, `REF`, takes a variable as an input parameter (type *ANY*) and provides its address as output parameter (type *REF_ANY*).

The EF, `MOVE`, can assign a reference to a variable to another reference. If the input parameter is (*REF(MyVariable)*), `MOVE` has the same effect as the `REF` EF.

The EF, `TON`, takes *ptrToMyBool^* as an input parameter to dereference the reference ptrToMyBool.

Export/Import IL Sections

Export/Import

The description for exporting/importing sections is found in the chapter Import / Export, page 1686.

LL984 Editor

What's in This Chapter

Creating a Program with LL984 Programming Language 1120
Editing Contacts 1139
Editing Coils 1146
Editing Function Blocks 1154
Editing Links 1158
Editing Equation Network Blocks 1163
Assigning Variables to LL984 Objects 1170
Calling Subroutines 1173
Entering Comments 1175
Customize LL984 Editor Colors 1177

Overview

This chapter describes the specifics of the LL984 Editor.

Creating a Program with LL984 Programming Language

Overview

This section describes the basics for creating a program in the LL984 programming language.

Project Settings for LL984 Programs

Introduction

Before creating an LL984 program, you should activate the following project settings:

- **Ladder Logic 984 (LL984)**

  (**Project Settings > Program > Languages**)

  To be able to create LL984 sections and networks.

- **Allow dynamic arrays (ANY_ARRAY_XXX)**

  (**Project Settings > Variables**)

  To be able to use dynamic arrays with the LL984 function blocks.

Structure of an LL984 Program

Introduction

The structure of an LL984 program corresponds to a rung for relay switching.

The left power rail is located on the left side of the LL984 Editor. This left power rail corresponds to the phase (L conductor) of a rung.

The right power rail corresponds to the neutral conductor.

A group of connected objects that are not connected to other objects (except for the power rails), is called a rung.

Properties of an LL984 program

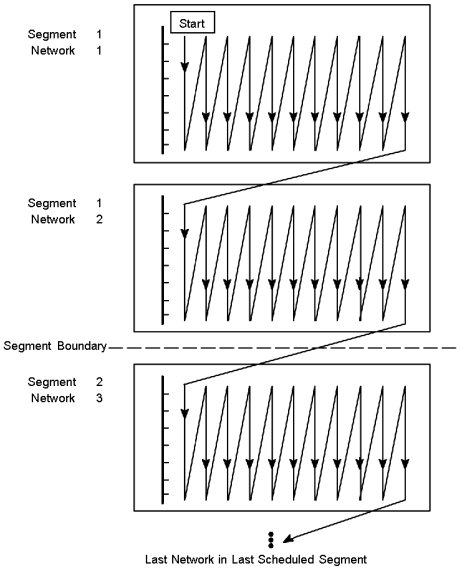
In LL984 program:

- The logic of LL984 is programmed in networks.
- Several networks are grouped in segments.
- Each network contains a matrix of 7 rows and 11 columns, where you can place contacts, coils, function blocks etc.
- LL984 programs are cell oriented, i.e. only 1 object can be placed in each cell.

  With the exception of vertical connections. They may coexist with other objects (contact, coils, function blocks).

- LL984 segments (and also LL984 networks inside segments) are executed in the sequence in which they are displayed in the **Structural View** of the **Project Browser**. To change this sequence, you can drag-and-drop LL984 segments or LL984 networks inside the **Structural View**.

  Objects in a network are executed top to bottom, left to right.



- A syntax and semantics check is performed immediately after the statement is entered. The result of this check is displayed in colored text and objects.

  Please refer to Syntax and Semantics Check During Programming, page 1126.
- Syntactically or semantically incorrect segments or networks can be saved.

LL984 Operating Modes

NOTE: to detect the first cycle in RUN, the recommended solution is to use **%S21** system bit.

LL984 Program Objects

The objects of the LL984 programming language help to divide a network into a number of:

- contacts, page 1139
- coils, page 1146
- function blocks

- equation network blocks, page 1163
- jumps and subroutine calls, page 1173

These objects can be linked with each other through:

- boolean connections and links, page 1158
- variables

Comments for the logic of the program can be added using text objects, page 1175.

Edit and View Functions

- Selecting objects
- Deleting objects
- Cutting, copying and pasting objects
- Moving objects (also between different LL984 sections)
- Replacing objects
- Undo and Redo
- Using Bookmarks, page 615
- Searching and replacing variables and function blocks, page 1239

FFB Input Features

The LL984 Editor provides 2 additional features to add function blocks to a network:

- LL984 Instruction Bar (see EcoStruxure™ Control Expert, LL984 Editor, Reference Manual)
- mnemonics (see EcoStruxure™ Control Expert, LL984 Editor, Reference Manual)

Using mnemonics, page 1157 you can also add coils and contacts to a network.

Online Functions

Displaying and modifying variable values using:

- animation tables, page 1492
- instruction editors, page 1501

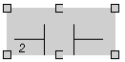
Creating an LL984 Program

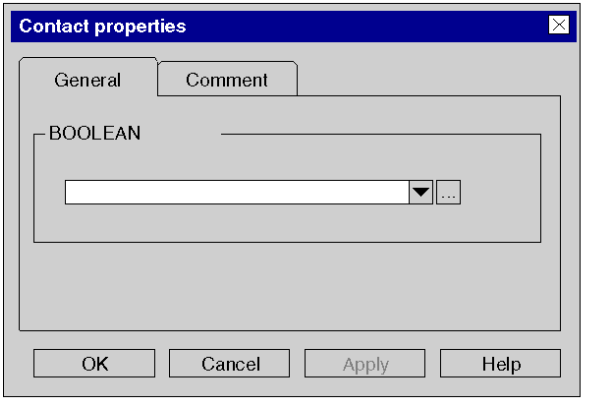
Introduction

The LL984 Editor window is made up of cells and 1 object can be placed in each one. The cells are separated visually using a grid which can be switched off (**View > Grid**).

Creating an LL984 Program with the Mouse

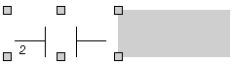
Process the following steps to create an LL984 program with the mouse:

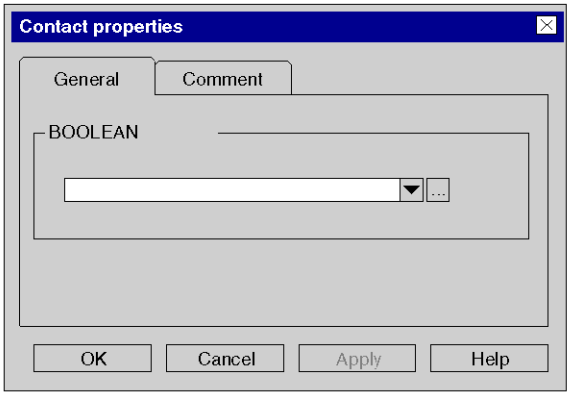
Step	Action
1	Create an LL984 network., page 539
2	Select the desired object using: • the menu commands in the **Edit > New > <Object>** menu or • the symbols for the objects in the toolbar **Result:** The mouse pointer indicates the selected object (placement mode).
3	Click the target cells in the LL984 section. **Result:** The selected object is inserted and the selection mode is active again. Example:  Return to selection mode by clicking the  symbol or pressing the **ESC** key.
4	To enter the respective actual parameters (variable/address), double-click the object. **Result:** The properties dialog box for the object is opened.

Step	Action
	Example: 
5	Enter the desired actual parameter and a comment (if required) and confirm the entries with **OK**. **Result:** The names of the associated variables are displayed above the object. **NOTE:** Entering a binary address opens the **Create variable** dialog. By this you can create a variable associated with the entered address.
6	Repeat these steps until all objects are entered.

Creating an LL984 Program with the Keyboard

Process the following steps to create an LL984 program with the keyboard:

Step	Action
1	Create an LL984 network., page 539
2	Using the keyboard, place the field with the gray background on the cell where the object should be inserted.
3	Select the desired object using the function keys for the objects. **Result:** The cursor symbol indicates the selected object.
4	Confirm the selection using the **ENTER** key. **Result:** The selected object is inserted into the cell with the gray background and the gray field is automatically shifted to the next cell. Example: 

Step	Action
	To return to selection mode, press the **ESC** key.
5	To enter the respective actual parameters (variable/address), use the **Properties** menu command from the shortcut menu. **Result:** The properties dialog box for the object is opened. Example: 
6	Enter the desired actual parameter and a comment (if required) and confirm the entries with **OK**. **Result:** The names of the associated variables are displayed above the object. **NOTE:** Entering a binary address opens the **Create variable** dialog. By this you can create a variable associated with the entered address.
7	Repeat these steps until all objects are entered.

Syntax and Semantics Check During Programming

Introduction

Syntax and semantics check is performed directly when creating the program.

The result of this check is displayed in 3 formats:

- Directly in the program section with colored objects or texts.
- As tooltip if the cursor is placed on an incorrect object.
- In the output window, if **Build > Analyze** is selected.

NOTE: Warnings are not visualized in the program section.

Colors of Objects/Texts

Color	Description
black	Syntax and semantics are correct.
blue	Possible causes: • Associated variable is not declared. • Data type of the variable does not match the data type of the pin. • Input or output pin not connected.
red wavy line below text	invalid text. For example, variables which are not declared, variables with an incorrect data type, etc.

Tooltips

If the cursor is placed over an incorrect object, a tooltip is displayed with a brief description of the cause of the error/warning.

Messages in the Output Window

Error and warning messages are displayed in the output window after **Build > Analyze**.

Double-click an error or a warning message in the output window to navigate to the affected object in the LL984 Editor.

Errors messages in contrast to warning messages must be fixed, because they inhibit a successful **Build**.

Error Messages

The following is a list of example error messages that **Analyze** will display (E).

Error Message
illegal link
variable '%1' must be located
only coils allowed in last column
only one coil per row
FFB type '%1' isn't supported
last object in row must be coil

Error Message
maximum column number allowed is '11'
maximum row number allowed is '7'
parameter '%1' expects constant value
illegal constant value '%1': %2
parameter '%1' expects variable
unsupported object
dimension of '%1' is too small, must be at least '%2'
dimension of '%1' must be '%2'
variable '%1' must be writable
parameter of '%1' must be a multiple of '%2'
parameter of '%1' must be a multiple of '%2'
usage of LL984 language is disabled
usage of SKP instruction is disabled
'%1' only allowed in subroutine network
'%1' must be in row1/column1
label '%1' isn't unique
label '%1' doesn't exist
subroutine segment doesn't exist
file '%1' is missing
IMC template '%1' doesn't exist
Non-latched usage of '%1' detected, which is used also as latched

Warning Messages

The following is a list of example warning messages that **Analyze** will display (W).

Warning Message
expression expected (if no actual parameter is assigned to a contact)
variable expected (if no actual parameter is assigned to a coil)

Tolerated Problems

Analyze will tolerate the following problems.

Problem
missing connections (to power rails or other logic elements)
skipping from a subroutine network to a network of another subroutine in the LL984 subroutine segment, if the project setting "SKP enabled" is checked

Network Display

Introduction

The LL984 Editor can display various information concerning variables attached to coils, contacts and function blocks.

This information is displayed above the respective coil, contact or function block node and can be displayed by tooltip, too.

4 Different Views

You can define 4 different views (**View 1** to **View 4**) via **Tools > Project Settings > Program > LL984 > Network Display**.

According to the settings for **View 1** to **View 4** the LL984 Editor will show the following information:

- address (e.g. %M100)
- 984-address (e.g. 400100)
- variable name
- variable comment
- descriptors 1 to 9
- topological address

For detailed information on how to define the different views, please refer to *Project Settings*, section *Program* (see EcoStruxure™ Control Expert, LL984 Editor, Reference Manual).

View Switching

You can switch between the different views defined, using **CTRL+W** or using the **Change View Mode** button ().

The button's tooltip shows the current view (**View 1, 2, 3 or 4**.

Navigating with the Keyboard

Using the Keyboard

The following keys and key combinations are provided for navigation:

Key combinations	Motion
Left arrow	Moves the gray field in a cell to the left by one cell and selects the contents of that cell. If the first line is reached, the previous line is automatically selected.
Right arrow	Moves the gray field in a cell to the right by one cell and selects the contents of that cell. If the last line is reached, the next line is automatically selected.
Up arrow	Moves the gray field in a cell to the up by one cell and selects the contents of that cell.
Down arrow	Moves the gray field in a cell to the down by one cell and selects the contents of that cell.
Shift+Left arrow	Moves the selected object and the gray field to the left by one cell. This also applies to several selected objects, the gray field must be behind one of the selected objects.
Shift+Right arrow	Moves the selected object and the gray field to the right by one cell. This also applies to several selected objects, the gray field must be behind one of the selected objects.
Shift+Up arrow	Moves the selected object and the gray field up by one cell. This also applies to several selected objects, the gray field must be behind one of the selected objects.
Shift+Down arrow	Moves the selected object and the gray field down one cell. This also applies to several selected objects, the gray field must be behind one of the selected objects.

Key combinations	Motion
Home	Positions the gray field in the first column and shows this new location.
End	Positions the gray field in the last column and shows this new location.
Ctrl+Home	Positions the gray field in the upper left hand cell in this network and shows this new location.
Ctrl+End	Positions the gray field in the upper right hand cell in this network and shows this new location.
Page Up	Scrolls the position of the gray field one page up and shows this new position, as long as the first row is not visible. Displays the previous network/segment, if the first row is already visible. You can also display the previous segment/network via **View > GoTo Previous Section**.
Page Down	Scrolls the position of the gray field one page down and shows this new position, as long as the last row is not visible. Displays the next network/section, if the last row is already visible. You can also display the next section/network via **View > GoTo Next Section**. Using **Page Down** in the last network of a segment, if the last row is already visible, will display a message box asking you to create a new network. Click **Yes** to open the **New Network** dialog. Click **No** to open the next segment/network.
Space bar	Selects or deselects the object in the cells highlighted in gray.
Shift + **Space bar**	The position of the gray field is moved one cell to the right.
Enter	In Insert mode: Inserts the selected object into the currently selected cells highlighted in gray and moves the position of the gray field one cell to the right.
Alt+Enter	Opens the properties dialog box for the selected object/pin.
Tab	Select the next pin in an FFB if the FFB or an FFB pin is selected.
Esc	Activates the Select mode.

NOTE: You can also display a segment/network by double-clicking the respective segment/network in the **Structural View** of the **Project Browser**.

Selecting Objects

Select Mode

Objects are selected in select mode.

Select mode can be activated using:

- the **Edit > Select Mode** menu command
- the  symbol,

  or
- the **Esc** key

The  cursor symbol indicates that select mode is active

Selecting an Object

Using the Mouse	Using the Keyboard
Left-click the object you want to select.	Move the gray field to the cell with the object to be selected, see also Navigating with the Keyboard, page 1130.

Selecting Several Objects

Using the Mouse	Using the Keyboard
1. Click the left mouse button and keep it pressed. 2. Drag the mouse across the objects you want to select. or 1. Left-click the first object you want to select. 2. Press the **Ctrl** key and keep it pressed. 3. Left-click the next object you want to select. 4. Repeat these steps until all desired objects are selected.	Selecting several objects: 1. Move the gray field to the cell with the first object to be selected, see also Navigating with the Keyboard, page 1130. 2. Press the **Shift+Space bar** key combination. **Result:** The object is added to the current selection and the gray field is shifted one cell to the right. 3. Repeat these steps until all desired objects are selected.

Selecting by Rows

Using the Mouse	Using the Keyboard
Selecting the contents of a row: 1. Left-click the number of the row you want to select in the vertical ruler. Selecting the contents of several lines: 1. Left-click the number of the first line you want to select in the vertical ruler. 2. Press the **Shift** key and keep it pressed. 3. Left-click the number of the last line you want to select in the vertical ruler.	-

Selecting by Columns

Using the Mouse	Using the Keyboard
Selecting the contents of a column: 1. Left-click the number of the column you want to select in the horizontal ruler. Selecting the contents of several columns: 1. Left-click the number of the first column you want to select in the horizontal ruler. 2. Press the **Shift** key and keep it pressed. 3. Left-click the number of the last column you want to select in the horizontal ruler.	-

Selecting All

Using the Mouse	Using the Keyboard
Use the **Edit** > **Select All** menu command.	Press the **Ctrl+A** key combination.

Deselecting Objects

Using the Mouse	Using the Keyboard
Left-click an empty space in the section.	Move the gray field.

Deleting, Cutting, Copying, Pasting and Moving Objects

Deleting Objects

Using the Mouse	Using the Keyboard
1. Select, page 1132 the object to be deleted. 2. Use the **Edit > Delete** menu command.	Deleting the selected object: 1. Select, page 1132 the object to be deleted. 2. Press the **Del** key. Deleting the object left of the gray field: 1. Press the **Backspace** key.
Result: The selected object is deleted. The actual parameters are deleted together with the object, even if they were not selected explicitly. Boolean links are only deleted if they are selected explicitly.	

Cutting Objects

Using the Mouse	Using the Keyboard
1. Select the object to be cut. 2. Use the **Edit > Cut** menu command. or Use **Cut** from the context menu (right-click).	1. Select the object to be cut. 2. Press the **Ctrl+X** key combination.
Result: The selected object is cut from the section and copied to the clipboard. This is also the case for the actual parameters.	

The cut object can be inserted in any other position (also in another LL984 network).

Copying Objects to the Clipboard

Using the Mouse	Using the Keyboard
1. Select the object to be copied. 2. Use the **Edit > Copy** menu command. or Use **Copy** from the context menu (right-click).	1. Select the object to be copied. 2. Press the **Ctrl+C** key combination.
Result: The selected object is copied to the clipboard. This is also the case for the actual parameters.	

The cut object can be inserted in any other position (also in another LL984 network).

Pasting Objects from the Clipboard

Using the Mouse	Using the Keyboard
1. Use the menu command **Edit > Insert**. or Use **Paste** from the context menu (right-click). 2. Left-click the target position.	1. Move the cursor to the target position pressing **Ctrl** and using the arrow keys. 2. Press the **Ctrl+V** key combination. 3. Press **Enter**.

NOTE: (for pasting contacts and coils) If the target position already contains a contact or coil, the existing object is overwritten by the new object.

NOTE: The following is the behavior of the **Copy**, **Cut** and **Paste** on a function block instance. It only applies to graphical languages as **FBD**, **LD** and **LL984**.

- Using the **Paste** function after a **Copy** of an object:

  As a result of a **Copy**, a new function block instance (**FBI name**) is used. The **Paste** function creates a new FBI whenever it is repeated. Accordingly, the FBI name is incremented.

- Using the **Paste** function after a **Cut** of an object:

  As a result of a **Cut**, the same instance of the function is used. The **Paste** function uses the same FBI whenever it is repeated. Accordingly, the FBI is identical.

NOTE: Using same instance of a function block is not recommended because it can lead to incorrect execution of the function blocks. A function block instance uses contextual variables that are owned by each instance usage and cannot be applied to any other. For instance, each current value is specific to one timer instance and one timer usage (Current value of a timer named TON_1 cannot be shared with another timer named TON_2).

Moving Objects

Using the Mouse	Using the Keyboard
1. Select the object to be moved. 2. Position the mouse pointer on the selected object. (On one of the selected objects if several are selected). **Result:** The mouse pointer changes its symbol to  . 3. Click the left mouse button and keep it pressed. **Result:** The mouse pointer changes its symbol to  . 4. Drag the object to the new position and release the mouse button.	1. Position the gray field on the object to be moved. 2. Press the **Space bar**. 3. Press the **Shift** key and keep it pressed. 4. Move the object to the target position using the arrow keys.
Result: The selected object is moved from its original position to the target position. The object's actual parameters are moved together with the object. Boolean links are only kept for horizontal movement operations. **NOTE:** A moving of FFBs to cells that are already occupied (e.g. by contact, coil, other FFB, boolean link), is not possible. **NOTE:** The move operation is also possible across sections between different opened LL984 networks. **NOTE:** When objects are moved beyond the visible area of the editor window, automatic scrolling of the window will not begin until you reach the edge of the window with the mouse pointer, not as soon as the objects are moved beyond the edge.	

Copying Objects via Drag-and-Drop

Using the Mouse	Using the Keyboard
1. Select the object to be copied. 2. Position the mouse pointer on the selected object. **Result:** The mouse pointer changes its symbol to  . 3. Click the left mouse button and keep it pressed. 4. Press the **Ctrl** key and keep it pressed. **Result:** The mouse pointer changes its symbol to  . 5. Drag the object to the new position. **NOTE:** This is also possible across sections between different opened LL984 networks. 6. Release the mouse button. **Result:** A copy of the selected object is pasted at the target position. The object's actual parameters (variable/address) are copied together with the object. A new instance is automatically created for FFBs.	-

Displaying the Properties

Displaying the Object Properties

You have the following options to display the object properties dialog:

- Double-click the object.
- Select, page 1132 the object and use the **Edit > Properties...** dialog.
- Select the object and use **Properties** from the context menu (right-click).
- Select the object and then press the **Alt+Enter** key combination.

When the properties dialog is called without an object being selected, the Network Properties, page 544 dialog is opened.

If several objects are selected, the commands do not work.

Displaying the Data Properties

You have the following options to display the **Data Properties** dialog, page 1202:

- Via the LL984 network:

  Select an object and use **Data Properties** from the context menu or press the **Ctrl +Enter** key combination.

- Via the Data Editor

  Select the line of the respective object in the Data Editor and use **Data Properties** from the context menu or press the **Ctrl+Enter** key combination or use the **Edit > Data Properties** dialog.

Inserting and Deleting Rows/Columns

Introduction

Each network contains a matrix of 7 rows and 11 columns.

But via **Insert/Delete row** and **Insert/Delete column** you can shift rows downwards/ upwards and shift columns to the right/left.

Rules for Inserting Rows/Columns

The following rules apply when inserting rows/columns.

- The new row is inserted above the first selected row.

  The content of the following rows will be shifted to the bottom.
- The new column is inserted left of the first selected column.

  The content of the following columns will be shifted to the right.
- If there are no free rows/columns at the bottom/right of the network, the menu items will be disabled.
- If the new row would intersect a middle or a bottom node of an FFB, the menu items will be disabled.

Rules for Deleting Rows/Columns

The following rules apply when deleting rows/columns.

- You can not delete a row/column containing any object.
- If there are no free rows/columns at the current position, the menu items will be disabled.

Inserting Rows

Step	Action
1	Select the desired row in the vertical ruler (click the row number).
2	Use the **Insert row** menu command from the context menu of the vertical ruler or use the **Edit > Insert row** menu command. **Result:** A row is inserted above the selected row.

Inserting Columns

Step	Action
1	Select the desired column in the horizontal ruler (click the column number).
2	Use the **Insert column** menu command from the context menu of the horizontal ruler or use the **Edit > Insert column** menu command. **Result:** A column is inserted left of the selected column.

Deleting Rows

Step	Action
1	Select the desired row in the vertical ruler (click the row number).
2	Use the **Delete row** menu command from the context menu of the vertical ruler or use the **Edit > Delete row** menu command. **Result:** The row is deleted. The rows below are shifted on row up. At the bottom an empty row is inserted.

Deleting Columns

Step	Action
1	Select the desired column in the horizontal ruler (click the column number).
2	Use the **Delete column** menu command from the context menu of the horizontal ruler or use the **Edit > Delete column** menu command. **Result:** The column is deleted. The columns from the right are shifted one column to the left. At the right side of the grid an empty column is inserted.

Editing Contacts

Overview

This section describes how to edit contacts in the LL984 programming language.

Selecting Contacts

Introduction

A contact is an LL984 element that transfers a status on the horizontal link to its right side. This status comes from the Boolean AND link of the status of the horizontal link on the left side with the status of the relevant Boolean current parameter.

Information Concerning Variables

The LL984 Editor can display various information concerning variables attached to coils, contacts and function block nodes.

This information can be displayed by tooltip, too.

For further information on how the variables attached to contacts are displayed, please refer to *Network Display*, page 1129.

Contact Types

Contact Type	Description	Call via Menu Command	Call via Icon	Call via Key
Normally open contact	For normally open contacts, the status of the left link is copied to the right link, if the status of the relevant Boolean actual parameter is 1. Otherwise, the status of the right link is 0.	**Edit > New > Normally open contact** or **Normally open contact** from the context menu		**F3**
Normally closed contact	For normally closed contacts, the status of the left link is copied to the right link, if the status of the relevant Boolean actual parameter is 0. Otherwise, the status of the right link is 0.	**Edit > New > Normally closed contact** or **Normally closed contact** from the context menu		**Shift+F3**
Positive transition-sensing contact	With contacts for detection of positive transitions, the right link for a program cycle is 1 if a transfer of the relevant actual parameter goes from 0 to 1 and the status of the left link is 1 at the same time. Otherwise, the status of the right link is 0. Also see Edge Recognition (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).	**Edit > New > Positive transition-sensing contact** or **Positive transition-sensing contact** from the context menu		**Ctrl+F3**
Negative transition-sensing contact	With contacts for detection of negative transitions, the right link for a program cycle is 1 if a transfer of the relevant actual parameter goes from 1 to 0 and the status of the left link is 1 at the same time. Otherwise, the status of the right link is 0. Also see Edge Recognition (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).	**Edit > New > Negative transition-sensing contact** or **Negative transition-sensing contact** from the context menu		**Ctrl + Shift + F3**

Placing Contacts

Introduction

Contacts can be placed in any free cell except cells directly on the right power rail.

If a contact is placed in a cell that was previously occupied by a contact, coil, a Boolean link or a label, the cell contents are replaced by the new contact.

If a contact is placed in a cell that is already occupied by another object (e.g. FFBs), an error message is returned.

The following are permitted as actual parameters for contacts:

- Boolean variables
- Boolean literals (`0`, `1`, `FALSE`, `TRUE`)
- Boolean addresses (topological addresses or symbolic addresses)
- ST expressions that return a Boolean result (e.g. `VarA > VarB`)

  ST expressions as formal parameters on contacts are an extension of IEC 61131-3 and must be explicitly enabled via the **Tools > Project Settings > Program > Languages > Common** by activating the **Usage of ST expressions** check box.

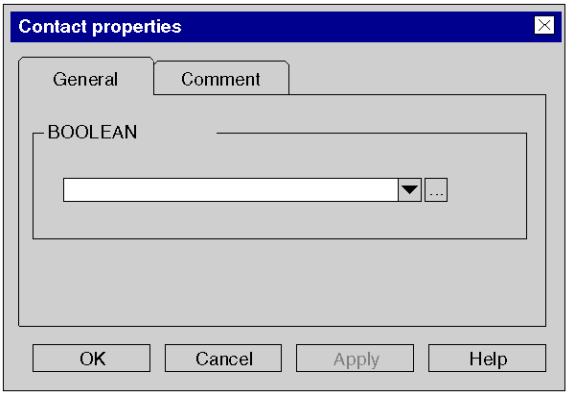
Automatically Assign a Variable

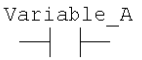
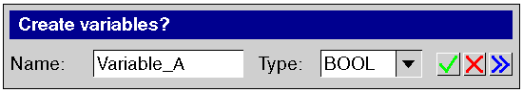
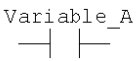
If via **Tools > Options > Data and Languages > Languages** the **Automatically assign a variable to a new graphical object** check box is activated, then the corresponding properties dialog is automatically opened when an object is placed. The procedures explained here are used when the check box is **deactivated**.

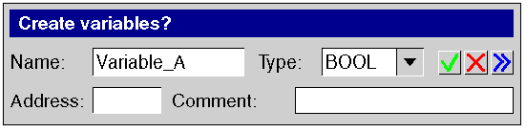
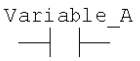
Placing Contacts

Step	Action
1	Select the desired contact, see also *Selecting Contacts*, page 1139.
2	Click the target cell in the LL984 section. or Use the **Arrow** keys to move the gray field to the target position, and press **Enter**. **Result:** The selected contact is inserted.
3	To place further contacts of the same type: • Click the target cell in the LL984 section. or Use the **Arrow** keys to move the gray field to the target position and press **Enter**. • To return to selection mode: Press the **Esc** key. • To insert other objects: Select the object you wish to insert.

Assigning Actual Parameters

Step	Action
1	Open the properties dialog, page 1137 for the contact. **Result**: The contacts properties dialog is opened. Example: 
2	You have the following options to enter the actual parameters:

Step	Action
	• You can enter %Mx (%Ix) and confirm using **Enter**. • You can enter the actual parameter and confirm using **Enter**. • You can select the actual parameter from the list of recently used names using the ▾ symbol and confirm using **Enter**. or • Using the button **...** you can open a **Data Selection** dialog and confirm the selected variables there with **OK**.
3	Confirm the selected variable with **OK**. **Result:** • If the selected variable is declared, it is enabled in the section. A syntax and semantics check, page 1126 is performed and the procedure ends here.  • If the variable is not declared, a **Create variable?** dialog opens. In this case, continue with the following steps of this procedure.  **NOTE:** If you wish to declare the variable later, close the dialog using the ☒ symbol. In this case, the variable name is enabled in the section, but not declared. **NOTE:** Entering a binary address (%M1 / %I1) opens the **Create variable** dialog proposing a variable name (_000001 / _100001) that can be modified. If there is already a variable associated to %M1 / %I1 it will be reused as proposal.
4	If you do not want to assign the variable an address or comment, confirm using **Enter** or the ✓ symbol. **Result:** The variable is declared and enabled in the section. A syntax and semantics check is performed and the procedure ends here. Example: 

Step	Action
5	If you do want to assign the variable an address and/or comment, use the symbol for obtaining an advanced dialog and enter the address and/or comment. 
6	Confirm with **Enter** or the symbol. **Result:** The variable is declared and enabled in the section. A syntax and semantics check is performed immediately after the statement is entered. Example: 

Usage of ST Expressions

To enter ST expressions for contacts the same rules apply as for entering ST expressions for FFB pins, page 826.

Properties Dialog for Contacts

Calling the Properties Dialog

See Displaying the Properties, page 1137.

Structure of the Properties Dialog

The contact properties dialog consists of 2 tabs:

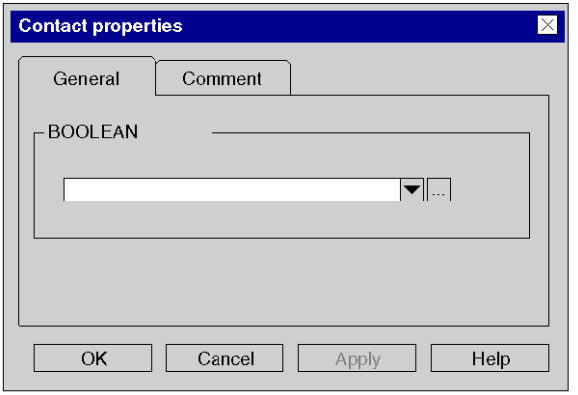
- **General**

  In this tab, you can enter the actual parameter, page 1142 of the contact.
- **Comment**

  In this tab, you can enter a comment about the contact.

General Tab

Representation of the **General** tab



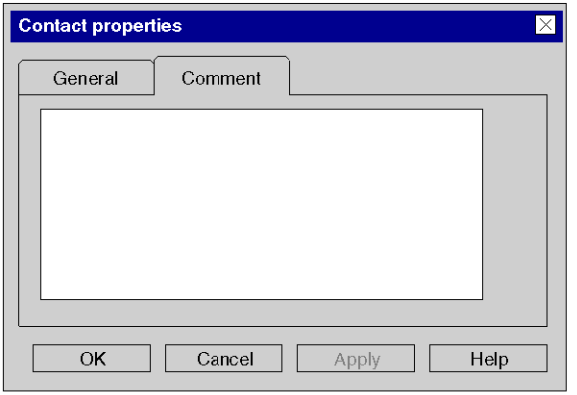
Elements of the **General** tab

Element	Description
BOOLEAN Expression	Enter the name of the actual parameter in this text box. You have the following options: • You can directly enter the name of the variable/address or paste it from the clipboard. • You can select the name of the variable from the list of recently used names using the ▼ symbol. or • Use the command button **...**to open a **Data Selection** dialog, page 1181. The actual parameters which are permitted are: • Boolean variables • Boolean constants • Boolean addresses (topological addresses or symbolic addresses) • ST expression delivering a Boolean result (e.g. `BoolVar1 OR BoolVar1`) • Literal (`1` or `0` or. `TRUE` or `FALSE`)
OK	Use this command button to accept the entries and close the dialog.
Apply	Use this command button to accept the entries without closing the properties dialog.

NOTE: Entering a binary address (%M1 / %I1) opens the **Create variable?** dialog proposing a variable name (_000001 / _100001) that can be modified. If there is already a variable associated to %M1 / %I1 it will be reused as proposed.

Comment Tab

Representation of the **Comment** tab



Elements of the **Comment** tab

Element	Description
Text box	Enter a comment about the contact. This comment is displayed as a tooltip when the cursor is placed over the contact. It is independent from the variable comment that can be assigned to the variable in the Data Editor.
OK	Use this command button to accept the entries and close the dialog.
Apply	Use this command button to accept the entries without closing the properties dialog.

Editing Coils

Overview

This section describes how to edit coils in the LL984 programming language.

Selecting Coils

Introduction

A coil is an LL984 element which transfers the status of the horizontal link on the left side, unchanged, to the horizontal link on the right side. The status is saved in the respective Boolean actual parameter. Coils normally follow contacts or FFBs, (functions and function blocks), but they can also be followed by contacts.

Information Concerning Variables

The LL984 Editor can display various information concerning variables attached to coils, contacts and function block nodes.

This information can be displayed by tooltip, too.

For further information on how the variables attached to coils are displayed, please refer to *Network Display*, page 1129.

Coil Types

Non-latched/ and latched coils have different behavior.

Non-latched Coils:	A reset of the coil will be performed during warm-start and after STOP/RUN of the controller. Reset during warm-start works for %M addresses and %M-located-variables attached only.
Latched Coils:	In the first scan latched coils have the status of last scan before warm-start.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

- Before using non-latched or latched coils, make sure that you completely understand the different behavior of non-latched and latched coils.
- Be sure of the output states after a controller warm start when using latched and non-latched coils.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Coil Type	Description	Call via Menu Command	Call via Icon	Call via Keyboard
Coil (non-latched)	With coils, the status of the left link is copied to the relevant Boolean actual parameter and the right link. For further information see below.	**Edit > New > Coil** or **Coil** from the context menu		**F5**
Latched coil	With latched coils, the status of the left link is copied to the relevant Boolean actual parameter and the right link. For further information see below.	**Edit > New > Latched coil** or **Latched coil** from the context menu		**Shift+F5**

Controller Start and Operation of Coils

- **Non-latched Coils**

  A reset of the coil will be performed during warm-start and after STOP/RUN of the controller. Reset during warm-start works for %M addresses and %M-located-variables attached only.

- **Latched Coils**

  Latched coils will maintain their state after a Stop/Start of the controller and on warm-start. In the first scan latched coils have the status of last scan before warm-start.

  This is the same behavior as normal coils show in IEC Ladder Diagram language.

Detailed information:

- For details when a warm-start is performed please see *Processing of Power Outage and Restoral for Premium/Quantum PLCs* (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).

  Especially for the case of a Power Cycle this depends on the position of the controller key switch (**MemPrt**/**Start**/**Stop**).

  - For firmware version 2.8:

    Latched coils are **not** maintained if the key switch is in the Start position.

  - For firmware version 2.6:

    Latched coils **are** maintained regardless of the key switch position.

- For details regarding the controller key switch refer to *Key Switches* (see Quantum using EcoStruxure™ Control Expert, Hardware, Reference Manual).

  For details regarding Cold Start and warm-start processing refer to *Processing on Cold Start for Premium/Quantum PLCs* (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) and *Processing on Warm Restart for Premium/Quantum PLCs* (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).

Non-latched Coil Restriction

The following restriction applies for non-latched coils:

- If an address (or a variable located on this address) is attached to a normal LL984 coil (non-latched) and this address has other (latched) write usages somewhere else in the program (for example an assignment in ST/FBD/LD or a latched coil in an LL984 network), this will lead to the following analyze message:

  `Non-latched usage of '%1' detected, which is used also as latched.`

  This message will be reported for the non-latched coil(s) usage in LL984 networks only.

  To help avoid this restriction, start a search for getting the occurrences of this conflicting address.

Latched Coil Representation

A latched coil is represented by a circle with an L or an M in the middle.

Whether an **L** or an **M** will be displayed inside a latched coil can be define via **Tools > Options > Data and Languages > LL984 > Show latched coil as**. Please refer to *Data and Languages* (see EcoStruxure™ Control Expert, LL984 Editor, Reference Manual).

Placing Coils

Introduction

For placing coils the following restrictions apply:

- Only 1 coil can be placed per row.
- No other object can be placed to the right of a coil. The coil is always the last object in a row.
- If a coil is placed in a cell that is already occupied by another object (e.g. FFBs), an error message is returned.

A placed coil automatically creates a connection with its neighboring objects on the left if they are of the `BOOL` data type, even if free cells are between them.

The following are permitted as actual parameters for coils:

- Boolean variables
- Boolean addresses (topological addresses or symbolic addresses)

Show Coils in Last Column

Using the main menu **View** you can activate the **Show coil in last column** setting.

If this setting is activated, each coil will be displayed in the last column with a dotted line between its "real" location cell and the last column.

Otherwise each coil will be displayed at its "real" location cell.

NOTE: Activating this setting will switch the LL984 Editor to **Read only** mode. No modifications are allowed inside the LL984 Editor. To leave the **Read only** mode, just deactivate this setting.

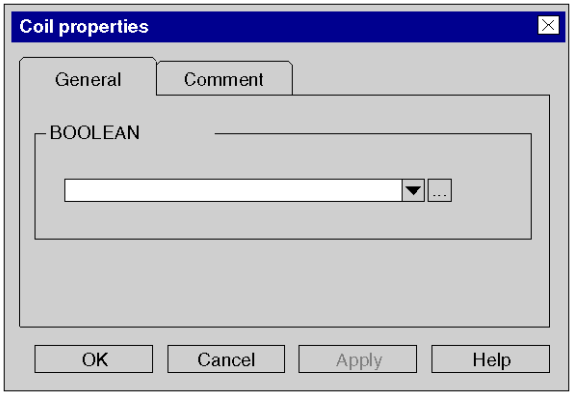
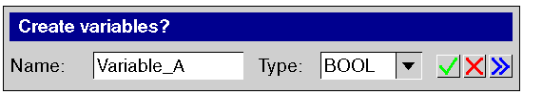
Automatically Assign a Variable

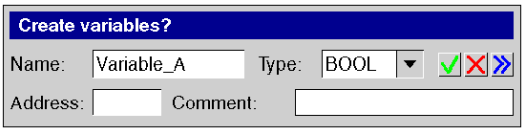
If via **Tools > Options > Data and Languages > Languages** the **Automatically assign a variable to a new graphical object** check box is activated, then the corresponding properties dialog is automatically opened when an object is placed. The procedures explained here are used when the check box is **deactivated**.

Placing Coils

Step	Action
1	Select the desired coil. See also Selecting Coils, page 1147.
2	Click the target cell in the LL984 section. or Use the **Arrow** keys to move the gray field to the target position, and press **Enter**. **Result:** The selected coil is inserted.
3	To place further coils of the same type: • Click the target cell in the LL984 section. or Use the **Arrow** keys to move the gray field to the target position and press **Enter**. • To return to selection mode: Press the **Esc** key. • To insert other objects: Select the object you wish to insert.

Assigning Actual Parameters

Step	Action
1	Open the properties dialog, page 1137 for the coil. **Result**: The coils properties dialog is opened. Example: 
2	You have the following options to enter the actual parameters: • You can enter %Mx (%Ix) and confirm using **Enter**. • You can enter the variable/address name and confirm using **Enter**. • You can select the variable name from the list of recently used names using the ▾ symbol and confirm using **Enter**. or • Using the button **...** you can open a **Data Selection** dialog and confirm the selected variables there with **OK**.
3	Confirm the selected variable with **OK**. **Result:** • If the selected variable is declared, it is enabled in the section. A syntax and semantics check, page 1126 is performed and the procedure ends here. • If the variable is not declared, a **Create variable?** dialog opens. In this case, continue with the following steps of this procedure.  **NOTE:** If you wish to declare the variable later, close the dialog using the ☒ symbol. In this case, the variable name is enabled in the section, but not declared. **NOTE:** Entering a binary address (%M1 / %I1) opens the **Create variable?** dialog proposing a variable name (_000001 / _100001) that can be modified. If there is already a variable associated to %M1 / %I1 it will be reused as proposal.

Step	Action
4	If you do not want to assign the variable an address or comment, confirm using **Enter** or the symbol. **Result:** The variable is declared and enabled in the section. A syntax and semantics check is performed and the procedure ends here.
5	If you do want to assign the variable an address and/or comment, use the symbol for obtaining an advanced dialog and enter the address and/or comment. 
6	Confirm with **Enter** or the symbol. **Result:** The variable is declared and enabled in the section. A syntax and semantics check is performed.

Property Dialog for Coils

Calling the Properties Dialog

See Displaying the Properties, page 1137.

Structure of the Properties Dialog

The coil properties dialog consists of 2 tabs:

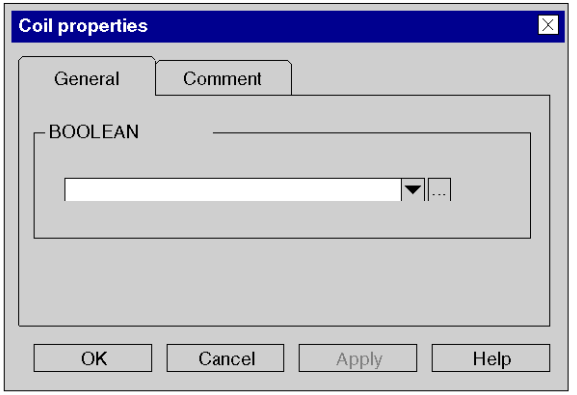
- **General**

  In this tab, you can enter the actual parameter, page 1151 of the coil.
- **Comment**

  In this tab, you can enter a comment about the coil.

General Tab

Representation of the **General** tab

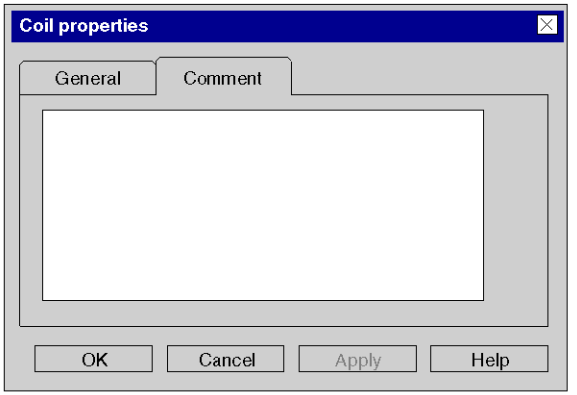


Elements of the **General** tab

Element	Description
BOOLEAN Expression	Enter the name of the actual parameter in this text box. You have the following options: • You can directly enter the name of the variable/address or paste it from the clipboard. • You can select the name of the variable from the list of recently used names using the ▼ symbol. or • Use the command button **...** to open a **Data Selection** dialog, page 1181. The actual parameters which are permitted are: • Boolean variables • Boolean addresses (topological addresses or symbolic addresses)
OK	Use this command button to accept the entries and close the dialog box.
Apply	Use this command button to accept the entries without closing the properties dialog box.

Comment Tab

Representation of the **Comment** tab



Elements of the **Comment** tab

Element	Description
Text box	Enter a comment about the coil. This comment is displayed as a tooltip when the cursor is placed over the coil. It is independent from the variable comment that can be assigned to the variable in the Data Editor.
OK	Use this command button to accept the entries and close the dialog box.
Apply	Use this command button to accept the entries without closing the properties dialog box.

Editing Function Blocks

Editing Function Blocks

Introduction

You can insert function blocks using one of the following features:

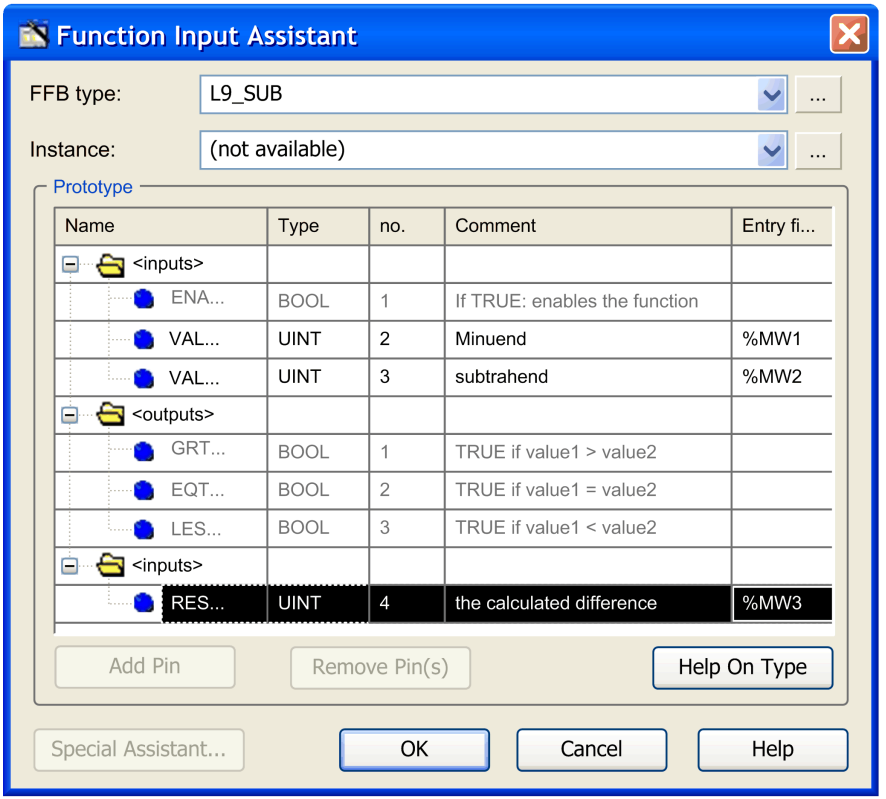
- **LL984 Instruction Bar**
- **Mnemonics**
- **Data Selection**

- **FFB Input Assistant**.
- **Types Library Browser**,
- **Copy/Paste**

Inserting a Function Block

Step	Action
1	In the LL984 Editor right-click and select from the context menu either • **Data Selection** (**Ctrl+D**) or • **FFB Input Assistant** (**Ctrl+I**). You can also insert a function block using: • drag-and-drop from the **Types Library Browser** (**Alt+3**), . • copy/paste • LL984 Instruction Bar • Mnemonics
2	Enter the complete name of the function block (e.g. L9_SUB).
3	Confirm with **Enter**.
4	Click the cell where you want to insert the function block. **Result**: The function block is inserted (if allowed at this position).

Representation of the FFB Input Assistant



You can also see the **Comments** displayed in the dialog above as a tooltip when the cursor is placed over the respective pin of the function block.

Editing Values

To open the **FFB Input Assistant** for an already inserted function block, right-click the function block and select **FFB Input Assistant** from the context menu.

Click the **Entry Field** and enter a value.

You are only allowed to edit the **Entry fields** displayed in black. Only the nodes (top, middle, bottom) of a function block can be edited.

You can not edit the **Entry fields** displayed in grey (disabled).

Restrictions

For inserting function blocks the following restrictions apply:

- Only LL984 library function blocks can be inserted in LL984 networks. If you try to insert a function block of another library a message will inform you, that this is not possible.
- You can not insert self-programmed DFBs in LL984 networks.
- LL984 library function blocks can not be inserted in IEC languages sections (FBD, LD, ST, IL). SFC does not use function blocks.
  - FBD and LD editors will refuse insertion and will display a message that insertion is not possible.
  - For textual languages (ST, IL) the call of an LL984 library function block will be detected during **Analyze**.

Using Mnemonics

Introduction

Instead of inserting LL984 objects via toolbar, **FFB Input Assistant**, **Data Selection** or drag-and-drop, you can use so called **Mnemonics**.

Mnemonics Option

Open the **Tools > Options** dialog.

Under **Data and Languages > Languages > LL984 > Mnemonics** the **Property label** column lists the contacts and coils and an assortment of frequently used functions blocks available for LL984 networks.

In the **Property value** column you can enter a mnemonic for each listed element.

A mnemonic can consist of up to 4 alphanumeric signs.

Using Mnemonics

As you start typing text, while the current cell marker is on an empty cell in the LL9894 Editor, a small edit box opens in the editor.

Entering a mnemonic (first key typed will be taken over into the edit box) and pressing the **Enter** key will insert the corresponding LL984 element at the current cell.

If the cells needed to insert the element are already occupied, a message box will inform you and the element will not be inserted.

Editing Links

Overview

This section describes how to edit links in the LL984 programming language.

Selecting Links

Introduction

Links are connections between LL984 objects (contacts, coils, FFBs etc.).

Boolean connections comprise one or more cell-segments that connect Boolean objects (contacts, coils) with one another.

- **Boolean Connection** (horizontal connection)

  This horizontal connection enables series switching of contacts and coils.

  The cell-segments of this connection can be created individually or as a complex connection comprising several cell-segments.

- **Boolean Link** (horizontal link)

  This horizontal link automatically creates a connection between neighboring Boolean objects to the left and right or between an object and the power rail.

- **Vertical Connection**

  This vertical connection enables parallel switching of contacts and coils.

Link Types

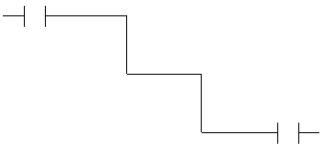
Link Type	Description	Call via Menu Command	Call via Icon	Call via Key
Horizontal connection	horizontal connection, 1 cell wide	**Edit > New > Boolean Connection** or **Boolean Connection** from the context menu		**F7**
Horizontal link	A horizontal link creates a connection between neighboring Boolean objects to the left and right. If there are no neighboring Boolean objects available, a connection to the power rail is made.	**Edit > New > Boolean Link** or **Boolean Link** from the context menu		**Alt+F6**
Vertical connection	vertical connection, 1 cell high	**Edit > New > Vertical Connection** or **Vertical Connection** from the context menu		**Shift+F7**

Combining Links

Boolean Links

Horizontal and vertical Boolean links can be combined in any way.

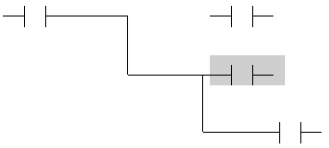
Example:



Contacts and Coils

Contacts and coils link automatically to existing horizontal or vertical links if no free cells are found between them.

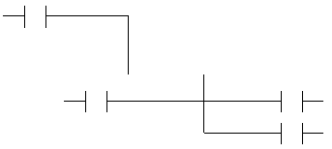
Example:



Crossing Boolean Links

If 2 Boolean links cross each other they are automatically connected. Since Boolean links may not cross each other there is no special label for them.

Example:



Placing Links

Placing Horizontal Connections

Process the following steps to place a horizontal link.

Step	Action
1	Select the horizontal connection (**Boolean Connection**). See also Selecting Links, page 1158.
2	Click the target cell in the LL984 section. or Use the **Arrow** keys to move the gray field to the target position and press **Enter**. **Result:** The link is inserted.
3	To place further links of the same type: • Click the target cell in the LL984 section. or Use the **Arrow** keys to move the gray field to the target position and press **Enter**. • To return to selection mode: Press the **Esc** key. • To insert other objects: Select the object you wish to insert.

Placing Horizontal Links

Process the following steps to place a horizontal link.

Step	Action
1	Select the horizontal link (**Boolean Link**). See also Selecting Links, page 1158.
2	Click the target cell in the LL984 section. or Use the **Arrow** keys to move the gray field to the target position and press **Enter**. **Result:** The link is inserted and a connection is created to the neighboring Boolean objects to the left and right. If there are no neighboring Boolean objects available, a connection to the power rail is made.
3	To place further links of the same type: • Click the target cell in the LL984 section. or Use the **Arrow** keys to move the gray field to the target position and press **Enter**. • To return to selection mode: Press the **Esc** key. • To insert other objects: Select the object you wish to insert.

Placing Vertical Connections

On the right side of the cells, there is space to create a vertical connection to the cell below or above. Process the following steps to place a vertical connection.

Step	Action
1	Select the vertical connection (**Vertical Connection**). See also Selecting Links, page 1158.
2	Click the object which should be connected with the cell below. or Use the **Arrow** keys to move the gray field to the target position and press **Enter**. **Result:** The link is inserted.
3	To place further links of the same type: • Click the target cell in the LL984 section. or Use the **Arrow** keys to move the gray field to the target position and press **Enter**. • To return to selection mode: Press the **Esc** key. • To insert other objects: Select the object you wish to insert.

Editing Links

Introduction

If you combine several horizontal links this will result in 1 horizontal link.

If you combine several vertical links this will result in 1 vertical link.

If you combine a horizontal and a vertical link this will result in a link made of 2 objects, a horizontal and a vertical link, that can be selected independent from each other.

Deleting, Cutting, Copying, Pasting and Moving Links

You can delete, cut, copy, paste and move links.

For further information please refer to Deleting, Cutting, Copying, Pasting and Moving Objects, page 1134.

Modifying the Size of Links

Step	Action
1	Select a link. **Result:** Size handles are displayed at the beginning and at the end of the link.
2	Click a size handle with the left mouse button and keep it pressed.
3	Drag the size handle to the desired new end position of the link and release the mouse button.

Editing Equation Network Blocks

Overview

This section describes how to edit Equation Network Blocks in the LL984 programming language.

Editing Equation Network Blocks

Introduction

Equation networks in the LL984 Editor are provided as **Equation Network Blocks**.

Equation network blocks offer an opportunity to program complex mathematical functions, with values stored in memory words.

An **Equation Network Block** contains an assignment expression written in ST language.

In general you can use the features provided by ST language as long as it is a valid ST assignment expression.

Please refer to the Structured Text (ST) chapter (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) and the ST Editor chapter, page 1082.

Using the Equation Network Block

You can insert an **Equation Network Block** using the respective toolbar button, via menu (**Edit > New > Equation Network block**) or via the editors context menu (right mouse click).

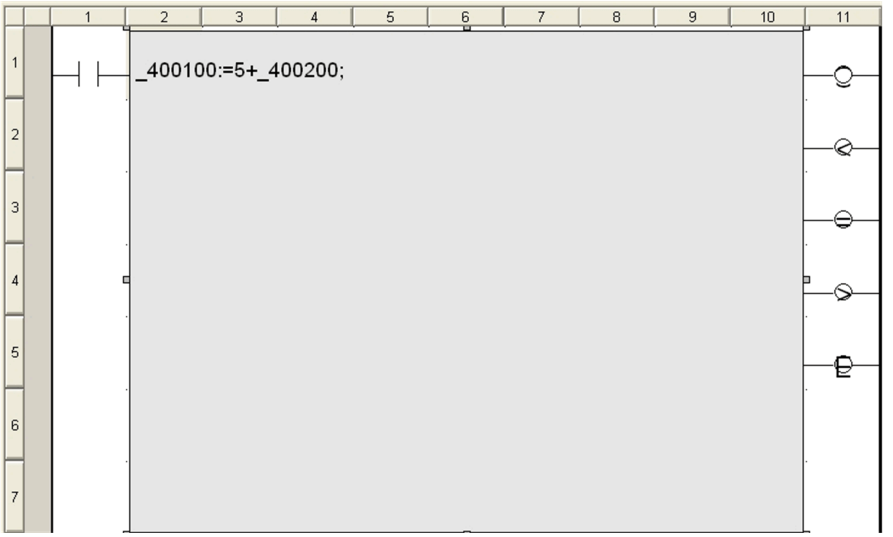
An **Equation Network Block** can only be inserted in an LL984 network at row 1, column 2. The block is 7 rows high and 9 columns wide.

On the left side of an equation network block there is an **Enable** input pin, where you can connect a contact or a boolean connection.

On the right side there are 5 output pins (**OK**, **Lower**, **Equal**, **Greater**,**Error**), where you can connect 5 normally open coils. After inserting these coils are displayed with symbols inside according to the pin names (**O**, **<**, **=**, **>**, **E**).

Representation

Representation of the **Equation Network Block**



Items of the **Equation Network Block**

Item	Description
input pin **Enable**	you can connect a contact or a boolean connection to enable the block
equation network block	expression written in ST language
output pins	**OK (O)**: set when the equation is solved without problems
	Lower (<): set when the equation result is less than zero
	Equal (=)set when the equation result is equal to zero
	Greater (>)set when the equation result is greater than zero

Item	Description
	Error (E): set when a problem occurred during solving the equation (see table below)

Problems During Solving

Problem	Description
invalid operation	An internal message generated by the math coprocessor.
overflow	A value is too large to be represented in its specified data type.
underflow	A number is too small to be represented in FP format (for floating point data only).
divided by 0	The variable, constant, or result of a function directly to the right of a / operator has a value of zero.

Editing the ST Assignment Expression

Introduction

After inserting an **Equation Network Block** double-click the displayed question marks (**?????**) to enter the ST assignment expression.

Confirming with **Enter** will start an analysis and the found detected problems will be displayed in a tooltip.

Animation of the memory words used inside the ST assignment expression can be seen in a **Variable Window**. Please refer to *Variable Display Window*, page 1467.

Memory Addresses

A memory address entered in the ST assignment expression must start with one of the following symbols, specifying the address type:

- %M (memory bit)
- %I (discrete input)
- %IW (input word)
- %MW (memory word)

The address type must be followed by an address number (no leading zeros required).

Creating Variables Automatically

After confirming the ST assignment expression with **Enter**, for every entered address a variable will be created automatically, which is used in the equation network.

Examples

Address	Variable
%M1	_000001
%I1	_100001
%IW1	_300001
%MW1	_400001

The variable type is derived from its address suffix.

Suffix		Variable Type
no suffix	%Mx, %Ix	EBOOL
	%IWx, %MWx	UINT
I		INT
DI		DINT
UD		UDINT
R		REAL
W		WORD

The variable is mapped to the associated internal located memory.

For example: %IW1DI becomes the variable `_300001_DI:DINT@%IW1`.

Constants

Constants in equation networks have the same syntax as in other IEC editors (that is without a leading # character).

Operators

Operators and their precedence

Operation	Symbol	Precedence
Parentheses	(expression)	HIGHEST
Function evaluation	identifier (argument list)	-
Negation	-	-
Complement	~	-
Exponentiation	**	-
Multiply	*	-
Divide	/	-
Add	+	-
Subtract	-	-
Comparison	< , > , <= , >=	-
Equality	=	-
Inequality	<>	-
Boolean AND	&	-
Boolean Exclusive OR	^	-
Boolean OR	|	-
Assignment	:=	LOWEST

The operators **, *, /, +, -, <<, >>, <, >, <=, >=, =, <>, &, ^ and | are supported as they are implemented now in Control Expert ST syntax. That is not all data type combinations for operand types and the result type are supported.

NOTE: Control Expert only supports a floating point value for the ** operator (EXPT-functions).

NOTE: Bitwise logical AND, OR or XOR operations work like in ST syntax.

Ternary Conditional

Ternary conditional (C?t:f), in contrast to legacy LL984, is not available in Control Expert ST syntax.

The already existing `SEL` function block provides this functionality and has to be used instead.

Example:

Legacy syntax:

```
400010 := 300010 < 300020 ? 300004 : 300005
```

Control Expert syntax:

```
%MW10 := SEL(%IW10 < %IW20, %IW4, %IW50);
```

With the following, automatically created by the editor:

```
_400010 := SEL(_300010 < _300020, _300004, _300005);
```

Bitwise Shift

Bitwise shift (<<, >>), in contrast to legacy LL984, is not available in Control Expert ST syntax.

The already existing `SHL` and `SHR` function blocks provide this functionality and have to be used instead.

Example:

Legacy syntax:

```
400010 = 300010U << 3
```

Control Expert syntax:

```
%MW100W := SHL(%IW10W, 3);
```

With the following, automatically created by the editor:

```
_400100_W := SHL(_300010_W, 3);
```

Function Blocks

In general you can use the function blocks provided for ST language.

In particular the following Control Expert function blocks replace the legacy function blocks listed below.

Legacy Function Name	Control Expert Function Name	Argument Type*	Return Type*	Meaning
ABS	ABS	S,U,L,UL,F	S,U,L,UL,F	absolute value
ARCCOS	ACOS	F	F	arc cosine
ARCSIN	ASIN	F	F	arc sine
ARCTAN	ATAN	F	F	arc tangent
COS	COS	F	F	cosine

Legacy Function Name	Control Expert Function Name	Argument Type*	Return Type*	Meaning
COSD	COSD	F	F	cosine of degrees
EXP	EXP	F	F	exponent function (power of e) (does not need to be a whole number)
FIX	REAL_TO	F	L	convert floating point to integer (presumes an FP argument)
FLOAT	INT_TO	S, U, L, UL	F	converts integer to floating point (presumes an integer argument)
LN	LN	F	F	natural logarithm (base e)
LOG	LOG	F	F	common logarithm (base 10)
SIN	SIN	F	F	sine of radians
SIND	SIND	F	F	sine of degrees
SQRT	SQRT	F	F	square root
TAN	TAN	F	F	tangent of radians
TAND	TAND	F	F	tangent of degrees
*: S=INT, U=UINT, L=DINT, UL=UDINT, F=REAL				

Example of an LL984 Expression

Legacy syntax:

```
40701 = 40702U + COS(40703UL) * #8.00135F + SIN(40704);
```

Control Expert syntax:

```
%MW701 := REAL_TO_UINT(WORD_TO_REAL(%MW702W) + COS(WORD_AS_REAL(%
MW703W, %MW704W)) * 8.00135 + SIN(WORD_TO_REAL(%MW704W)));
```

With the following, automatically created by the editor:

```
_400701 := REAL_TO_UINT(WORD_TO_REAL(_400702_W) + COS(WORD_AS_REAL
(_400703_W, _400704_W)) * 8.00135 + SIN(WORD_TO_REAL(_400704_W)));
```

Assigning Variables to LL984 Objects

Assigning Variables to LL984 Objects

Introduction

In the LL984 Editor you can assign a variable (like `_400001` or `sym1`) or an address (like %M1) to an FFB pin or to a contact/coil.

If the variable is not declared, the **Create variable?** dialog opens.

If the variable or address assigned is not compatible with the pin or contact/coil, a message will be displayed in the output window after **Build > Analyze**.

Automatic Variable Naming

If you, for example, assign an address (e.g. %MW1) to a pin, the **Create variable?** dialog opens proposing a variable name (e.g. _400001) and a data type (e.g. UINT).

The following rules apply to this automatic proposal, depending on the address of the actual parameter, the pin type and dimension (n) in case of an array type:

Address	Pin Type	Dimension (for array types)	Variable Name
%MW1	UINT	1 (n)	_400001 (_400001_n)
	INT		_400001_I (_400001_In)
	WORD		_400001_W (_400001_Wn)
%IW1	UINT	1 (n)	_300001 (_300001_n)
	INT		_300001_I (_300001_In)
	WORD		_300001_W (_300001_Wn)
%I1	EBOOL	1 (n)	_100001 (_100001_n)
%M1	EBOOL	1 (n)	_000001 (_000001_n)

NOTE: For the most used pin types UINT and EBOOL no type suffix are appended to the proposed variable name.

Use Case 1

Assignment to a pin of type UINT

User-Assigned Address	Variable	Variable Created and Assigned	Variable Type
%MW1	not existing yet	_400001	UINT

NOTE: As the pin type was UINT no type suffix was appended to the variable name.

Use Case 2

Assignment to a pin of type ARRAY[1...n] of UINT

User-Assigned Address	Variable	Variable Created and Assigned	Variable Type
%MW1	not existing yet	_400001_n	ARRAY[1...n] of UINT

Use Case 3

Assignment to a pin of type ANY or ANY_ARRAY_UINT or ANY_ARRAY_BOOL (dynamic array)

The dimension (n) of a dynamic array will be provided by another dimension pin (e.g. LENGTH) of the function block.

This dimension pin will be initialized during block insertion.

The dimension of the array will be calculated based on the dimension pin value entered as actual parameter of the dimension pin.

- If the resulting type is a UINT array:

  dimension = dimension pin value.
- If the resulting type is a Boolean array:

  dimension = dimension pin value * 16.

User-Assigned Address	Variable	Variable Created and Assigned	Variable Type
%MW1	not existing yet	_400001_n	ARRAY[1...n] of UINT

If you modify n on the dimension pin to n2:

User-Assigned Address	Variable	Variable Created and Assigned	Variable Type
%MW1	not existing yet	_400001_n2	ARRAY[1...n2] of UINT
%MW1	already existing	_400001_n2 attributes (like descriptors, comments etc.) are taken over	ARRAY[1...n2] of UINT

Use Case 4

Assignment to a pin of type DDT1

User-Assigned Address	Variable	Variable Created and Assigned	Variable Type
%MW1	not existing yet	_400001_DDT1	DDT1

Memory Consumption

If you create any kind of located variables (elementary type, array or DDT) this will not increase memory consumption on the PLC (except for Upload information).

New Variable Creation

If you assign a different address or modify a dimension pin, a new variable will be created.

The old attributes (except the address) will be overtaken and the old variable will be deleted, if it is not used anywhere else in the program.

Invalid Variables

If you create a located variable with an address range which exceeds the address range defined under **State RAM** of the CPU´s **Configuration** tab, the following message will be displayed after **Build > Analyze**:

```
Topological address index is not valid.
```

There are different ways to solve this issue:

- Click the message displayed in the output window to open the **Data Editor** and adapt the address of the variable.

- Click the message displayed in the output window to open the **Data Editor** and delete the variable and create a new variable with an applicable address range.
- Adapt the address range defined under **State RAM** of the CPU´s **Configuration** tab.

NOTE: If you just create a new variable at the pin of the function block without deleting the previously created, invalid variable the message mentioned above will be displayed again after next **Build > Analyze**.

Example:

Under **State RAM** of the CPU´s **Configuration** tab you entered a value 10000 for **%MW**.

In your program you have an `L_BLKM` function block with `LENGTH` = 10.

At the `SOURCE` pin of the function block you enter following value: `%MW9999`.

This results in creation of the following variable:

```
_409999:ARRAY[1..10] OF UINT@%MW9999
```

As the start address is `%MW9999` and the length is 10, the end address will be `%MW10009`.

This exceeds the address range (`10000`) defined under **State RAM** of the CPU´s **Configuration** tab.

Therefore the message mentioned above will be displayed after **Build > Analyze**.

Calling Subroutines

Overview

This section describes how to call subroutines in the LL984 programming language.

Calling Subroutines

Introduction

Calling subroutines is managed by 3 function blocks:

- `L9_JSR`
- `L9_LAB`
- `L9_RET`

For detailed information please refer to the function block descriptions (see EcoStruxure™ Control Expert, UnityLL984, Block Library).

Up to 1023 subroutines can be established.

Control Expert does not support interrupt handling with LL984 subroutines.

Restrictions

For the calling subroutines feature the following restrictions apply:

- Only one LL984 segment can be created inside the subroutine (**SR Section**) node of the **MAST** task.
- `L9_LAB` and `L9_RET` function blocks can only be placed in an LL984 subroutine network in the **SR Sections**.
- An `L9_LAB` function block can only be placed in row 1, column 1 of an LL984 subroutine network.

Subroutine Structure

A subroutine is identified by a number (1.. 1023).

Subroutines are called by the `L9_JSR` function block within an LL984 network (normal or subroutine network).

Two additional function blocks, which are only available in subroutine networks, mark the starting point (`L9_LAB` function block) and ending point (`L9_RET` function block) of a subroutine.

A subroutine may span multiple networks.

The `L9_RET` function block is optional and this explicit ending point of a subroutine can be omitted,

- if the last subroutine network is the last network of the subroutine segment or
- if a new subroutine network (`L9_LAB`) would entered.

The `L9_RET` function block or the implicit forms, return from the current subroutine network to the calling `L9_JSR` function block with an OK status.

Nesting

A subroutine may call other subroutines or recursively call itself up to 100 times.

If the maximum nesting level is reached, an implicit return is performed and the ERR output of the L9_JSR function block is set.

A nesting level problem message is propagated to the (99) cascaded L9_JSR function blocks.

Further cascaded L9_JSR function blocks, which are being executed after a nesting level problem, do not succeed to call a subroutine and the ERR output is being set until the nesting level stack has been unwound.

NOTE: The message is only visible at those L9_JSR function blocks, contributing to the problem. A dedicated problem treatment is possible on each calling level. Subsequent L9_JSR function blocks of a segment are not influenced by this problem.

Please also refer to the L9_JSR function block description (see EcoStruxure™ Control Expert, UnityLL984, Block Library).

Entering Comments

Overview

This section describes how to enter comments in the LL984 programming language.

Entering Comments

Introduction

Comments can be placed in LL984 sections in the form of text objects.

Text objects can overlap other objects.

The size of the object, depending on the size of the text, can be extended vertically and horizontally to fill further grid units.

The same rules apply for entering text and navigating within text objects as for editing ASCII text in standard text editors. (To create a line break in text objects you must press the **Ctrl +Enter** key combination).

NOTE: Keep in mind that each change to a comment (e.g. change to a comment text, change to the size of the text object) makes it necessary to recreate the section involved (**Build > Build Changes**).

Displaying or hiding the comments

Use the  button in the toolbar to show or hide the comments.

The background color of the button in blue indicates that the comments are hidden.

NOTE: To select or place text objects, check that the  button in the toolbar is not activated.

Selecting the Text Object

You have the following options to select text objects:

- Use the **Edit > New > Comment** menu command.
- Use the **Comment** menu from the context menu.
- Press the **F8** key.

  or

- Select the  symbol.

The active placement mode for text objects is indicated by the  cursor symbol.

Placing Text Objects

Step	Action
1	Select the text object.
2	Click the target cell in the LL984 section. or Use the **Arrow** keys to move the gray field to the target position and press **Enter**. **Result:** The text object is inserted. Example: 
3	Enter the comment.

Step	Action
4	Confirm the text entered with: • Clicking with the mouse outside of the text object or • Pressing the **Enter** key.
5	To insert further text objects: • Click the target cell in the LL984 section. or Use the **Arrow** keys to move the gray field to the target position and press **Enter**. • To return to selection mode: Press the **Esc** key. • To insert other objects: Select the object you wish to insert.

Customize LL984 Editor Colors

Overview

This section describes how to customize the LL984 editor colors.

Customize LL984 Editor Colors

Overview

To customize LL984 editor colors, open the **Tools > Colors Settings...** dialog box.

The **Color Settings** dialog box allows to:

- Modify the color of the selected item in the list.
- Reset the colors with initial values.
- Import an *.ini file that define all the colors associated to the LL984 editor.
- Export the definition colors associated to the LL984 editor.

Colors Setting File

Depending on the operating system of the PC, the colors settings file 984Colors.ini is located in one of the following folder:

- `c:\Program Files\Schneider Electric\Control Expert\`
- `c:\Program Files (x86)\Schneider Electric\Control Expert\`

For more detail on destination folder when Control Expert is installed, refer to *EcoStruxure™ Control Expert, Installation Manual*.

NOTE: Make a copy of this file before modifying it.

Description

This table describes the elements of the color file:

Parameter	Default value	Description
BkAnimation	`192,192,192`	Editor background color when animation is ON and you are connected in programming mode.
StatementError	`255,0,0`	Element in error is underlined with this color.
InspectBk	`0,255,255`	Inspect window background color
InspectText	`0,0,0`	Inspect window text color
InspectBkMin	`255,255,0`	Inspect window background color when the value is less than the minimum value defined in the inspect window settings.
InspectBkMax	`255,0,255`	Inspect window background color when the value is greater than the maximum value defined in the inspect window settings.
BkAnimMonitoring	`255,211,211`	Editor background color when animation is ON and you are connected in monitoring mode.
AnalyzeError	`0,0,255`	Drawing color when an element generates an analyze error.
GraphAnimBoolTrue	`0,150,0`	Color of `ANY_BOOL` variable when its value is TRUE.
GraphAnimBoolFalse	`255,0,0`	Color of `ANY_BOOL` variable when its value is FALSE.
BkComment	`255,255,204`	Background color of comment area
BkCommentEdit	`225,225,225`	Background color of comment area when you are editing it.
BkAnimNumeric	`255,255,0`	Background color of numeric variables
BlockBreakpoint	`128,0,0`	Bullet color in the element where the breakpoint is set.
BlockCurrentStep	`255,255,0`	Color of the current element in debug mode
AnchorLink	`128,0,0`	Color of anchor link

Parameter	Default value	Description
LLFgExpression	`0,0,0`	Text color of boolean expressions
LLFgAddress	`100,100,230`	Text color of variable address
LLFgSymbol	`0,0,0`	Text color of variable name
LLFgComment	`0,128,0`	Text color of variable comment
LLBkExpression	`255,255,255`	Background color of boolean expressions
LLBkAddress	`255,255,255`	Background color of variable address
LLBkSymbol	`255,255,255`	Background color of variable name
LLBkComment	`255,255,255`	Background color of variable comment
984CoilAlignLine	`192,192,192`	Colors of coils align line
984BkEqNet	`228,228,228`	Background color of equation network
Background	`255,255,255`	Editor background color

Programming Languages Data Selection Dialog Box

What's in This Chapter

Instance selection dialog box 1181
FFB Selection Dialog box 1186
FFB Type Selection Dialog box 1190
Data selection dialog box 1192
Function Input Assistant 1199
Data Properties..... 1202
Generate variables..... 1204

Overview

This chapter describes the dialog boxes provided for data selection that can be called from the various programming languages.

Instance selection dialog box

Structure of the dialog box

The **Instance selection** dialog box consists of two or three tabs:

- **Variables**

  You can select an elementary variable or an element of a multi-element variable in this tab.

- **Function Blocks**

  You can select a formal parameter of a function block instance in this tab.

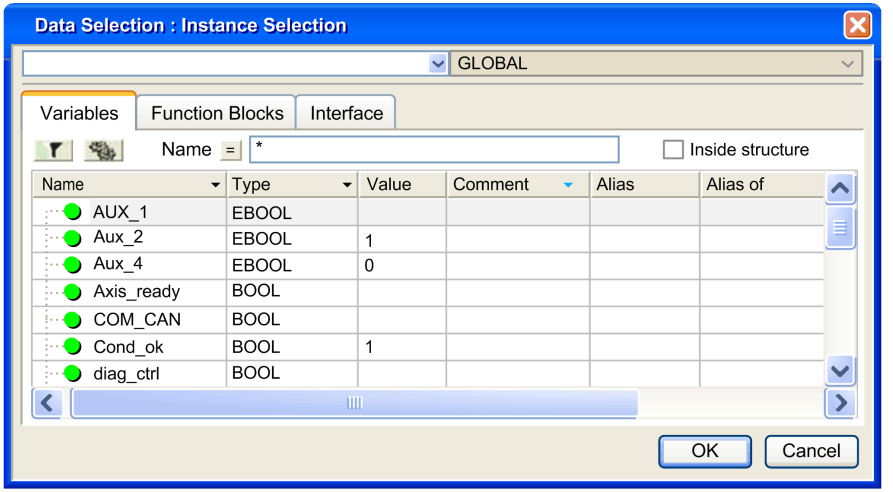
- **Interface**

  You can select a variable of a Program Unit in this tab.

  NOTE: Program Units are only available for Modicon M580 and M340. Without Program Unit in the program structure the **Interface** tab is empty.

Variables tab

Representation of the tab **Variables** tab:



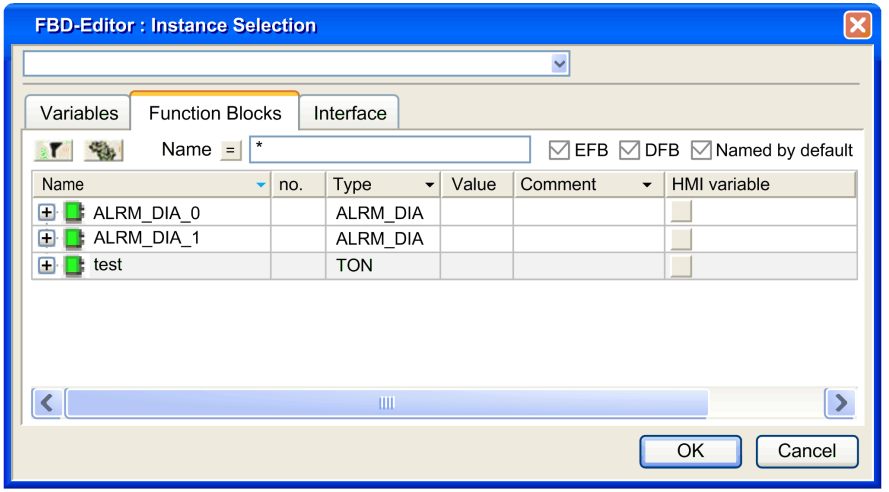
Elements of the **Variables** tab:

Element	Description
Text box	The selected variables are displayed in this text box.
Scope list	Allow to reduce the scope for the variable selection.
button 	Clicking on this button updates the display corresponding to the filter term defined in the **Name** field.
button 	Clicking on this button opens the Filtering data, page 434 dialog box where you can define the conditions for the individual attributes.
Name	The name of the variable to search for can be entered in this text box. You can also use the joker * and **?**.
button 	Clicking on this button inverts the filter. The button changes from = to <> and vice versa.
Inside Structure	When you activate this check box, all structured variables are displayed.
Name	The names of the variables are displayed in this column.

Element	Description
Type	The types of the variables are displayed in this column.
Value	The initial value of the variables are displayed in this column.
Comment	The comments for the variables are displayed in this column.
Alias	The alias name of the IODDT, DDT, or Array element are displayed in this column.
Alias of	The complete paths of the variables when an alias name has been set are displayed in this column.
Address	The address in the controller are displayed in this column.
HMI Variable	This column displays if the variable is embedded in the data dictionary.
R/W Rights of Referenced Variable	This column displays the access rights when the variable is a Reference data type.
Effective Parameter	Only when the **Instance selection** dialog box is called from a section included in a Program Unit. This column displays the variable assignment for Program Unit parameters (input, output and input/output parameters).
Nature	Only when the **Instance selection** dialog box is called from a section included in a Program Unit. This column displays the nature of the Program Unit variables.

Function Blocks tab

Representation of the **Function Blocks** tab:



Elements of the **Function Blocks** tab:

Element	Description
Text box	The selected function block instances are displayed in this text box.
button 	Clicking on this button updates the display corresponding to the filter term defined in the **Name** field.
button 	Clicking on this button opens the Filtering data, page 434 dialog box where you can define the conditions for the individual attributes.
Name	The name of the function block instance to search for can be entered in this text box. You can also use the joker * and **?**.
button 	Clicking on this button inverts the filter. The button changes from = to <> and vice versa.
EFB	When you activate this checkbox, the instances of elementary function blocks (EFBs) are displayed.
DFB	When you activate this checkbox, the instances of derived function blocks (DFBs) are displayed.

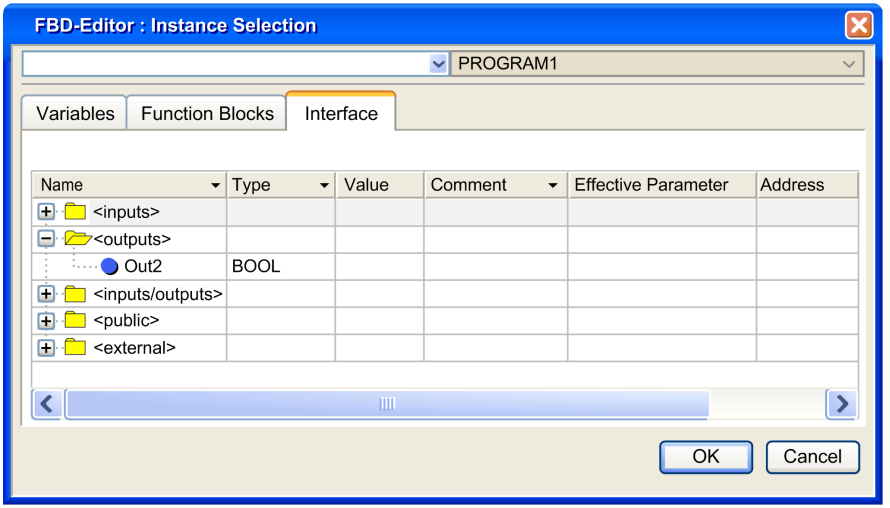
Element	Description
Named by default	When you activate this check box, the function block instances are displayed whose default names (e.g. `FBI_19_3`, `TON_1`) have not been changed.
Name	This column displays the names of the function block instances and their formal parameters (inputs, outputs, public variables).
no.	This column displays the order number of the formal parameters in the function block.
Type	This column displays the types of the function block instances and the data types of their formal parameters (inputs, outputs, public variables).
Value	This column displays the initial value of the formal parameters.
Comment	This column displays comments about the function block instances and their formal parameters (inputs, outputs, public variables).
HMI Variable	This column displays if the formal parameter is embedded in the data dictionary.
R/W Rights of Referenced Variable	This column displays the access rights when the formal parameter is a Reference data type.

NOTE: Help for the Type..

Hold down the **Alt+F1** keys and then click on the function block type for which you wish to access Help.

Interface Tab

Representation of the **Interface** tab:



Elements of the **Interface** tab:

Element	Description
Text box	The selected function block instances are displayed in this text box.
Scope list	Allow to select the data scope for the variable selection among the different Program Units.
Name	This column displays the names of the variables belonging to the Program Unit selected into the scope list. The variables are grouped by **Nature** (**<inputs>**, **<outputs>**, **<inputs/outputs>**, **<public>**, **<external>**).
Type	This column displays the data types of the variables.
Value	This column displays the initial value of the variables.
Comment	This column displays the comment for the variables.
Effective Parameter	Only when the **Instance selection** dialog box is called from a section included in a Program Unit. This column displays the variable assignment for Program Unit parameters (input, output and input/output parameters).
Address	Only when the **Instance selection** dialog box is called from a section included in a Program Unit. The address in the controller are displayed in this column.

FFB Selection Dialog box

Structure of the properties dialog box

The dialog box for selecting function block instances and function types/function block types consists of two tabs in FBD and LD:

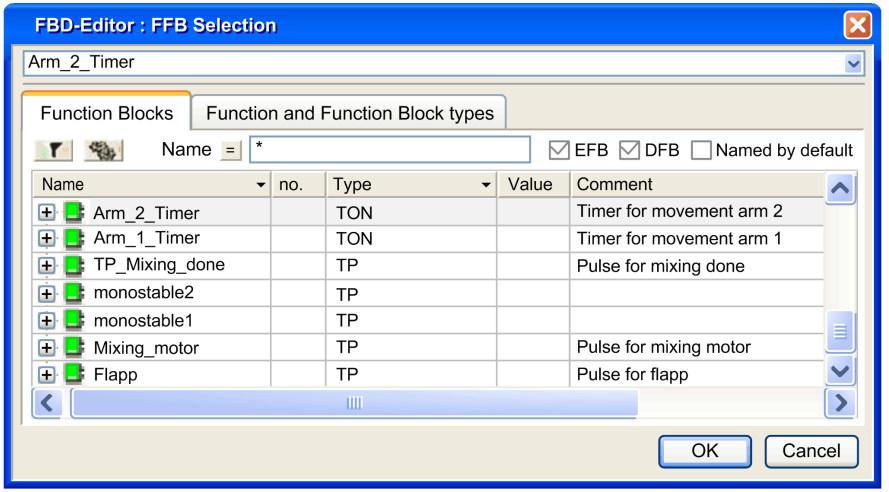
- **Function Blocks**

  You can select a function block instance in this tab.
- **Function and Function Block Types**

  You can select a function type or function block type in this tab.

Function Blocks tab

Representation of the **Function Blocks** tab:



Elements of the**Function Blocks** tab:

Element	Description
Text box	The selected function block instances are displayed in this text box.
button 	Clicking on this button updates the display corresponding to the filter term defined in the **Name** field.
button 	Clicking on this button opens the Filtering data, page 434 dialog box where you can define the conditions for the individual attributes.
Name	The name of the function block instance to search for can be entered in this text box. You can also use the joker ***** and **?**.
button 	Clicking on this button inverts the filter. The button changes from = to <> and vice versa.
EFB	When you activate this check box, the instances of elementary function blocks (EFBs) are displayed.
DFB	When you activate this check box, the instances of derived function blocks (DFBs) are displayed.

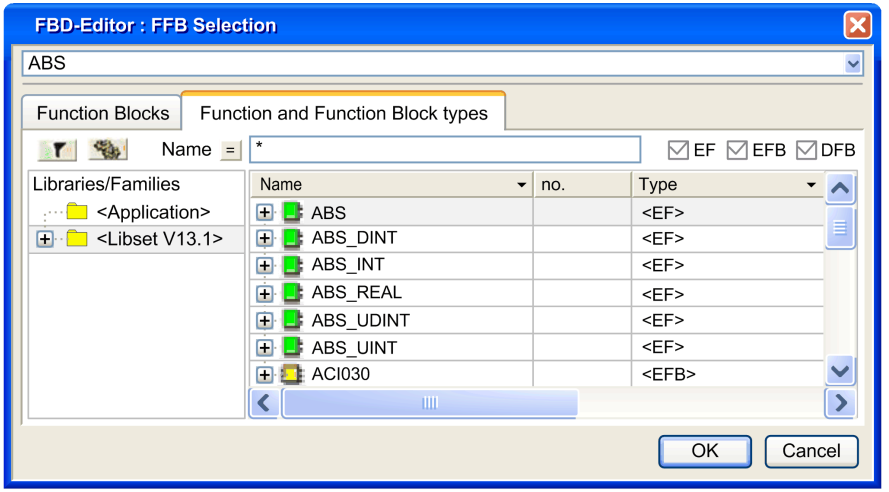
Element	Description
Named by default	When you activate this check box, the function block instances are displayed whose default names (e.g. FBI_19_3, TON_1) have not been changed.
Name	This column displays the names of the function block instances and their formal parameters (inputs, outputs).
no.	This column displays the order number of the formal parameters in the function block.
Type	This column displays the types of the function block instances and the data types of their formal parameters (inputs, outputs).
Value	This column displays the initial value of the formal parameters.
Comment	This column displays comments about the function block instances and their formal parameters (inputs, outputs).
HMI Variable	This column displays if the formal parameter is embedded in the data dictionary.
R/W Rights of Referenced Variable	This column displays the access rights when the formal parameter is a Reference data type.

NOTE: Help for the Type.

Hold down the **Alt+F1** keys and then click on the function block type for which you wish to access Help.

Function and Function Block Types tab

Representation of the **Function and Function Block Types** tab:



Elements of the **Function and Function Block Types** tab:

Element	Description
Text box	The selected function block instances are displayed in this text box.
Help On Type	Use this button to open the help for the function block types shown. **Note:** Alternatively, you can also hold down **Alt+F1** and then click on the function block type for which you want to access Help.
button 	Clicking on this button updates the display corresponding to the filter term defined in the **Name** field.
button 	Clicking on this button opens the Filtering data, page 434 dialog box where you can define the conditions for the individual attributes.
Name	The name of the function type or function block type to search for can be entered in this text box. You can also use the joker * and **?**.
button 	Clicking on this button inverts the filter. The button changes from = to <> and vice versa.
EF	When you enable this check box, Elementary Function types (EFs) are displayed.
EFB	When you enable this check box, Elementary Function Block types (EFBs) are displayed.
DFB	When you activate this check box, Derived Function Blocks (DFBs) are displayed.
Libraries/Families	The scope of the displayed function types and function block types is selected in this column: • To display all function types and function block types used in the project, select **<Application>**. • To display all function types and function block types of a library independently of the library assignment, select **<Libset>**. • To display all function types and function block types of a library, select the desired library. • To display all function types and function block types of a family, select the desired family.
Name	This column displays the names of the function types and function block types and their formal parameters (inputs, outputs).
no.	This column displays the order number of the formal parameters.

Element	Description
Type	This column displays the types of the function types and function block types and the data types of their formal parameters (inputs, outputs).
Value	This column displays the initial value of the formal parameters.
Comment	This column displays comments about the function types and function block types and their formal parameters (inputs, outputs).
R/W Rights of Referenced Variable	This column displays the access rights when the formal parameter is a Reference data type.

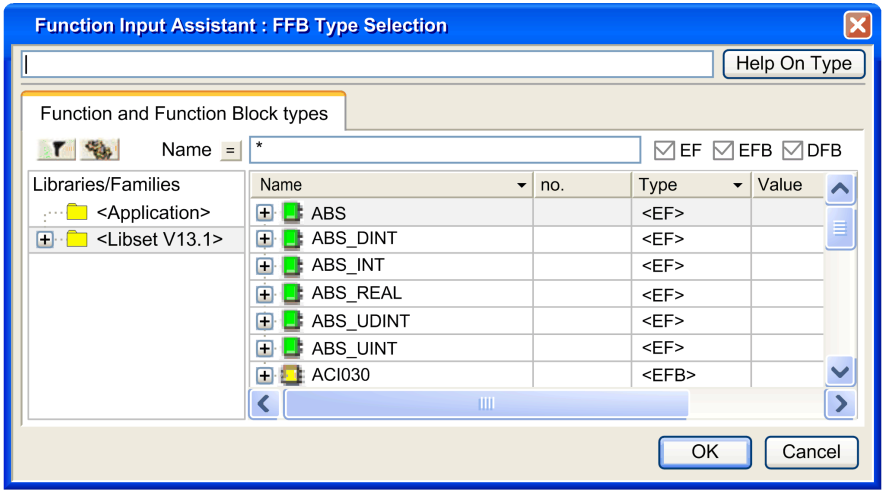
FFB Type Selection Dialog box

Structure of the properties dialog box

In the **Function Input Assistant dialog box: FFB type selection** you can select a function type or function block type in this tab.

Function Input Assistant: FFB Type Selection

Representation of the dialog box:



Elements of the **Function and Function Block Types** dialog box:

Element	Description
Text box	The selected function or function block types are displayed in this text box.
Help On Type	Use this button to open the help for the function block types shown. **Note:** Alternatively, you can also hold down **Alt+F1** and then click on the function block type for which you want to access Help.
button 	Clicking on this button updates the display corresponding to the filter term defined in the **Name** field.
button 	Clicking on this button opens the Filtering data, page 434 dialog box where you can define the conditions for the individual attributes.
Name	The name of the function type or function block type to search for can be entered in this text box. You can also use the joker ***** and **?**.
button 	Clicking on this button inverts the filter. The button changes from = to <> and vice versa.
EF	When you enable this check box, Elementary Function types (EFs) are displayed.
EFB	When you enable this checkbox, Elementary Function Block types (EFBs) are displayed.
DFB	When you activate this checkbox, Derived Function Blocks (DFBs) are displayed.
Libraries/Families	The scope of the displayed function types and function block types is selected in this column: • To display all function types and function block types used in the project, select **<Application>**. • To display all function types and function block types of a library independently of the library assignment, select **<Libset>**. • To display all function types and function block types of a library, select the desired library. • To display all function types and function block types of a family, select the desired family.
Name	This column displays the names of the function types and function block types and their formal parameters (inputs, outputs).
no.	This column displays the order number of the formal parameters.
Type	This column displays the types of the function types and function block types and the data types of their formal parameters (inputs, outputs).
Value	This column displays the initial value of the formal parameters.

Element	Description
Comment	This column displays comments about the function types and function block types and their formal parameters (inputs, outputs).
R/W Rights of Referenced Variable	This column displays the access rights when the formal parameter is a Reference data type.

Data selection dialog box

Structure of the dialog box

The **Data selection** dialog box consists of four tabs in IL and ST:

- **Variables**

  You can select an elementary variable or an element of a multi-element variable in this tab.

- **Function Blocks**

  You can select a function block instance in this tab.

- **Function and Function Block Types**

  You can select a function type or function block type in this tab.

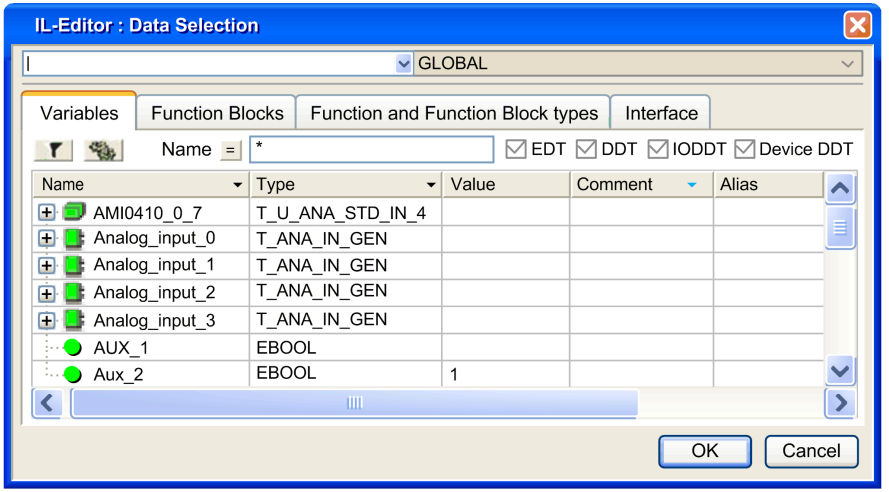
- **Interface**

  You can select a variable of a Program Unit in this tab.

  NOTE: Program Units are only available for Modicon M580 and M340. Without Program Unit in the program structure the **Interface** tab is empty.

Variables tab

Representation of the **Variables** tab:



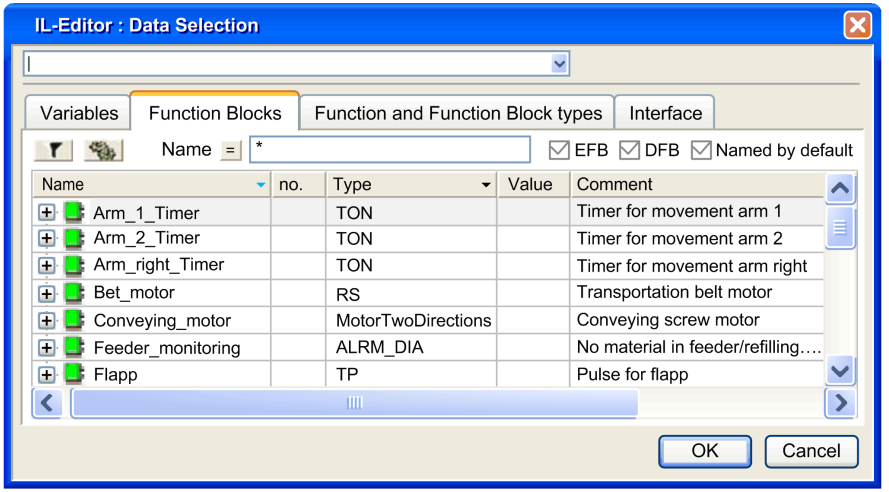
Elements of the **Variables** tab:

Element	Description
Text box	The selected variables are displayed in this text box.
Scope list	Allow to reduce the scope for the variable selection.
button 	Clicking on this button updates the display corresponding to the filter term defined in the **Name** field.
button 	Clicking on this button opens the Filtering data, page 434 dialog box where you can define the conditions for the individual attributes.
Name	The name of the variable to search for can be entered in this text box. You can also use the joker * and **?**.
button 	Clicking on this button inverts the filter. The button changes from = to <> and vice versa.
EDT	When you activate this check box, elementary variables are displayed.

Element	Description
DDT	When you activate this checkbox, multi-element variables (derived variables) are displayed.
IODDT	When you activate this checkbox, derived I/O variables (IODDT variables) are displayed.
Device DDT	When you activate this checkbox, derived DDT are displayed.
Name	The names of the variables are displayed in this column.
Type	The types of the variables are displayed in this column.
Value	The initial value of the variables are displayed in this column.
Comment	The comments for the variables are displayed in this column.
Alias	The alias name of the IODDT, DDT, or Array element are displayed in this column.
Alias of	The complete paths of the variables when an alias name has been set are displayed in this column.
Address	The address in the controller are displayed in this column.
HMI Variable	This column displays if the variable is embedded in the data dictionary.
R/W Rights of Referenced Variable	This column displays the access rights when the variable is a Reference data type.
Effective Parameter	Only when the **Data selection** dialog box is called from a section included in a Program Unit. This column displays the variable assignment for Program Unit parameters (input, output and input/output parameters).
Nature	Only when the **Data selection** dialog box is called from a section included in a Program Unit. This column displays the nature of the Program Unit variables.

Function Blocks tab

Representation of the **Function Blocks** tab:



Elements of the **Function Blocks** tab:

Element	Description
Text box	The selected function block instances are displayed in this text box.
button 	Clicking on this button updates the display corresponding to the filter term defined in the **Name** field.
button 	Clicking on this button opens the Filtering data, page 434 dialog box where you can define the conditions for the individual attributes.
Name	The name of the function block instance to search for can be entered in this text box. You can also use the joker ***** and **?**.
button 	Clicking on this button inverts the filter. The button changes from = to <> and vice versa.
EFB	When you activate this checkbox, the instances of elementary function blocks (EFBs) are displayed.
DFB	When you activate this checkbox, the instances of derived function blocks (DFBs) are displayed.

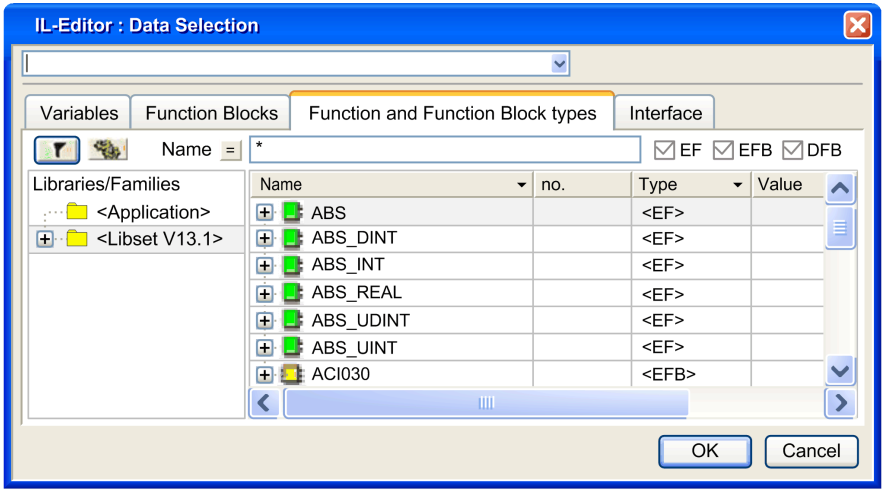
Element	Description
Named by default	When you activate this check box, the function block instances are displayed whose default names (e.g. FBI_3, TON_1) have not been changed.
Name	This column displays the names of the function block instances and their formal parameters (inputs, outputs, public variables).
no.	This column displays the order number of the formal parameters in the function block.
Type	This column displays the types of the function block instances and the data types of their formal parameters (inputs, outputs, public variables).
Value	This column displays the initial value of the formal parameters.
Comment	This column displays comments about the function block instances and their formal parameters (inputs, outputs, public variables).
HMI Variable	This column displays if the formal parameter is embedded in the data dictionary.
R/W Rights of Referenced Variable	This column displays the access rights when the formal parameter is a Reference data type.

NOTE: Help for the Type..

Hold down the **Alt+F1** keys and then click on the function block type for which you wish to access Help.

Function and Function Block Types tab

Representation of the **Function and Function Block Types** tab:



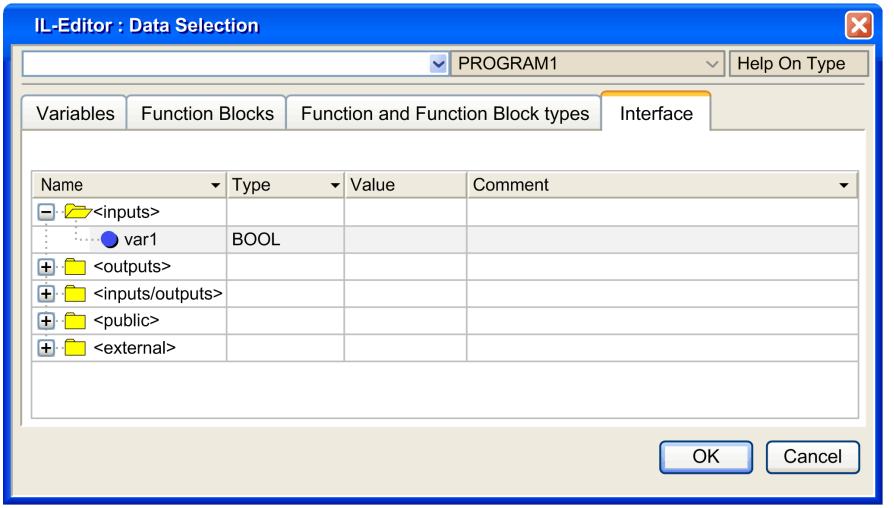
Elements of the **Function and Function Block Types** tab:

Element	Description
Text box	The selected function block types are displayed in this text box.
Help On Type	Use this button to open the help for the function block types shown. **NOTE:** Alternatively, you can also hold down **Alt+F1** and then click on the function block type for which you want to access Help.
button 	Clicking on this button updates the display corresponding to the filter term defined in the **Name** field.
button 	Clicking on this button opens the Filtering data, page 434 dialog box where you can define the conditions for the individual attributes.
Name	The name of the function type or function block type to search for can be entered in this text box. You can also use the joker ***** and **?**.
button 	Clicking on this button inverts the filter. The button changes from = to <> and vice versa.
EF	When you enable this check box, Elementary Function types (EFs) are displayed.
EFB	When you enable this checkbox, Elementary Function Block types (EFBs) are displayed.
DFB	When you activate this checkbox, Derived Function Blocks (DFBs) are displayed.
Libraries/Families	The scope of the displayed function types and function block types is selected in this column: • To display all function types and function block types used in the project, select **<Application>**. • To display all function types and function block types of a library independently of the library assignment, select **<Libset>**. • To display all function types and function block types of a library, select the desired library. • To display all function types and function block types of a family, select the desired family.
Name	This column displays the names of the function types and function block types and their formal parameters (inputs, outputs).
no.	This column displays the order number of the formal parameters.
Type	This column displays the types of the function types and function block types and the data types of their formal parameters (inputs, outputs).
Value	This column displays the initial value of the formal parameters.

Element	Description
Comment	This column displays comments about the function types and function block types and their formal parameters (inputs, outputs).
R/W Rights of Referenced Variable	This column displays the access rights when the formal parameter is a Reference data type.

Interface tab

Representation of the **Interface** tab:



Elements of the **Interface** tab:

Element	Description
Text box	The selected function block instances are displayed in this text box.
Scope list	Allow to select the data scope for the variable selection among the different Program Units.
Name	This column displays the names of the variables belonging to the Program Unit selected into the scope list. The variables are grouped by **Nature** (**<inputs>**, **<outputs>**, **<inputs/outputs>**, **<public>**, **<external>**).
Type	This column displays the data types of the variables.
Value	This column displays the initial value of the variables.

Element	Description
Comment	This column displays the comment for the variables.
Effective Parameter	Only when the **Data selection** dialog box is called from a section included in a Program Unit. This column displays the variable assignment for Program Unit parameters (input, output and input/output parameters).
Address	Only when the **Data selection** dialog box is called from a section included in a Program Unit. The address in the controller are displayed in this column.

Function Input Assistant

Introduction

If **no**FFB is selected when the function input assistant is called the FFB input assistant is used for assigning parameters to the new FFBs generated.

Example for:

- FBD, page 671
- LD, page 808
- IL, page 1066
- ST, page 1097

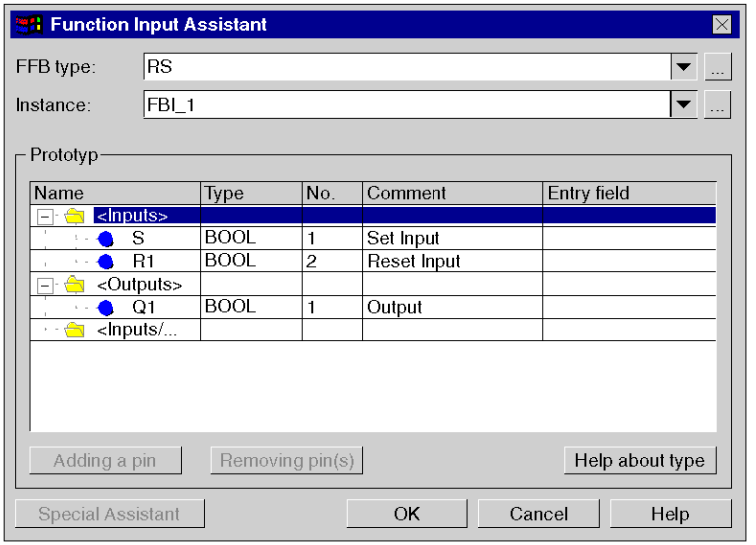
If an FFB is selected when the function input assistant is called the FFB input assistant is used for assigning parameters to the new FFB generated.

Example for:

- FBD, page 689
- LD, page 828
- IL, page 1076
- ST, page 1106

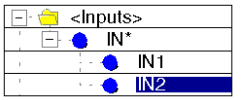
Function Input Assistant

Representation:



Elements:

Element	Description
FFB type	Enter the function type or function block type to edit in this text box. You have the following options to enter the function type or function block type: • You can directly enter the type. • You can select the type from the list of recently used types using the ▾ symbol. or • Using the button **...** you can open a dialog box to select FFB , page 1188 types.
Instance	For existing function blocks this textbox shows the name of the selected function block instance. For function blocks to be generated enter the name of the selected function block instance in this textbox. You have the following options to enter the function instance or function block instance: • You can directly enter the instance. • You can select the instance from the list of recently used instances using the ▾ symbol. or • Using the button **...** a dialog box for FFB instances selection, page 1187 .

Element	Description
Name	This column displays the formal parameters (inputs, outputs) of the function instance or function block instance.
Type	This column displays the data types of the formal parameters (inputs, outputs) of the function instance or function block instance.
No.	This column displays the pin positions of the formal parameters (inputs, outputs) of the function instance or function block instance.
Comment	This column displays the comments of the formal parameters (inputs, outputs) of the function instance or function block instance.
Entry field	Enter the actual parameters of the function instance or function block instance in this column. You have the following options to enter actual parameters: • You can directly enter the actual parameter. • You can select the actual parameter from the list of recently used actual parameters using the ▼ symbol. or • Using the button **...** a dialog box for variables selection, page 1181 .
Adding a pin	Using this button you can increase thenumber of inputs, page 692 for expandable functions (e.g. AND, OR). Expandable pins can be recognized from the following display:  To add further pins select the last pin in the structure and confirm using the **Add pin** button.
Removing pin(s)	Using this button you can decrease the number of inputs, page 692 for expandable functions (e.g. AND, OR). To decrease the number of inputs select the last pin(s) in the structure and use the **Remove pin(s)** button.
Help about type	Use this button to call up the FFB.
Special Assistant	Some complex FFBs provide additional windows for entering actual parameters. To access these windows use the **Special Assistant** button. **NOTE:** take care that for the same FFB the **Special Assistant** could be available for some platforms and not for other platforms.
OK	For new FFBs the FFB insert mode for placing FFBs becomes active. Use this button to accept the allocated actual parameters for existing FFBs.
Close	With this button, you can close the dialog box without accepting changes.
Help	Use this button to call up the dialog.

Data Properties

Introduction

The anchored window, page 1827 **Data Properties** is used to display and modify the variables and FFB properties.

The **Data Properties** window can be called up

- using the Data Editor, page 348.

1. Select one or several lines in the Data Editor.
2. Select **Properties** from the context menu.

- the programming language sections

1. Select one or several element(s) in the programming language section.
2. Select **Data Properties** from the context menu or press **CTRL+Enter**.

- the tab **I/O Objects** of a module

1. Select one or several lines in the **Address/Name/Type** table.
2. Select **Data Properties** from the context menu.

How the window behaves and the options in the **Data Properties** window depend on whether the window was opened using the Data Editor or the Programming Languages section.

NOTE: If more than one object is selected, only the common data for all the selected objects is displayed. If this data is modified, the modification applies to all the selected objects. In this way it is possible to assign the same comment to different variables or assign the same initial value, see also: Changing the attribute in several instances, page 420

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

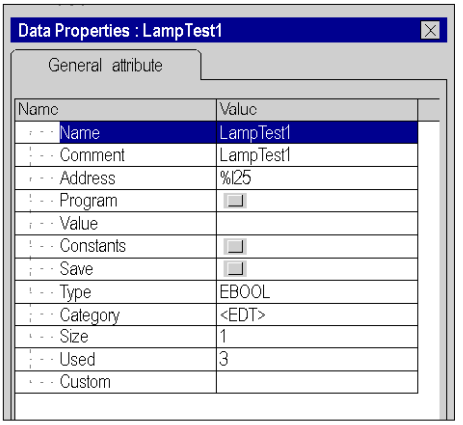
Before changing data in the **Data Properties** window, ensure that the change(s) is(are) appropriate not only to the selected instance object, but also to all other selected objects.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

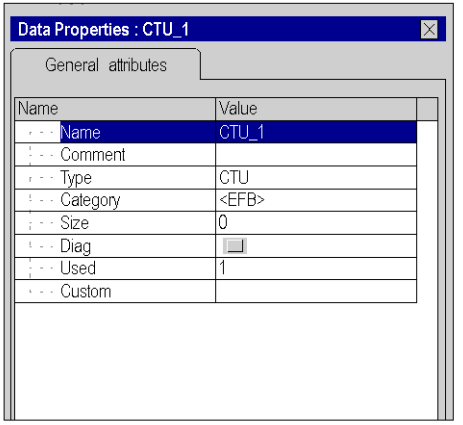
Structure of the Window

The contents of the window depend on the selected object:

Window **Data Properties** for elementary variables:



Window **Data Properties** for function blocks:



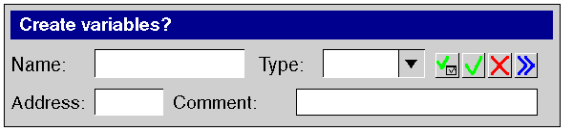
Available Attributes

The list of available attributes is the same as the **Data Editor** columns. Refer to the Data Editor columns, page 426 for more information.

Generate variables

Representation

Representation of the dialog box:



Items

Elements of the dialog box:

Element	Description
Name	Name of the variables to generate (maximum length = 32 characters).
Type	Datatype of the variables to generate. If you want to allocate the variable for the elementary data type, you can select it from the list. If you want to allocate the variables for the derived data type, enter the data type in the text box.
	This button is only available in the IL and ST editor under the following conditions: • The cursor is located where a variable can be inserted (i.e. not in a word). • The name of the variable is automatically entered in the text box **Name**. You can confirm the entries using this button and the variable will be declared and inserted in the section.
	In the FBD and LD-Editor: You can confirm the entries using this button and the variable will be declared and (if not already inserted) inserted in the section. In the IL and ST Editor: You can confirm the entries using this button and the variable will be declared (but not inserted in the section).
	Cancel the generate of the variables using this button.
	Use this button to show or hide the **Address** and **Commentary**text boxes.

Element	Description
Address	Address of the variables to generate (optional).
Comment	Comment for the variables to generate (optional).

I/O Objects tab

What's in This Chapter

Introduction to the I/O Object Tab 1207
I/O Objects Tab for a Processor..... 1209
I/O Objects Tab for a Module..... 1211
Details of Areas in the I/O Objects Tab..... 1213
I/O Objects Tab for a Bus Device..... 1219
How to Use the I/O Objects Tab 1219

Subject of this Chapter

This chapter describes the **I/O objects** tab, which is used to associate variables with the module I/Os and to manage these different variables.

Introduction to the I/O Object Tab

General Points

This tab is used to manage the I/O objects of a module, a field bus device or the memory and system objects of the PLC.

The following table summarizes the different functionalities according to the type of device associated with the tab:

Function	Processor	Module or bus device
Displaying all located objects of the PLC	X	-
Displaying I/O objects (topological addressing)	X (1)	X (2)
Filtering all located objects of the PLC	X	-
Filtering I/O objects	X (1)	X (2)
Sorting all located objects of the PLC	X	-
Sorting I/O objects	X (1)	X (2)
Creating an IODDT variable for a module	-	X
Creating an IODDT variable for a channel	-	X
Creating a topological EDT variable	-	X

Function	Processor	Module or bus device
Presymbolizing a set of IODDT variables for a module	-	X
Presymbolizing a set of topological EDT variables	-	X
For Quantum and Modicon M340 PLCs, displaying the correspondence between the topological address and State RAM.	X	X
Legend: **X**: available functionality, **-**: non-available functionality **(1)**: of all I/O objects of the PLC **(2)**: only the I/O objects of the module or the device concerned.		

Access Rules

To access this tab, the following conditions must be met:

- The tab can be accessed from the configuration screen for the module, processor or bus device (it is located at the same level as the description tab).
- The tab is accessible in offline and online mode.

Operating Rules

The main operating rules are as follows:

- This tab can be used to create and associate variables, but not to delete them.
- To delete a variable created by this tab, you must use the data editor.
- Multiple presymbolization can only be used on compatible objects (of the same type).
- For the processors, the tab only allows PLC objects to be displayed (variables cannot be associated).
- The control loops built into the processor are regarded as module channels, and consequently, creation and presymbolization are possible.
- It is possible to name a variable derived after symbolization.

  For example, if `Counter_0` is a `T_COUNT_STD` type variable, the presymbolized counting value `Counter_0.CUR_MEASURE` can be renamed (`Value_C0`, for example).

I/O Objects Tab for a Processor

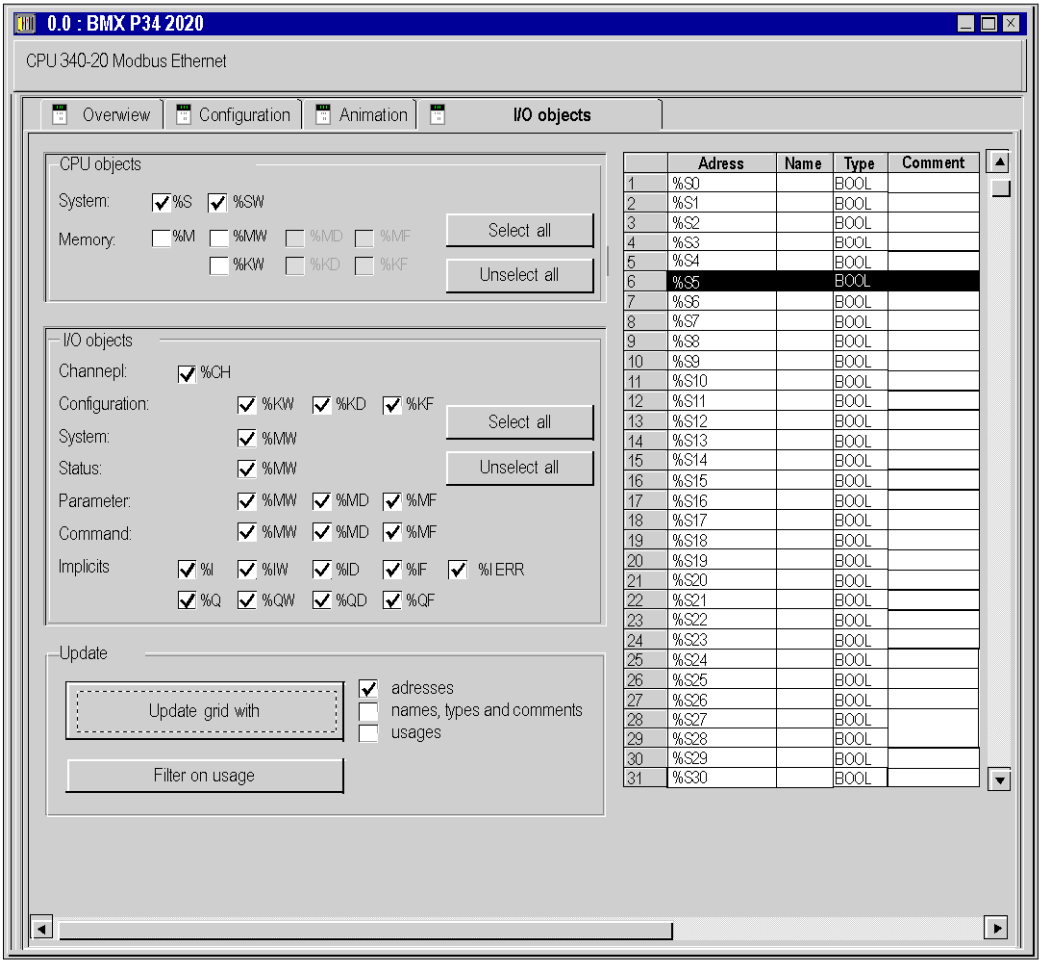
At a Glance

The processor I/O Objects tab can be used to view PLC objects (system objects, memory objects) as well as I/O objects, regardless of whether they belong to an in-rack module or to a communication bus device (for example, Fipio).

NOTE: This screen can only be used to view objects; for modifications, use the **I/O objects** tab associated with the modules or bus devices, page 1211.

Illustration

Example of an I/O Objects screen for a Modicon M340 processor.



Description

The different areas of the screen are as follows:

Zone	Description
CPU objects, page 1214	This zone can be used to select PLC system and memory objects that you would like to view in the **Address Name Type Comment** zone.
I/O Objects, page 1215	This zone can be used to select processor I/O objects that you would like to view in the **Address Name Type Comment** zone.
Update, page 1218	This zone allows you to start the update of the information in the **Address Name Type Comment** zone.
Address Name Type Comment, page 1217	Zone used to view and select data. The contents of this zone are obtained using the buttons and selections of the previous zones. The Quantum **State RAM** column is added.

I/O Objects Tab for a Module

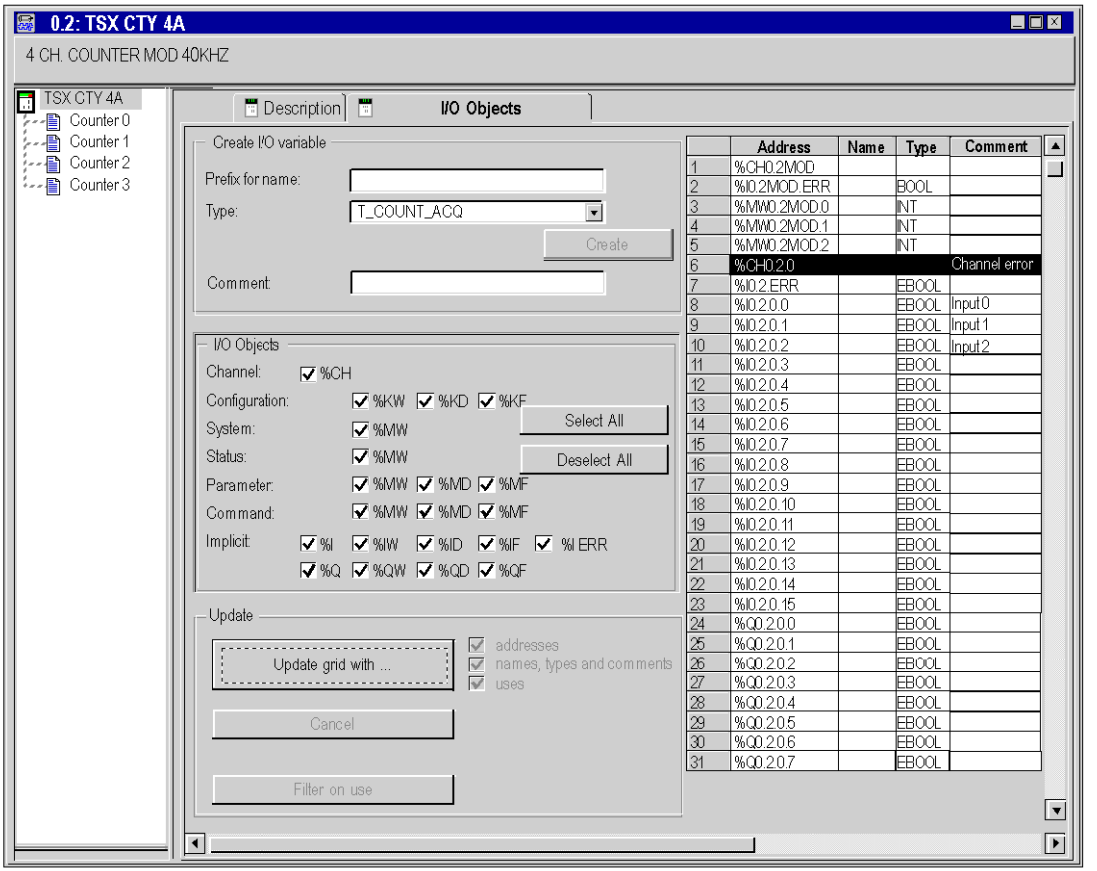
At a Glance

For a module, this tab can be used to view module I/O objects and to manage the association of these objects with variables.

NOTE: To view PLC objects (system, internal memory or internal constant objects), use the **I/O objects** tab associated with the processor, page 1209.

Illustration

Example of an I/O objects screen for an Premium X bus module.



Description

The different areas of the screen are as follows:

Zone	Description
Create I/O variable, page 1216	This zone can be used to create variables and to associate them with channels or channel elements with a comment for each variable.
I/O Objects, page 1215	This zone can be used to select processor I/O objects that you would like to view in the **Address Name Type Comment** zone.
Update, page 1218	This zone allows you to start the update of the information in the **Address Name Type Comment** zone.
Address Name Type Comment, page 1217	Zone used to view and select data. The contents of this zone are obtained using the buttons and selections of the previous zones. For Quantum and Modicon M340, the **State RAM** column is added.

NOTE: For Modicon M340 I/O modules only the topological I/O objects are displayed. It is not possible to instantiate an IODDT.

Details of Areas in the I/O Objects Tab

At a Glance

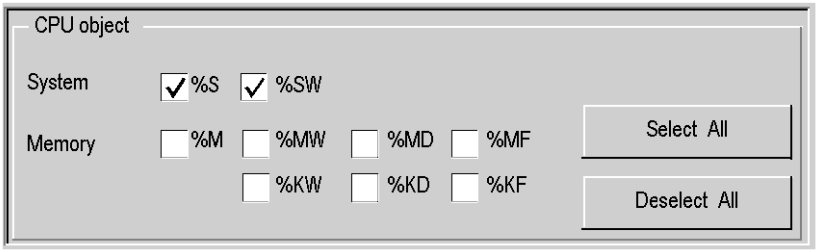
The I/O Objects tab is available for:

- Modules in the rack and their channels,
- Processors and their channels,
- Process control loops,
- Communication bus devices.

The following paragraphs give details of the areas available from this tab.

Illustration of the CPU Objects Area

The CPU Objects area is only available for the processor. This appears as follows:



Description of the CPU Objects Area

The selection of different objects using checkboxes %S, %SW, %M, %MW, %MD, %MF, %KW, %KD and %KF enables these to be displayed in the **Address Name Type Comment** area for Premium or the **Address Name Type State RAM Comment** area for Modicon M340 and Quantum, once the **Update Grid with...** button is pressed, in the **Update** area.

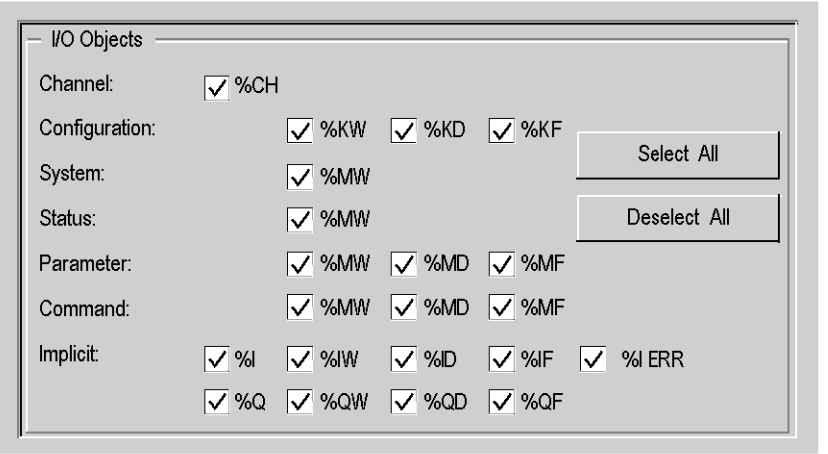
NOTE: For Premium/Atrium PLCs double-type instances of located data (%MD<i>, %KD<i>) or floating (%MF<i>, %KF<i>) should be located by an integer type (%MW<i>, %KW<i>). Only I/O objects make it possible to locate type instances (%MD<i>, %KD<i>, %QD, %ID, %MF<i>, %KF<i>, %QF, %IF) by using their topological address (for example %MD0.6.0.11, %MF0.6.0.31).

NOTE: For Modicon M340 PLCs, double-type instances of located data (%MD<i>, %KD<i>) or floating (%MF<i>, %KF<i>) are not available.

NOTE: For Modicon M340 PLCs, the index (i) value must be even (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) for double-type instances of located data (%MW and %KW).

Illustration of the I/O Objects Area

The I/O Objects area is available for modules, processors, control loops and bus devices. It appears as follows:



Description of the I/O Objects Area

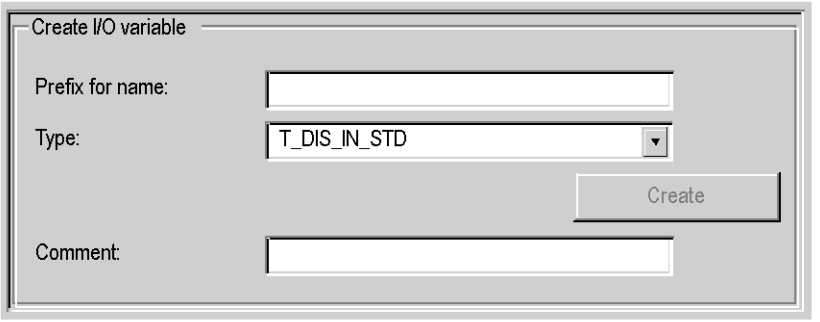
The selection of different objects using checkboxes enables these to be displayed in the **Address Name Type Comment** area, once the **Update Grid with...** button is pressed, in the **Update** area.

The different objects can be selected by type:

- **Channel**: for module channels or a bus device.
- **Configuration**: for configuration language objects.
- **System**: for language objects managing explicit exchanges.
- **State**: for status language objects (accessible by READ_STS).
- **Parameter**: for configuration language objects (accessible by READ_PARAM, WRITE_PARAM, SAVE_PARAM, RESTORE_PARAM).
- **Command**: for command language objects (accessible by WRITE_CMD).
- **Implicit**: for implicit language objects of the module or a bus device.

Illustration of the Create I/O Variable Area

The **Create I/O Variable** area is only available for modules, control loops and bus devices. It appears as follows:



Description of the Create I/O Variable Area

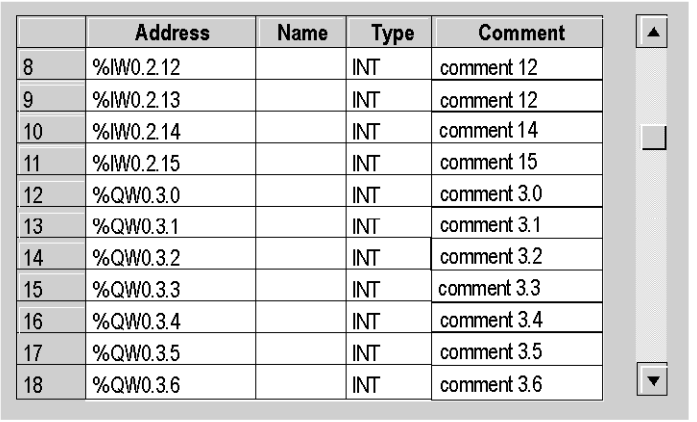
Having selected one or more objects in the **Address Name Type Comment** area, you can select an IODDT type and create one or more variables of this type by clicking **The I/O Objects tab is available for:Create**.

Operating rules:

- By selecting a line in the **Address Name Type Comment** area, you can create a variable for which you can enter the name. You can also assign a comment to the variable.
- By selecting several homogenous (same type) lines from the **Address Name Type Comment** area, you can automatically create several variables with identical prefixes (the first variable with the suffix 0 the second 1, the third 2, etc.). This method also applies for the variable comment (the first comment will have the suffix 0, the second 1, and the third 2, etc.).
- When the selected variable(s) is/are of EDT type, the type area is grayed out. Type selection is only available when several types are available.

Illustration of the Address Name Type Comment Area

The Address Name Type Comment area is available for all I/O Objects tabs. These appear as follows:



	Address	Name	Type	Comment
8	%IW0.2.12		INT	comment 12
9	%IW0.2.13		INT	comment 12
10	%IW0.2.14		INT	comment 14
11	%IW0.2.15		INT	comment 15
12	%QW0.3.0		INT	comment 3.0
13	%QW0.3.1		INT	comment 3.1
14	%QW0.3.2		INT	comment 3.2
15	%QW0.3.3		INT	comment 3.3
16	%QW0.3.4		INT	comment 3.4
17	%QW0.3.5		INT	comment 3.5
18	%QW0.3.6		INT	comment 3.6

Description of the Address Name Type Comment Area

This area enables you to:

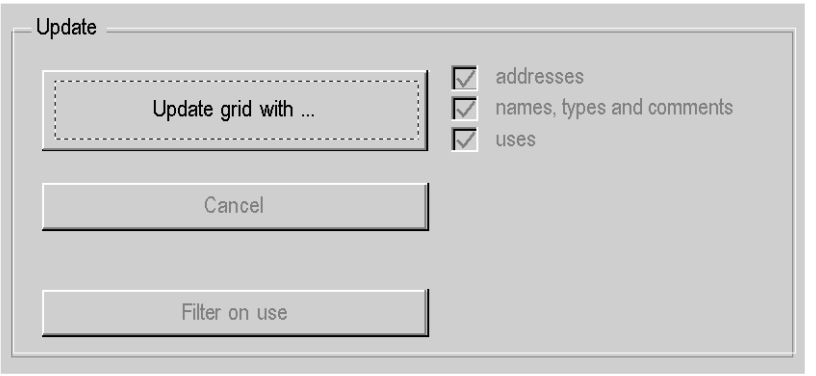
- Display the objects selected in the **CPU Objects** and **I/O Objects** areas,
- Select one or more object lines in order to create variables and associate variables with these,
- Open, page 1222 the **Data Properties** window,
- View the comment associated with the variable.

For Quantum PLCs, an additional column is available:

- **State RAM** is used to view the correspondence between the topological address and State RAM.

Illustration of the Update Area

The Update area is available for all **I/O Objects** tabs. These appear as follows:



Description of the Update Area

By clicking the **Update Grid with...** button, you can update the **Address Name Type Comment** area from the information selected in the **CPU Objects** and **I/O Objects** areas.

The **Address Name Type Comment** area is also used to select objects in order to create variable names and comments for them (only true for modules, control loops and communication bus devices).

The **Cancel** button is used to cancel the update of object names from the **Address Name Type Comment** area.

The **Filter on use** button is used to display only those objects used in the project.

The **addresses**, **names, types and comments** and **use** checkboxes are used, respectively, to display:

- The **Address** column,
- the **Name** and **Type** and **Comments** columns,
- In bold the variables used in the program.

I/O Objects Tab for a Bus Device

At a Glance

All bus devices that have a description window also have an **I/O objects** tab.

Currently, **Fipio** (Premium) and **CANopen** (Modicon M340 and Modicon M580) bus devices have an **I/O objects** tab. This tab behaves in exactly the same way as the I/O objects of a module, page 1211 tab.

How to Use the I/O Objects Tab

At a Glance

The I/O tab is used to facilitate the management of located variables in an automation project using Control Expert.

For information on the different possible actions, refer to the pages describing this tab for:

- Processors, page 1209
- Modules, page 1211
- Communication bus devices, page 1219

A description of the main procedures for carrying out these actions is given below.

How to Display Required Objects

The following table shows the procedure for displaying a list of objects associated with a processor, module or communication bus device.

Step	Action
1	In the **CPU objects**, page 1214 area or **I/O objects**, page 1215 area, check the boxes of the objects you want to display (%S, %SW, %MW, %CH, %M, %MW, %I, %Q, etc.). **Note**: The **Select All** and **Deselect All** buttons can be used to check or un-check all boxes.
2	In the **Update**, page 1218 area, click **Update Grid with...**. **Result**: The **Address Name Type Comment**, page 1217 area is updated and displays all the selected variables available on the processor, module or current bus device. **Note**: When searching for names, it is possible to cancel an update in progress by clicking **Cancel**.

How to Select an Object

The following table shows the procedure for selecting an object associated with a processor, module or communication bus device.

Step	Action
1	In the **Address Name Type Comment**, page 1217 area, click the line of the object to select. **Result**: The line appears highlighted, and the **Create I/O variable**, page 1216 area becomes accessible for modules and communication bus devices, for creating a variable associated with the selected line. **Note**: If the **Address Name Type Comment** area is empty or contains no object to select, see the section **How to Display Required Objects**, page 1219.

How to Create a Variable for an Object

The following table shows the procedure for creating a variable for a module object or communication bus device object.

Step	Action
1	In the **Address Name Type Comment**, page 1217 area, click the line of the object to select. **Result**: The line appears highlighted, and the **Create I/O variable**, page 1216 area becomes accessible for modules and communication bus devices, for creating a variable associated with the selected line. **Note**: If the **Address Name Type Comment** area is empty or contains no object to select, see the section **How to Display Required Objects**, page 1219.
2	In the **Create I/O variable**, page 1216 area, select the type of variable to create (if available).

Step	Action
	Note: When the variable is an EDT type (such as DINT, INT, BOOL, EBOOL type variables) or when the available IODDT is unique, the choice of type is not available.
3	In the **Create I/O variable**, page 1216 area, enter the name of the variable to be created.
4	In the **Create I/O variable**, page 1216 area, enter the comment for the variable to be created.
5	In the **Create I/O variable**, page 1216 area, click **Create**. **Result**: A new variable is created, which is associated with the object selected in the **Address Name Type Comment**, page 1217 area.

How to Select a List of Objects

The following table shows the procedure for selecting several objects associated with a module or communication bus device.

Step	Action
1	In the **Address Name Type Comment**, page 1217 area, click the left column (gray column of object numbers) on the first object to select.
2	Hold down the mouse button and slide the cursor down to the last element to select. **Result**: The list of selected objects appears highlighted.

How to Create Variables Associated with a List of Objects

The following table shows the procedure to create variables associated with several homogenous objects from a module or communication bus device.

Step	Action
1	In the **Address Name Type Comment**, page 1217 area, click the left column (gray column of object numbers) on the first object to select.
2	Hold down the mouse button and slide the cursor down to the last element to select. **Note**: If the **Address Name Type Comment** area is empty or contains no objects to select, see the section **How to Display Required Objects**, page 1219 in order to select a contiguous list of objects to select.
3	In the **Create I/O variable**, page 1216 area, select the type of variable to create (if available). **Note**: When the selected variables are of EDT type (such as DINT, INT, BOOL, EBOOL type variables) or when a single IODDT is available, the choice of type is not available.

Step	Action
	If heterogeneous (not of the same type) objects appear in the list, the creation of variables will not be available. You must therefore select a list of homogenous objects.
4	In the **Create I/O variable**, page 1216 area, enter the prefix to be associated with all selected objects. **Note**: This prefix will be associated with a number to form the name of each variable. The first object will be number 0, the second 1, the third 2, etc. **Example**: If you enter the prefix **Valve**, the created variables will be: Valve0, Valve1, Valve2, Valve3, etc.
5	In the **Create I/O variable**, page 1216 area, enter the prefix of the comment to be associated with all selected objects. **Note**: This prefix will be associated with a number to form the comment for each variable. The first object will be number 0, the second 1, the third 2, etc. **Example**: If you enter the prefix **Comment**, the created comments will be: Comment0, Comment1, Comment2, Comment3, etc.
6	In the **Create I/O variable**, page 1216 area, click **Create**. **Result**: The new variables are created and appear in the **Address Name Type Comment**, page 1217 area.

How to Access Data Properties

The following table provides a step-by-step procedure for accessing the **Data Properties** window from the I/O tab.

Step	Action
1	In the **Address Name Type Comment**, page 1217 area click on the variable whose properties you wish to display.
2	Right-click on **Properties**. **Result**: the **Data Properties** window is displayed. Once it is open, this window displays the properties of the variable selected in the **Address Name Type Comment** area.

Frequently Asked Questions

Below is a list of the most commonly asked questions you may have on the use of the I/O Objects tab.

Question	Answer
How can I sort the objects list into alphabetical order in a column?	Simply click on the name of the column to be sorted.
How can I display the variables and objects used in my project?	The variables and objects used in a program appear in bold and the others are not yet used.
How can I filter the objects in the **Address Name Type Comment** area?	Use the check boxes of the **Update** area.
Why are the **Address Name Type Comment** or **Address Name Type State RAM Comment** areas empty?	Possible reasons include: • You have not yet updated the area by clicking **Update Grid with...**. • You have not yet checked the boxes in the **CPU Objects** or **I/O Objects** areas. • No object for which the boxes are checked appears in this module, processor or communication bus device.
Why is the Quantum **State RAM** column empty?	Possible reasons include: • The **Address Name Type State Ram Comment** area is empty, refer to the previous question. • The elements displayed are not associated with State RAM.
Why is variable creation not available?	Possible reasons include: • You are in the processor, and it is therefore impossible to display the objects. • You have selected a list of heterogeneous objects, and it is only therefore possible to create lists of variables for objects of the same type. • You have selected an object that is already associated with a variable. You must go into the data editor to delete this variable, in order to create another. • You have selected a list of objects of which at least one is already associated with a variable (see previous point).

Memory Tab

What's in This Chapter

Memory Tab 1225

Subject of this Chapter

This chapter describes the **Memory** tab, which is only available for Quantum and Modicon M340 PLCs that are equipped with a State RAM.

Memory Tab

Overview

The **Memory** tab is only available for the Modicon M340 hardware platform equipped with a State RAM memory area.

It allows defining the type of memory management for the selected input / output module.

The option you select here must comply with the type of memory management you selected for the processor, page 1348.

The following settings are possible:

If...	Then...
Topological memory management is configured for the processor	you can only select **Topological** memory management for the discrete and analog modules.
Mixed topological and State RAM memory management is configured for the processor	you can select either **Topological** or **Mixed topological and State RAM** memory management for the discrete and analog modules.

The memory management settings of the modules are compared with the memory management settings of the processor during **Analyze**. If any inconsistencies are detected, an appropriate message is displayed in the output window.

The type of memory management you selected for the processor will be used as default setting for the **Memory Management** parameter in the **Memory** tab.

Any application made with Control Expert is automatically configured with address configuration for topological addressing as the default setting.

NOTE:

To import a legacy LL984 Compact application that uses Modbus requests to communicate with an HMI device, use State RAM addressing to preserve the Modbus exchange between the controller and the HMI.

Specific Configuration Dialogs in Case of LL984 Import

If an ST section has been created during import to emulate Compact configuration behavior (see EcoStruxure™ Control Expert, LL984 Editor, Reference Manual), the converter creates a specific I/O module configuration with specific configuration dialogs (see EcoStruxure™ Control Expert, LL984 Editor, Reference Manual).

The **Memory Management** area is grayed and the choice is fixed to **Mixed topological and State RAM**.

In the **State RAM** area **Format** and **Coding** are defined by the converter.

You can only delete these modules. Copy and paste of these modules is not possible.

Memory Management

The area **Memory Management** of the **Memory** tab provides the following options:

Option	Description
Topological	Select this option to work with pure topological addressing. If this mode is selected, the parameters in the **State RAM** area of this dialog box are disabled.
Mixed topological and State RAM	Select this option to work with State RAM addressing for IO access. Topological addressing remains possible for some IO objects (%CH, %I.ERR, ...) but it's not possible to access IO values (%I, %Q). With this option selected, define further State RAM options in the **State RAM** area of this tab for the different types of modules. **NOTE:** With this type of memory management, all channels are associated with only one task (MAST or FAST).

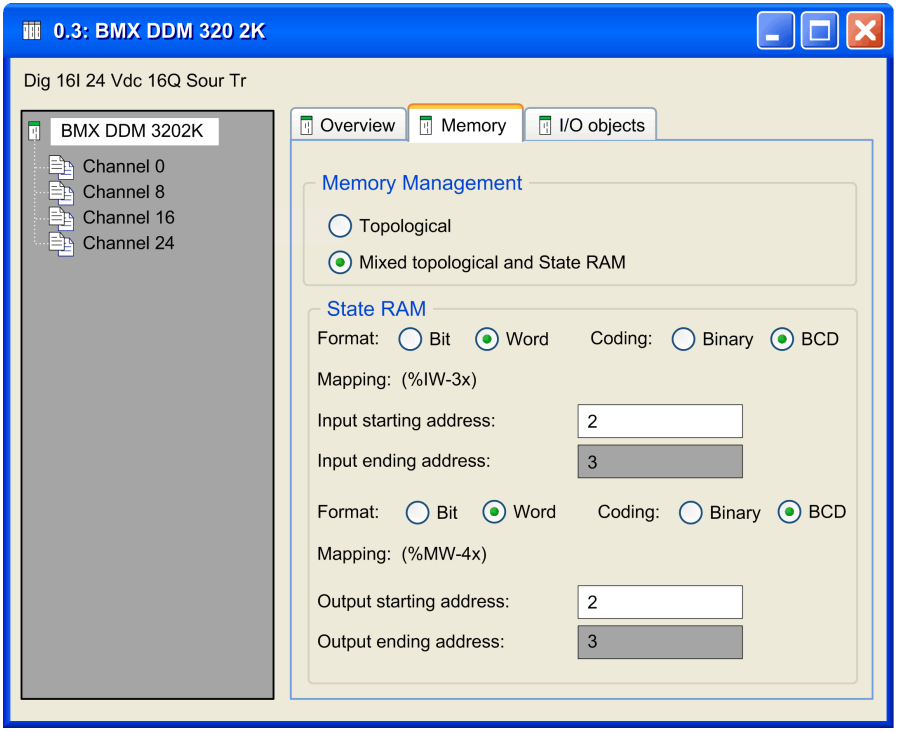
NOTE: If you select **Mixed topological and State RAM** for **Memory Management** the following online features are disabled:

- **I/O Objects**
- **Memory**
- **Debug**
- **Fault**

State RAM Configuration for Discrete Modules

Depending on the selected module, the **State RAM** area of the **Memory** tab contains parameters for discrete input modules, discrete output modules or for discrete mixed modules. They are available if the option **Mixed topological and State RAM** is selected in the **Memory Management** area.

The figure below shows the **Memory** tab for a discrete mixed input / output module as an example:



The **State RAM** area contains the following parameters:

Parameter	Description
Format	• **Bit**: Select this option for channel mapping in the discrete mode. This means that each channel is mapped to a %I (inputs) or %M (outputs). • **Word**[1]: Select this option for channel mapping in the word mode. This means that a group of 8 channels is mapped to a %IW (inputs) or %MW (outputs).
Coding	This parameter is only available for channel mapping in word mode, that is if the parameter **Format** is set to **Word**. • **Binary**: Select this option to use the binary integer format. • **BCD**: Select this option to use the BCD coding format.
Mapping	This parameter allows you to define the **starting address** for the inputs and outputs. The following sizes of the memory areas are possible: • For inputs in **Bit** format: %I-1x • For inputs in **Word** format: %IW-3x • For outputs in **Bit** format: %M-0x • For outputs in **Word** format: %MW-4x **NOTE:** The State RAM address starts at address 0, but the first available address for discrete and analog modules is %I1, %M1, %IW1, and %MW1.

(1) When **Word** is selected as the **Format**, the LSB of the **Word** is mapped to:

- Input 0 on discrete input modules
- Output 0 on discrete output modules

NOTE: All channels of a discrete module must be configured. It is not possible to leave a group of channels un-configured.

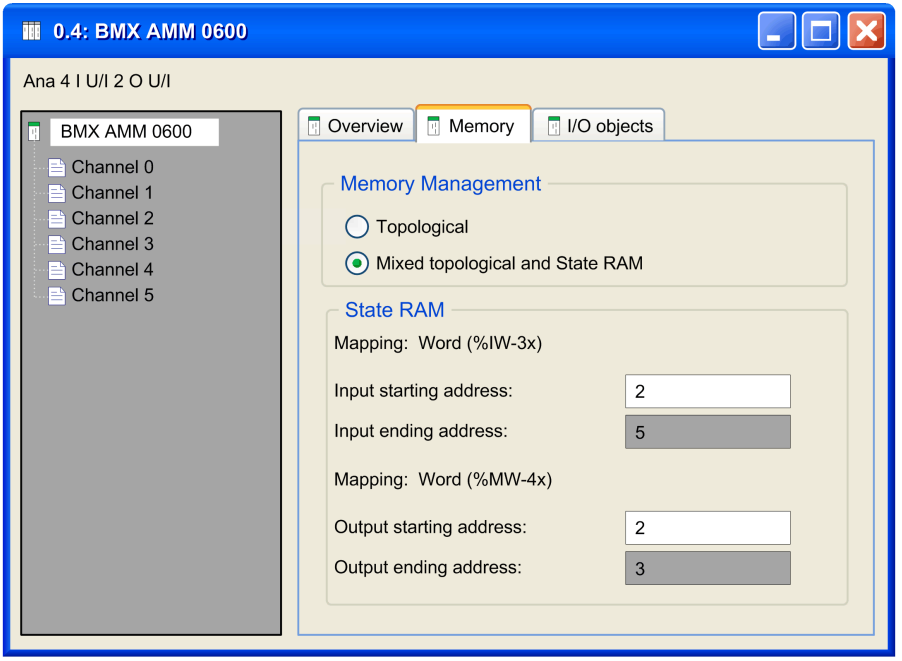
NOTE: You can use State RAM objects in event sections, but they are refreshed with the frequency of the task where they are declared.

NOTE: State RAM addresses are not checked during input. If any range overlapping is detected or any address exceeds the range of the State RAM, an appropriate message is displayed at the validation screen or during **Analyze**.

State RAM Configuration for Analog Modules

Depending on the selected module, the **State RAM** area of the **Memory** tab contains parameters for analog input modules, analog output modules or for analog mixed modules. They are available if the option **Mixed topological and State RAM** is selected in the **Memory Management** area.

The figure below shows the **Memory** tab for a mixed analog input / output module as an example:



The **State RAM** area contains the following parameters:

Parameter	Description
Mapping	This parameter allows you to define the **starting address** for the inputs and outputs. The following sizes of the memory areas are possible: • For inputs in **Word** format: `%IW-3x` • For outputs in **Word** format: `%MW-4x`

NOTE: State RAM addresses are not checked during input. If any range overlapping is detected or any address exceeds the range of the State RAM, an appropriate message is displayed at the validation screen or during **Analyze**. For instance, if a located variable is created in the Data Editor, using a %MW IO map word of a configured IO base, the following build error will occur: ‘Error: Memory overlapping with protected memory address '%MWi'. Please relocate the resources in '%MWi' to avoid overlapping.’ (%MWi is the address that have to be changed in the **Data Editor**).

Type Library Browser

What's in This Chapter

Access to the Type Library Browser 1231
Description of the Type Library Browser 1233
Configuration of Columns of the Type Library Browser 1235
Drag & Drop of FFBs in a program section 1237

Overview

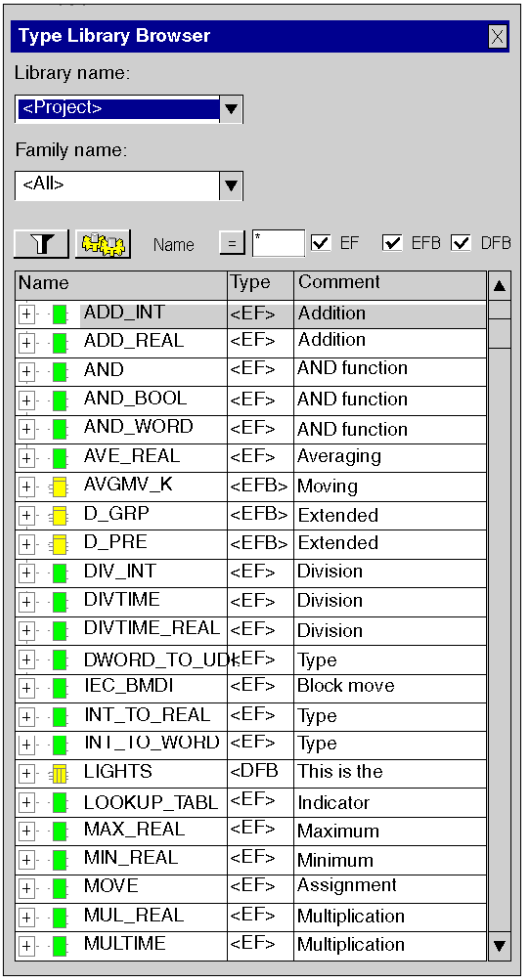
This chapter describes the type library browser.

Access to the Type Library Browser

At a Glance

Functions and function blocks (FFBs) can be dragged from the type library browser into a program section and stored (Drag & Drop).

Accessing the Browser

Step	Action
1	In the **Tools** menu, select the **Type Library Browser** option or click the **Type Library Browser** button  in the language editor toolbar. (To close the **Type Library Browser** click the button again).
2	**Result:** The following window appears. 

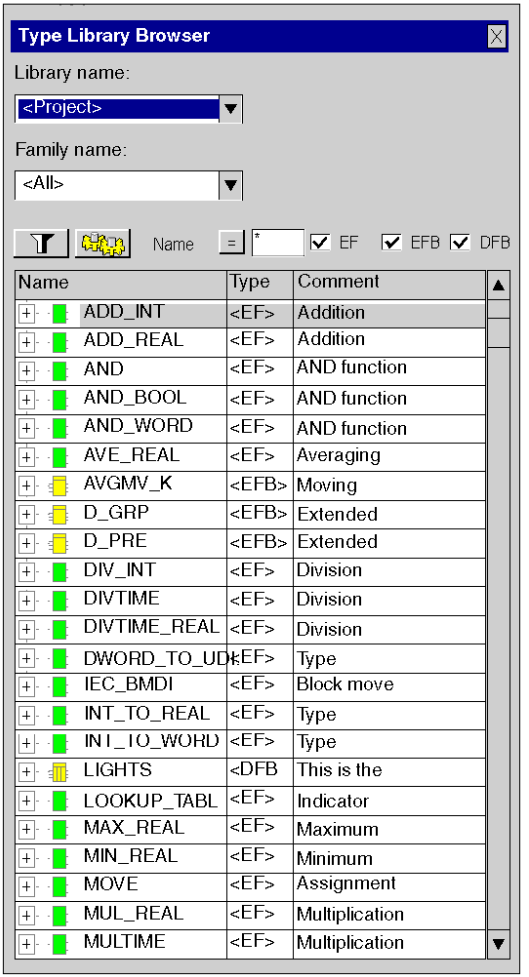
Description of the Type Library Browser

Calling the Type Library Browser

See Access to the Type Library Browser, page 1231.

Structure of the window

Appearance:



Elements:

Element	Description
Library name	The scope of the displayed function types and function block types is selected in this column: • To display all function types and function block types used in the project, select **<Project>**. • To display all function types and function block types of a library independently of the library assignment, select **<Libset>**. • To display all function types and function block types of a library, select the desired library.
Family name	To display all function types and function block types of a library family, select the desired family.
Button 	Click this button to update the display corresponding to the filter term defined in the **Name** field.
Button 	Click this button to open a dialog box, page 433 for defining the filters.
Button 	Click this button to invert the filter. The button changes from **=** to **<>** and vice versa.
Name (Text Box)	The name of the functions and function block types to search for can be entered in this text box. You can also use the joker ***** and **?**.
EF	When you enable this check box, Elementary Function types (EFs) are displayed.
EFB	When you enable this checkbox, Elementary Function Block types (EFBs) are displayed.
DFB	When you enable this checkbox, Derived Function Blocks (DFBs) are displayed.
Name (List)	This column displays the names of the function types and function block types and their formal parameters.
Type	This column displays the types of the function types and function block types.
Comment	This column displays the comments for the function types and function block types.

Closing the **Type Library Browser** the currently selected **Library name**, **Family name** and type check boxes (**EF**, **EFB**, **DFB**) are saved. If you open the **Type Library Browser** again, these items will be preset.

Shortcut menu structure

Additional different commands can be executed via the context menu of the Type Library Browser.

Shortcut menu commands:

Menu command	Description
Copy into Project	This menu command is only available for functions and function block types that are not used yet in the project. Use this menu command to load the selected function and function block types into the project. Select via drop down list the **Library Name** which makes it quicker to access the type in the project.
Expand All	Use this menu command to display the formal parameters of the selected function and function block type. The same function is also available above the **+** symbol in front of the function block type names.
Collapse	Use this menu command to hide the formal parameters of the selected function and function component type. The same function is also available via the **-** symbol in front of the function block type names.
Customize Columns...	Using this menu command you can: • define the columns, page 1235 to be displayed and • define the sequence, page 1235 for the columns shown.
Help on Type	Use this button to open the help for the selected function or function block types shown.
Data Properties	Use this menu command to call the Data Properties, page 1202 dialog box for the selected function and function block type.

Configuration of Columns of the Type Library Browser

Introduction

You can use the **Column Configuration** dialog box to:

- define the columns that you want to display,

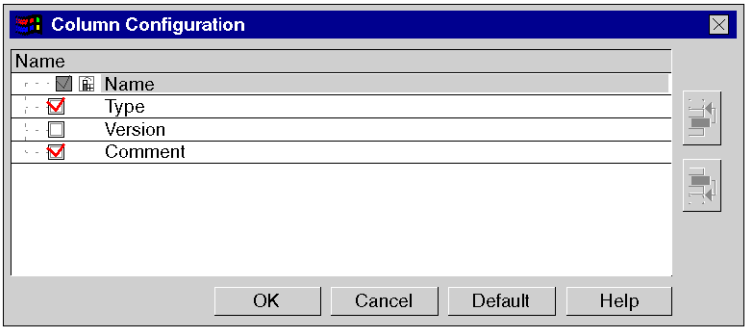
- define the sequence for the columns shown.

Call the dialog box:

You can call up the **Column configuration** dialog box from the shortcut menu in the Type Library browser, page 1233.

Call the dialog box

Representation:



Elements:

Element	Description
Name	This checkbox is to show the names of the functions and function blocks in the Type Library Browser and cannot be unchecked.
Type	When you activate this checkbox, the Type Library Browser and the types (EFB, FB etc.) for the functions and function blocks are shown.
Version	When you enable this checkbox, the Type Library Browser and the versions of the functions and function blocks types are shown.
Comment	When you enable this checkbox, the Type Library Browser and the comments for the functions and function blocks types are shown.
	Use this button to move the location of the selected attribute one place forward. The location of the attribute **Name** cannot be modified.
	Use this button to move the location of the selected attribute one place back. The location of the attribute **Name** cannot be modified.

Drag & Drop of FFBs in a program section

Procedure

To insert a function or function block (FFBs) using drag & drop, carry out the following steps:

Step	Action
1	Open a program section
2	Open the Types Library Browser per drag & drop. • the **Tools > Type Library Browser**menu command. or • press the key combination **Alt+3**.
3	Select the desired function/function block and hold down the left mouse button.
4	Drag the selected function/ function block types in the program section.
5	Release the mouse button. **Result:** • The FFB is inserted in the program section. • In the **Tools > Options** dialog box on the **Data and Languages** tab, if the **Automatically assign a variable to a new graphical object** checkbox is selected, then the Function input assistant, page 1199 is automatically opened for inputting the current parameters.

Search/Replace Tool

What's in This Chapter

Overview of the Search / Replace tool 1239
General Features for Search/Replace 1241
Searching for Data 1242
Replacing Data 1243
Search Toolbar 1244
Data Search Dialog 1246
Cross-References (Search Results) 1247
Cross-References (Replace) 1250
Tracing Variables / Addresses 1252

Subject of this Chapter

This chapter presents the **Search / Replace** tool and how to use its main functions:

- the **Search** function is used to look for data,
- the **Replace** function is used to replace data.

Overview of the Search / Replace tool

Introduction

The **Search / Replace** tool provides you with the following functions:

- the **Search** function,
- the **Replace** function.

Search Function

This function lets you **look for objects** such as:

- **instances** of elementary data types (EDT),
- **instances and types** of derived data (DDT / IODDT),

- **instances and types** of function block data (EFB / DFB),

  NOTE: In online mode, searching DFB instances can also be performed using the DFB instances animation button in the programming language editor toolbar. For more detailed information, refer to Searching / animating DFB Instances, page 1284.
- **instances and types** of data relating to Sequential Function Charts (SFC),
- **topological addresses** (`%M100`).

The search is carried out in the following areas of the project:

- sections that make up the application program,
- animation tables,
- operator screens,
- configuration.

Replace Function

This function lets you **replace objects** such as:

- **instances** of elementary data types (EDT),
- **instances** of derived data types (DDT / IODDT),
- **instances** of function block data (EFB / DFB),
- **topological addresses** (`%M100`).

The replacement objects must be declared, except for direct addressing data (`%M`, `%MW`, etc.) and their type must correspond to that of the objects being replaced.

The replacement can done on the search results and initiated at the result window.

Accessing the Data Search Dialog

To open the **Data Search** dialog

- click the **Data Search**  button in the toolbar or
- use the **Tools > Search / Replace** menu (**Alt+5**).

Predefined Search

To start a predefined search

- from the **Data Editor** or
- from a language editor window,

select the object whose references you want to see and use **Initialize Search** from the context menu.

Another way is to select a variable in a language editor window. Doing so, the list box of the search toolbar, page 1244 contains the variable name. Press **Enter** to start searching. You can also use 2 special buttons from the toolbar to search for reading/writing references, page 1245.

General Features for Search/Replace

Copy / Paste

You can use copy and paste in the following instances:

- copying a data item from any given editor in the programming software and pasting it into an entry field in the window,
- copying a data item from the window and pasting it into any given editor in the programming software,
- copying / pasting a data item within a window.

Drag-and-Drop

You can use drag-and-drop for search.

Drag a data item from any given editor in the programming software and drop it at the end of the list of displayed references; the corresponding tree is created.

Protection

The **Search** function is always available. Protection applies to:

- the references in the protected sections of the program, which are not displayed,
- the references in the protected sections of the DFBs, which are not displayed.

The **Replace** function complies with the protection applied to the different areas of the project:

- if the application field is in read-only, the **Replace** button is not available,

- if certain references are protected, they can not be selected for replacement.

Online Mode

The use and behavior of **Search / Replace** in online mode are identical to the use and behavior in offline mode.

Save Context

When you close the Search / Replace window, the following data is saved:

- the position and size of the window,
- the selection of search parameters,
- the 10 most recent entries in the drop-down menu,
- the current view (structural/functional and full/shortened/flat),
- the contents of the Control Expert **Search / Replace** output window.

Searching for Data

Searching via Data Search Dialog

To search data using the **Data Search** dialog, you must perform the following actions:

Step	Action
1	Open the **Data Search** dialog • by clicking the **Data Search** button  in the toolbar or • by using the **Tools > Search / Replace** menu (**Alt+5**) and pressing the **Data Search** button in the **Cross-References** window, page 1247. **Result**: The **Data Search Dialog** is displayed.
2	Specify the search options in the **Data Search Dialog**, page 1246.
3	Click the **Search** button. **Result**: The search results are displayed in the **Cross-References** window, page 1247.

Predefined Search

To start a predefined search

- from the **Data Editor** or
- from a language editor window,

select the object whose references you want to see and use **Initialize Search** from the context menu.

Another way is to select a variable in a language editor window. Doing so, the list box of the search toolbar, page 1244 contains the variable name. Press **Enter** to start searching. You can also use 2 special buttons from the toolbar to search for reading/writing references, page 1245.

Replacing Data

To access the data replace function, you must perform the following actions:

Step	Action
1	After searching for data, page 1242 the search results are displayed in the **Cross-References** dialog, page 1247.
2	In the **Cross-References** dialog click the  button to display the **Replace with** area of the dialog.
3	Enable the check boxes for those references of the search results which should be replaced.
4	• Enter the replacement data (only data compatible with the data being searched for is tolerated) • or click the **Down** arrow to select the replacement data out of a list providing the last 10 entries • or click the **...** button to open a **Data Selection** dialog to select the instance or data to be replaced (only the instances compatible with the instance being searched for are displayed)
5	Click the **Replace** button to start the replacement operation in the preselected **Scope**. Please refer to **Data Search** dialog, page 1246.

Search Toolbar

Introduction

The main toolbar of Control Expert provides a **Data Search** button with a list box beside it to select or to enter a search string.



Start Searching

To start searching the selected or entered search string press **Enter**.

The default search options are used.

The options can be changed by pressing the **Data Search** button left to the list box instead of pressing **Enter**.

The search results are displayed in the **Cross-References** dialog, page 1247 and a jump to the first occurrence of the searched data is done.

Data Search Dialog

To open the **Data Search** dialog, page 1246 click the binoculars symbol.

Searching for Reading/Writing References

To search for reading/writing references only, you can use

- the **Data Search** dialog, page 1246 or
- two special buttons from the toolbar (see below).

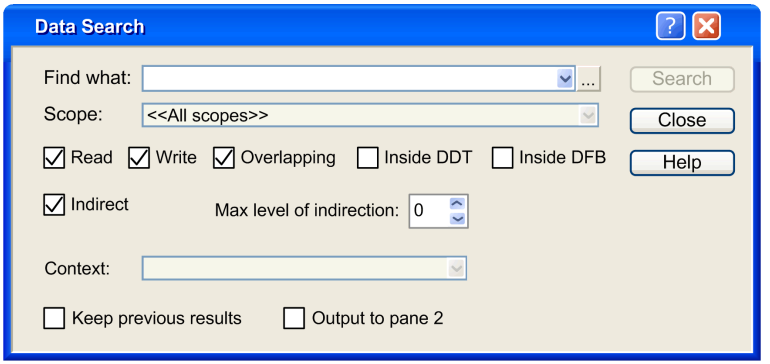
Button	Description
	Click this button to start searching for reading references only. The current search options, defined in the **Data Search** dialog, are used but the **Write** option is ignored.
	Click this button to start searching for writing references only. The current search options, defined in the **Data Search** dialog, are used but the **Read** option is ignored.

Using one of these 2 buttons initiates a direct jump to the first found occurrence of the reading/writing reference (for example, into the containing network).

If instead of a direct jump to the first found occurrence of the reading/writing reference the **Data Editor** is opened, either deactivate the Overlapping, page 1246 option or click the **Go to next reference** button, page 1248.

Data Search Dialog

Representation



Description

Elements of the **Search** tab

Element	Description
Find what	Enter the complete variable name you want to search for. Click the **Down** arrow to display a list providing the last 10 search strings. Click the **...** button to open a **Data Selection** dialog to select the object you want to search for.
Scope	Select the scope where you want to search. The different scopes are: • Global (application) • a specific Program Unit of the application • a specific DFB of the project • all scopes (application + all Program Units + all DFBs)
Read	Limits the search to reading references.
Write	Limits the search to writing references.
Overlapping	Activate this check box to search for all complete and partial references.
Inside DDT	Activate this check box to search inside DDT variables.
Inside DFB	Activate this check box to search inside DFB instances (except protected DFB types).

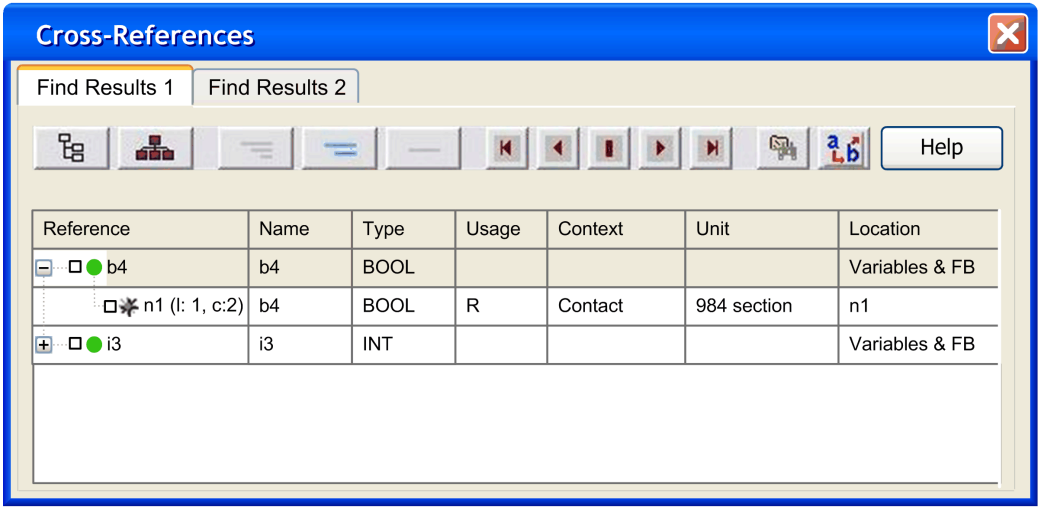
Element	Description
Indirect	Activate this check box to search for sub-elements and direct parents of structured variables and arrays.
Max level of indirection	Defines the maximum level of indirect elements (sub-elements/parents) to be displayed.
Context	Limits the search to a specific context. • **Contact**: variable attached to a contact • **Coil**: variable attached to a LD/LL984 coil • **SFC Object**: variable used with an SFC object • **FFB / Subroutine**: variable passed to an FFB • **ST / IL Statement**: variable used with an ST/IL statement The context is displayed as an optional column in the search results.
Keep previous results	Activate this check box to keep the previous search results. Otherwise, the search results window (**Cross-References**) is cleared before the new search results are displayed.
Output to pane 2	Deactivate this check box to display the search results in the **Find Result 1** tab of the search results window (**Cross-References**). Activate this check box to display the search results in the **Find Result 2** tab of the search results window (**Cross-References**).
Search button	Starts the search and opens search results window (**Cross-References**).
Close button	Closes the dialog.
Help button	Opens the Help for this dialog.

Cross-References (Search Results)

Introduction

The **Cross-References** search results dialog displays the results of a search.

Representation



Description

Elements of the **Cross-References** dialog

Element	Description
Find Result 1 tab **Find Result 2** tab	Depending on the respective check box activated in the **Data Search** dialog, page 1246, the search results are displayed in one of the 2 tabs.
	Displays the search results in a similar manner to the structural view in the **Project Browser**.
	Displays the search results in a similar manner to the functional view in the **Project Browser**.
	Displays the search results in **Full** form. • Data declaration • Path • Data usage
	Displays the search results in **Shortened** form. • Data declaration • Data usage
	Displays the search results in **Flat** form.

Element	Description
	• Data usage
	Selects the previous reference of the search results.
	Selects the next reference of the search results.
	Jumps directly to the currently selected found occurrence of the reference.
	Jumps directly to the previous found occurrence of the reference (for example into the containing network).
	Jumps directly to the next found occurrence of the reference.
	Opens the **Data Search** dialog.
	Displays/hides the **Replace with** area of the dialog, page 1250.
Reference	This column displays the name of the instance or type (DDT/IODDT/EFB/DFB) of the data searched for. The details displayed depend on the display settings described above (**Full**, **Shortened**, **Flat**).
Type	This column displays • type and reference of the data searched for, • programming language used in the sections, subroutines, events, etc. that contain references to the data searched for.
Usage	This column displays • the nature of the references to the data searched for (read only, write only, read / write), • the data search area.
Context	Specific context as defined in the Data Search Dialog, page 1246.
Unit	This column displays the unit of the **Location**.
Location	This column displays the location of the data searched for in a short form.

Sorting Order

When the search results are displayed in **Full** or **Shortened** form sorting can only be done by **Reference** or **Type**.

When the search results are displayed in **Flat** form sorting can also be done by reference related columns (**Usage**, **Context**, **Unit** and **Location**).

The search results are first sorted by their path (Program, Task, Segment, Section/Network).

The search results of sections/networks are sorted by a second key, depending on the kind of the section/network.

- References within IEC language sections (FBD, LD, SFC, ST, IL) are sorted by row and column.
- References within LL984 networks are sorted by column and row.

You can re-sort the search results by clicking on a specific column.

Re-sorting by multiple keys is not supported.

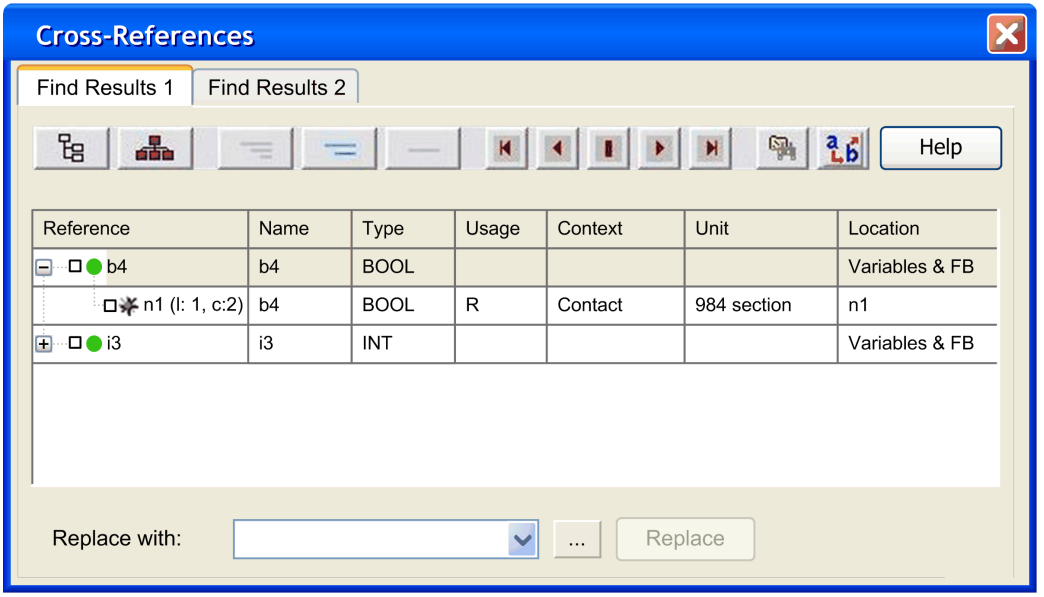
Cross-References (Replace)

Introduction

The **Cross-References** (Search Results) dialog, page 1247 displays the results of a search.

To display the **Replace with** area of the dialog use the  button.

Representation



Description

Elements of the **Replace with** area of the dialog

Element	Description
Replace with	To display this area of the dialog use the  button. • Enable the check boxes for those references of the search results which should be replaced. • Enter the replacement data (only data compatible with the data being searched for is tolerated), • or click the **Down** arrow to display a list providing the last 10 entries, • or click the **...** button to open a **Data Selection** dialog to select the instance or data to be replaced (only the instances compatible with the instance being searched for are displayed).
Replace button	Starts the replacement operation in the preselected **Scope**. Please refer to **Data Search** dialog, page 1246.

Tracing Variables / Addresses

Introduction

Using the tracing feature, you can locate the occurrences of a variable/address in an application.

Tracing

Step	Action
1	Select an object in a language editor window. **Result**: The address attached to the object is displayed in the edit field of the Search Toolbar, page 1244.
2	Start a search either • by pressing the **Enter** key or • by means of one of the 2 special buttons next to the edit field of the Search Toolbar (Searching for Reading/Writing References, page 1244). **Result**: • The search results are displayed in the **Cross-References** dialog, page 1247. • A direct jump is initiated to the first found occurrence of the reading/writing reference (for example into the containing network). • A new automatic **Search_"<Search String>"** bookmark is created (to be able to jump back later on). This bookmark is inserted into the bookmarks list like any other manually created bookmark. Automatically created bookmarks, page 620 are displayed in gray and can not be renamed.
3	Double-click a search result, displayed in the **Cross-References** dialog, to jump to the respective read/write location.
4	With the bookmarks feature you can mark one or more of the shown locations in the language editor. These bookmarks can be used later on to jump to from the bookmarks window.
5	From the bookmarks window you can also select the automatic **Start of Search** bookmark for going back to the original starting position of the search. **NOTE:** You can delete automatically generated bookmarks via the bookmarks window.

User's Function Blocks

What's in This Chapter

DFB Type 1254
DFB Instance 1273
How to Create Nested DFBs 1286

Overview

This chapter describes how to use the user's function blocks: DFB.

DFB Type

Purpose of this section

This section describes the types of DFB.

DFB Type

General information about DFBs

A **DFB** (Derived Function Block) is a user function block that has been customized to take the specific nature of your project into consideration. It can be stored in the user-defined library.

To use a DFB in your application, you must:

- Create a user's function block model called DFB type (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual),
- Use the available copies of the model called **DFB instances** (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) in your project.

Online Help for DFB Types (Help on Type)

It is possible to link an HTML help file to each DFB in the user-defined library.

- This file must have a name that is identical to the linked DFB and the extension must be **.htm*.
-
  - For Windows XP this file must be located in the directory:

    *C:\Documents and Settings\All Users\Application Data\Schneider Electric\Control Expert\CustomLibset/Vx.x/**Language***.
  - for Windows Vista this file must be located in the directory:

    *C:\ProgramData\Schneider Electric\Control Expert\CustomLibset/Vx.x/**Language***.

  Language is named ***ENG***, ***FRE***, ***GER***, ***ITA***, ***SPA*** or ***CHI*** according to the language desired.

Creating **Help on Type**

Step	Action
1	Create an HTML help file for your DFB (for example with any kind of HTML editor). **Note:** The HTML file name must be exactly the same as the name of your DFB.
2	Copy this file into the respective language folder (***ENG***, ***FRE*** ...).
3	Create an additional folder called ***HELP*** in the same directory as the language folders.
4	Copy all the files (for example graphics) you are referencing in your HTML file to the ***HELP*** folder.
5	Install your user-defined library in Control Expert. **Result:** All the files will be copied to the ***Libset*** directory and the HTML help file will be launched when you click the **Help on Type** button.

Accessing a DFB Type

To create a DFB type, you must use the program **Control Expert**.

DFB types are accessed via the **Derived FB Types** directory in the project browser.

NOTE: You can also access the DFB types definition tool via the directory **Variables & FB instances**.

Accessing DFB Types Via the Derived FB Types Directory

To access the DFB types via the **Derived FB Types** directory, do the following:

Step	Action
1	In the structural view of the project browser, right-click on the directory **Derived FB Types**. **Result**: A shortcut menu is opened.
2	Activate the **Open** command. **Result**: The data editor window is opened. It is positioned directly under the **DFB Types** register tab and shows the list of existing DFBs. A double-click on the first available line indicated by an arrow provides access to the creation of a new DFB type.

Creating a DFB Type

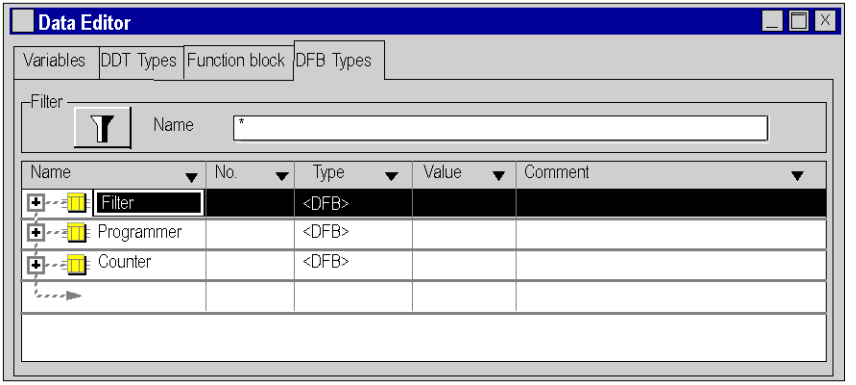
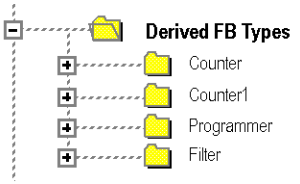
How to create a DFB type

To create a DFB type (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual), you must do the following:

- Enter a name on the empty line to create the empty structure of the DFB type,
- configure the DFB type,
- program the DFB Type (in the codes section).

Creating a DFB type

Do the following to create the empty structure of a DFB Type:

Step	Action
1	Open the **DFB types**, page 1254 register tab in the data editor. **Result**: The list of existing DFBs is displayed. 
2	Select the first empty **Name** cell (shown by an arrow) and enter the parameter or variable name and confirm with **Enter**. **Result**: The empty structure of the DFB type is now created. This new DFB added to the list of existing DFBs. It also appears in the directory tree **Derived FB Types**. 

Configuration of a DFB Type

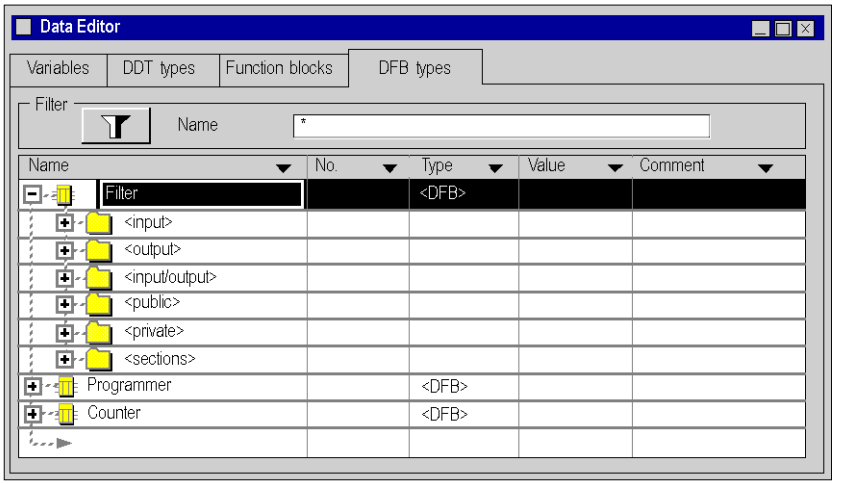
How to Configure a DFB Type

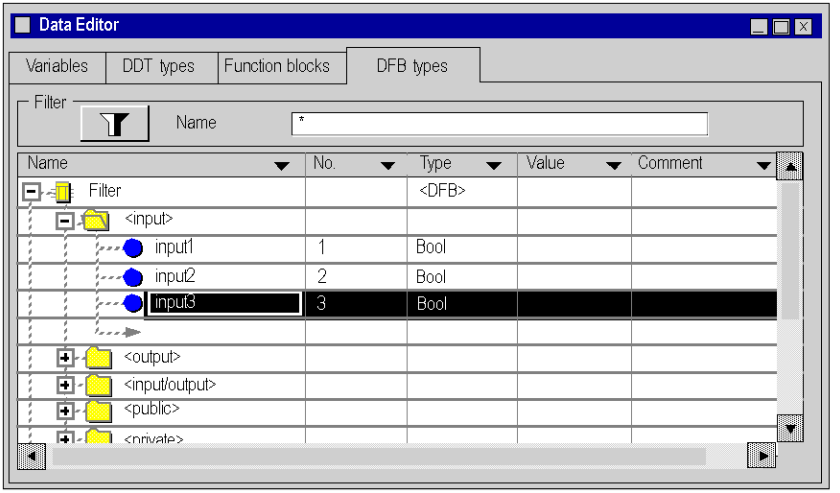
To configure a DFB type (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) you must:

- input the parameters of the DFB: input, output and input/output
- input the DFB variables: public or private
- Input the DFB description.

Input of the Parameters and Variables

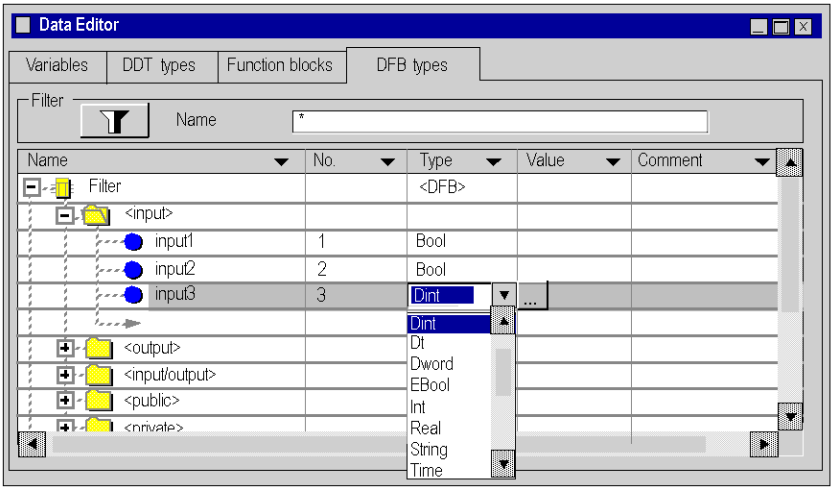
Carry out the following actions to input the parameters and the variables for a DFB:

Step	Actions
1	Go to the **DFB types**, page 1254 tab of the data editor. **Result**: the list of existing DFBs is displayed on the screen.
2	Click on the **+** sign positioned in front of the name of the DFB type to be configured. **Result**: the DFB type is expanded, which gives access to its various directories (**input**, **output**, etc.). 
3	Click on the **+** sign of the directory that you want to expand: **input**, **output**, **input/output**, **public**, **private**.

Step	Actions
4	Select the first empty **Name** cell (indicated by an arrow) and enter the name of the parameter or the variable, then confirm using **Enter**. **Result**: the data is created with a `BOOL` type by default (or with the last type selected). 
5	Repeat steps 4 and 5, for all data to be input.

Parameter Type Modification

Carry out the following actions to modify the type of DFB parameter:

Step	Actions
1	Double click on the **Type** cell associated with the desired parameter. **Result**: you have access to the list of available types, page 1192. 
2	Select a new type and confirm using **Enter**. **Result**: data is modified with the new type. **Note:**The MFB can be programmed in a DFB. To do this setup the type of an input by typing AXIS_REF. (this type name is not proposed in the list box but is a valid type for a DFB input). **Note:** The IODDT related to CANopen devices for Modicon M340 cannot be used as a DFB I/O parameter. During the analyse/build step of a project, the following message:"This IODDT cannot be used as a DFB parameter" advises the limitations to the user.

New Pin Number Allocation

Carry out the following actions to allocate a new pin number to a DFB parameter (when parameter is created, a default pin number is allocated):

Step	Actions
1	**Only** for **input**, **output** and **input/output** directories: if you want to allocate a new pin number to every element of data, double click on the relevant **No.** cell. **Result**: the arrow is replaced by a data entry field.
2	Enter the desired new pin number then **Enter** to validate. **Result**: the data is associated with the selected pin number.

Programming a DFB type

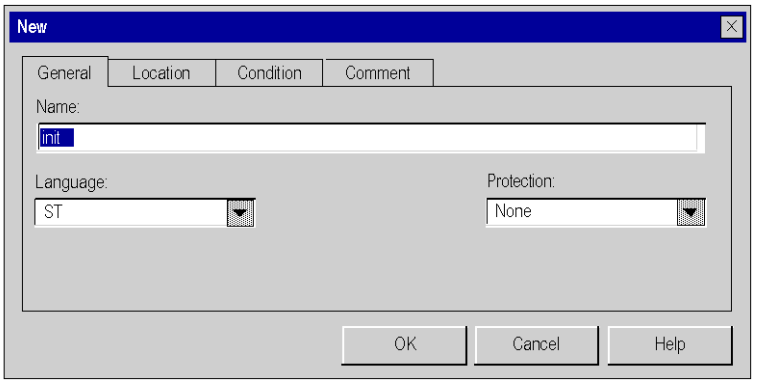
How to program a DFB type

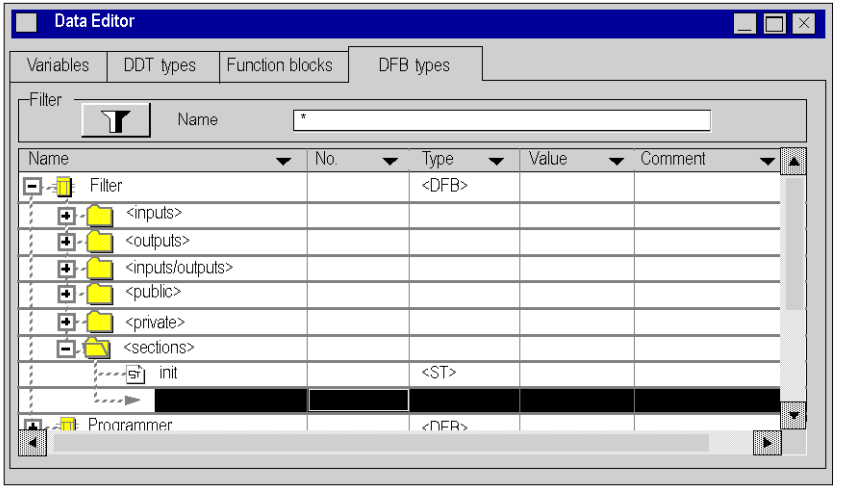
To program a DFB type (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) you must:

- create the DFB section(s),
- enter the program code for each section.

Creating a section

Carry out the following actions in order to create a DFB section:

Step	Action
1	Go to the **DFB types**, page 1254 tab of the data editor. **Result**: the list of existing DFBs is displayed on the screen.
2	Expand the DFB type that you would like to program.
3	Expand the **sections** directory by clicking on **+**.
4	Select the **Name** cell and enter the name of the section. Confirm with **Enter**. **Result**: a dialog box is displayed on the screen to allow you to create a new section. 
5	Select the programming language desired in the drop-down menu in the **Language** field.
6	Click **OK** to confirm the command. **Result**: the section appears in the **sections** directory.

Step	Action
	
7	If the **sections** directory already contains sections and you want to create a new one, expand the directory and select the first empty **Name** cell. **Result**: the dialog box used to enter section parameters is displayed on the screen.
8	Enter the section parameters and confirm with **OK**. **Result**: the new section appears in the **sections** directory.

Entering the program

Carry out the following actions in order to enter the program for the section:

Step	Action
1	Expand the **sections** directory of the DFB whose code you would like to enter. **Result**: the list of sections is displayed on the screen.
2	Right-click on the name of the section to program. **Result**: the contextual menu appears.
3	Activate the **Open** command. **Result**: the section programming screen is displayed on the screen, in the selected language.
4	Enter the section program, page 471.

Programming restrictions

All language instructions are allowed, except instructions that use input/output module variables (READ_STS, READ_PARAM, WRITE_CMD, etc..).

Connection to a label (JUMP) is only possible within the same section.

You may not use the following objects:

- input/output objects (%I, %Q, etc.),
- the application's global objects (%MW, %KW, ...), except system bits and words %S and %SW.

NOTE: When the **Create variable** option in the contextual menu is used from a DFB section, the variable created is a private variable.

NOTE: For PL7 Function blocks (Obsolete Library): it is recommended not to use the PL7_••• function blocks inside DFBs because you are not allowed to create new instances in online mode.

Deleting a DFB type element

Deleting an element

To delete an element, do the following:

Step	Action		
1	Open the **DFB types**, page 1254 tab of the data editor and expand the DFB type from which you want to remove an element.		
2	Expand the directory that contains the element you want to delete.		
3	**Either:** Select the element that you want to delete.		**Or:** Right-click on the element that you wish to delete. **Result**: A contextual menu appears.
4	Press **Delete**. **Result**: The item is removed from the directory.		Activate the command **Delete**. **Result**: The item is removed from the directory.

Archiving a DFB Type

Introduction

After having created a DFB type (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual), you can back it up in a user library, page 375 if the following conditions are met:

- You are using Control Expert software,
- The computer is in offline mode,
- You have the required access rights,
- The DFB Type to back up is compatible (the nested DFBs and DDTs are also compatible),
- The DFB type name doesn't already exist in the library (the name of the nested DFBs and DDTs must also not already exist in the library).

When a DFB Type is backed up in a library, it can be used if the following conditions are met:

- You are using Control Expert software,
- The computer is in offline mode,
- You have the required access rights.

Archiving a DFB Type

To archive a DFB type in a user library, you must do the following:

Step	Action
1	Open the **DFB types**, page 1254 register tab in the data editor.
2	Right-click on the DFB type that you wish to back up. **Result**: A shortcut menu is opened.
3	Activate the **Copy into library** command.
4	Select the library and the destination family. Click **OK** to confirm.

How to Reinforce Protection of a DFB Type

At a Glance

There are 4 levels of protection enforcement for a DFB type:

- **None**: The DFB type is not protected.
- **No modification of version info**: The DFB type is not protected except the DFB version number.
- **Read-Only**: The DFB type parameters directories (**inputs**, **outputs**, **inputs/outputs**, **public**, **private** and **sections**) are in read only format.
- **No Read & Write**: The DFB types parameters directories **private** and **sections** are not displayed. All the other the DFB type parameters directories (**inputs**, **outputs**, **inputs/outputs** and **public**) can be accessed from the data editor in read only format.

  NOTE: If a DFB uses a DDT, you can modify the DDT type even if the protection of DFB is enforced.

Password Mechanism

From lowest to highest level of security, the password mechanism of a DFB Type with the reinforced protection can be:

- **Not encrypted**
- **Encrypted**
- **Encrypted with hardened encryption** (except for level of protection enforcement: **No modification of version info**)

  NOTE: Choosing the hardened encryption mechanism contributes to prevention from any export of a DFB Type with the reinforced protection in an unencrypted format. This option increases protection against theft of intellectual property.

Protection Enforcement of a DFB Type

To reinforce the protection of a DFB Type, you must do the following:

Step	Action
1	Open the **DFB types** register tab in the data editor.
2	Select the DFB Type to reinforce the protection.
3	Select the menu command **Edit > Properties**. **Result**: The respective screen is displayed.

Step	Action
4	Select the protection enforcement level from the drop-down menu.
5	After selecting, enter your password in the **Entry** field (maximum 8 characters).
6	Confirm your password in the **Confirmation** field.
7	Check the **Password Encrypted** check box if an enhanced password protection is required. **NOTE:** A DFB Type with an encrypted password can not be edited with Unity Pro 4.0 or earlier. Unity Pro is the former name of Control Expert for version 13.1 or earlier. When **Password Encrypted** option is selected, you can reinforce the security by selecting the option **Hardened Encryption** check box. **NOTE:** A DFB Type with hardened encrypted password can not be edited with Control Expert 14.1 or earlier.
8	Click **OK** to confirm.

Modification of the protection enforcement level

To change a DFB Type protection enforcement level, you must do the following:

Step	Action
1	Open the **DFB types** register tab in the variables editor.
2	Select the DFB Type for which you need to change the level of protection enforcement.
3	Select the menu command **Edit > Properties**. **Result**: The respective screen is displayed.
4	Select the attribute **Protection** and click on the related box in the **Value** column. **Result**: A drop-down menu appears.
5	Select the new protection enforcement level from the drop-down menu (**None**, **No modification of version info**, **Read-Only**, **No Read & Write**).
6	After selecting, enter the current password if the protection of DFB was already reinforced.
7	Click **OK** to confirm.

Modify Password

To change a DFB type password, you must do the following:

Step	Action
1	Open the **DFB types** register tab in the variables editor.
2	Select the DFB type to change the password for.
3	Select the menu command **Edit > Properties**. **Result**: The respective screen is displayed.
4	Enable the attribute **Protection** by clicking on the **+** sign.
5	Select the attribute **Password** and click on the related box in the column **Value**. **Result**: The **Change password** dialog box is opened.
6	Enter the current password in the **Enter password** field.
7	Enter your new password in the **Entry** field (maximum 8 characters).
8	Confirm your new password in the **Confirmation** field.
9	Check the **Password Encrypted** check box if an enhanced password protection is required. **NOTE:** A DFB Type with a crypted password can not be edited with Unity Pro 4.0 or earlier. Unity Pro is the former name of Control Expert for version 13.1 or earlier. When **Password Encrypted** option is selected, you can reinforce the security by selecting the option **Hardened Encryption** check box. **NOTE:** A DFB Type with a hardened encryption password can not be edited with Control Expert 14.1 or earlier.
10	Click **OK** to confirm.

Deleting a DFB Type

Deleting a DFB type

To delete a DFB type, do the following:

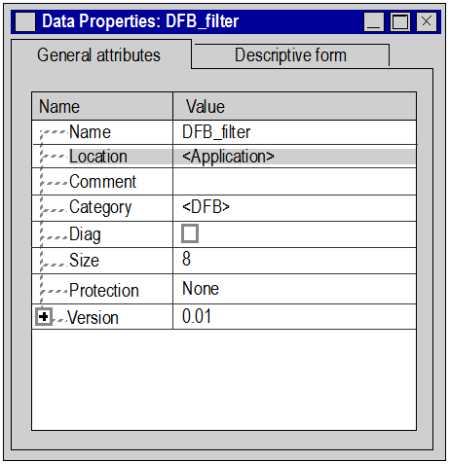
Step	Action		
1	Open the **DFB types**, page 1254 tab in the data editor. **Result**: The list of existing DFBs is displayed.		
2	**Either:** Select the DFB that you want to delete.		**Or:** Right-click on the DFB that you wish to delete. **Result**: A contextual menu appears.

Step	Action		
3	Press **Delete**. **Result**: The item is removed from the list.		Activate the command **Delete**. **Result**: The item is removed from the list.

How to access the attributes of a DFB type

Accessing the Attributes of a DFB Type

You should perform the following steps in order to access the attributes of a DFB type:

Step	Action
1	Go to the **DFB Types**, page 1254 tab of the data editor.
2	Right-click the DFB type whose attributes you would like to display. **Result**: The contextual menu appears.
3	Select **Properties**. **Result**: The corresponding dialog box appears on the screen. 
4	If you would like to associate an explanatory text with the definition of the DFB type (several lines of text with hyperlinks), select the **Descriptive form** tab. **Result**: The **Descriptive form** dialog box appears, page 1796.

Attribute List

The attributes of a DFB type are the following:

- **Name**
- **Location**
- **Comment**
- **Category**
- **Diag**
- **Size**
- **Protection**
  - **None**
  - **Read Only** (password protection)
  - **No Read & Write** (password protection)
  - **No modification of version info** (password protection)
- **Version**

  The version number is automatically incremented after a modification of the DFB type.
  - **Date**
  - **Template Signature**
  - **Code Signature**

  NOTE: The following attributes can be accessed in **write** mode depending on **Protection** item setting:
  - **Name**
  - **Comment**
  - **Diag**
  - **Protection**
  - **Version**

Authorized DFB Type Modifications

At a Glance

The modifications possible on a DFB type (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) are defined by the access rights authorized by your user profile.

The following table summarizes the authorized modifications when you have a default user profile:

Modification	Read only	Operation	Adjustment	Debugging	Program-ming
DFB type in offline mode	No	No	No	No	Yes
DFB type in online mode	No	No	No	No	Yes
DFB instance in offline mode	No	Yes	Yes	Yes	Yes

NOTE: In offline mode any kind of modification of a DFB Type is authorized on condition that a non-blocking protection level is used.

Authorized DFB type modifications

The following table shows the operations that you can perform with DFB types in online mode:

Action	Non-used DFB type	Used DFB type
Create a DFB type	Yes	No
Delete a DFB type	Yes	No
Modify the name	Yes	No
Change Protection Level	Yes	No
Modify the comment	Yes	Yes
Modify the description	Yes	Yes
Export a DFB type	No	No
Import a DFB type	No	No
Make a back up to a library of a DFB Type	No	No
Use a DFB type from a library	Yes	Yes

Authorized modifications on the input/output parameters

The following table shows the operations that you can perform for the input/output parameters of a DFB Type in online mode:

Action	Non-used DFB type	Used DFB type
Create a parameter	Yes	No
Delete a parameter	Yes	No
Move a parameter	Yes	No
Modify the name	Yes	No
Change type	Yes	No
Modify the comment	Yes	Yes
Change the initial value	Yes	Yes
Modify the other attributes	Yes	No

Authorized modifications on public variables

The following table shows the operations that you can perform on public variables of a DFB type in online mode:

Action	Non-used DFB type	Used DFB type
Create a variable	Yes	Yes
Delete a variable	Yes	No
Modify the name	Yes	No
Change type	Yes	No
Modify the comment	Yes	Yes
Change the initial value	Yes	Yes
Modify the other attributes	Yes	No

Authorized modifications on private variables

The following table shows the operations that you can perform on private variables of a DFB type in online mode:

Action	Non-used DFB type	Used DFB type
Create a variable	Yes	Yes
Delete a variable	Yes	No
Modify the name	Yes	No
Change type	Yes	No
Modify the comment	Yes	Yes
Change the initial value	Yes	Yes
Modify the other attributes	Yes	No

Authorized modifications of the sections

The following table shows the operations that you can perform on the sections of a DFB Type in online mode:

Action	Non-used DFB type	Used DFB type
Create a section	Yes	Yes
Delete a section	Yes	Yes
Move a section	Yes	Yes
Modify the name	Yes	No
Modify the comment	Yes	Yes
Modify the code	Yes	Yes

DFB Instance

Purpose of this section

This section describes DFB instances. DFB instances management is identical to the management of EFB instances, page 380.

DFB Instance

At a Glance

A DFB instances (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) is a copy of a validated DFB type (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) that can be used for programming.

You can create **multiple instances** of a same DFB type. In this case the inputs/outputs parameters and the variables (public and private) are duplicated. The DFB type code is not duplicated.

You can use the **same instance** of the DFB multiple times in your application. This operation is nevertheless **strongly discouraged**.

The DFB instance name must be different from the name of a Control Expert reserved word, a symbol, or from an elementary function (EF) name, or DFB type name.

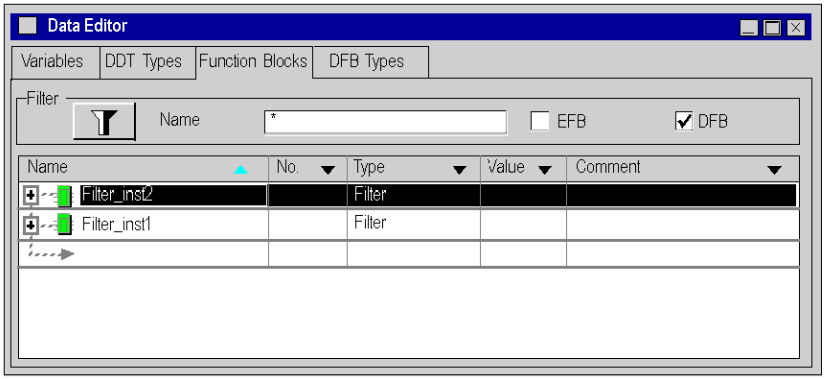
How to access to a DFB instance

To access a DFB instance, you must use the program Control Expert.

DFB instances are accessed via the **Variables & FB instances** directory in the project browser.

Accessing DFB instances via the Derived FB instances directory

To access the DFB instances via the **Derived FB instances** directory, you must do the following:

Step	Action
1	From the project browser structural view, expand the directory**Variables & FB instances**.
2	Double-click on the folder **Derived FB instances**. **Result**: The data editor window is opened. It is automatically placed under the **Function Blocks** register tab. The list of instances already in the system is displayed. 

Creating a DFB Instance

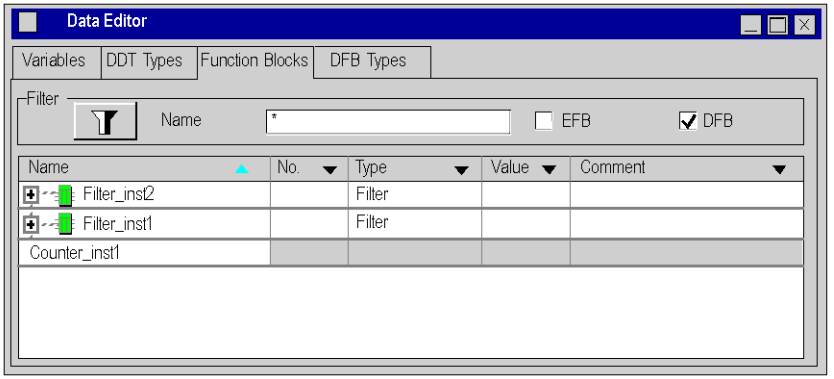
At a Glance

There are two possible ways in which you can create a DFB instance (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual):

- by inserting a new instance in the function block list,
- by inserting an instance in an editor.

Inserting a new DFB instance in the data editor

To create a new DFB instance, you must do the following:

Step	Action
1	Open the **Function Blocks**, page 1273 register tab in the data editor.
2	Select the first empty **Name** cell (shown by an arrow) and enter name of the new DFB instance and confirm with **Enter**. 
3	Then change the associated attributes, page 1279.

Inserting a new DFB instance in a language editor

To insert a new DFB instance in a language editor, you must do the following:

Step	Action
1	Open the language editor (the LD editor for example).
2	Click on an empty zone in the language editor. **Result**: A shortcut menu is opened.
3	Activate the **Data Selection** command.
4	In the drop-down list, select the DFB type of which you wish to insert an instance.
5	Confirm with **Enter**. **Result**: The mouse cursor is shown as a function block.
6	Click on the location where you wish to insert the DFB instance. **Result**: A new DFB instance is created with a default name. It is inserted in the language editor and created in the data editor at the same time.

Creating several DFB Instances

At a Glance

Several DFB instances can be created at once using the import function.

Creation of several DFB instances by importing a source file

Several DFB instances (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) can be created by importing, page 1686 a variables exchange file (.xsy or .txt format). If you have chosen to export the variables with the associated DFB types, the latter are imported.

Deleting a DFB instance

Deleting a DFB instance

To delete a DFB instance, you must do the following:

Step	Action		
1	Open the **Function Blocks**, page 1273 tab in the data editor.		
2	**Either:** Select the DFB instance that you want to delete.		**Or:** Right-click on the DFB instance that you wish to delete. **Result**: A contextual menu appears.
3	Press **Delete**. **Result**: The item is removed from the list.		Activate the command **Delete**. **Result**: The item is removed from the list.

Operations on DFB Instances

At a Glance

For each DFB instance (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual), the **Function Blocks** register tab in the data editor provides

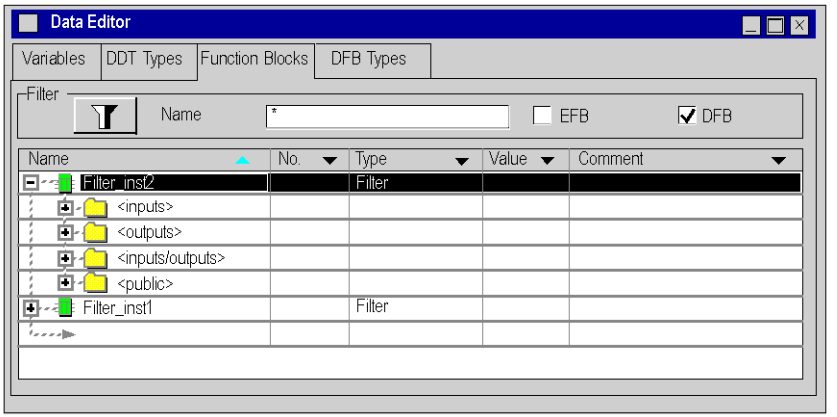
one line of information by default: Instance name, DFB type, comment, etc. From this screen you can also:

- Apply a filter on the list of instances, page 433,
- View the elements of a DFB instance,
- View the attributes of a DFB instance,
- Sort the DFB instances according to their attributes.

NOTE: The DFB instance structure cannot be modified using the **Function Blocks** register tab (name and type of the elements). However, the attributes of the instance itself can be modified (name, type and comments of the DFB instance).

View the elements of a DFB instance

To view the elements of a DFB instance, you must do the following:

Step	Action
1	Open the **Function Blocks**, page 1273 register tab in the data editor.
2	Open the DFB instance of which you want to view the elements. 

View the elements of a DFB instance

To view the attributes of a DFB instance, you must do the following:

Step	Action		
1	**Either:** Open the **Function Blocks**, page 1273 register tab in the data editor.		**Either:** Open the DFB instance to gain access to its elements.
2	Right-click on the DFB instance whose attributes you wish to view. **Result**: A shortcut menu is opened.		Right-click on the element whose attributes you wish to view. **Result**: A shortcut menu is opened.
3	Activate the **Properties** command. **Result**: The related dialog box is opened.		

Attributes list

The specific DFB instance attributes are:

- **Instance name**,
- **Comments**,
- **Type**,
- **Category**,
- **Size**,
- **Diag**,
- **Used (frequency of use in program)**,
- **Customize**.

NOTE: All the DFB instance attributes are available in **read/write** mode except the **Category** and the **Used** attributes of instances, which are available **in read only**.

Sorting DFB instances

To change the classification criteria, do the following:

Step	Action
1	Open the **Function Blocks**, page 1273 register tab in the data editor.
2	Click on the title of the column that you want for the first criteria. **Result**: The DFB instance classification criteria become: • Firstly, a classification by alphabetical order in the column on which you have clicked, • then, a classification by alphabetical order by name. **Note**: The DFB instances are classified by name alphabetically as default.

Modifying the Attributes of a DFB Instance

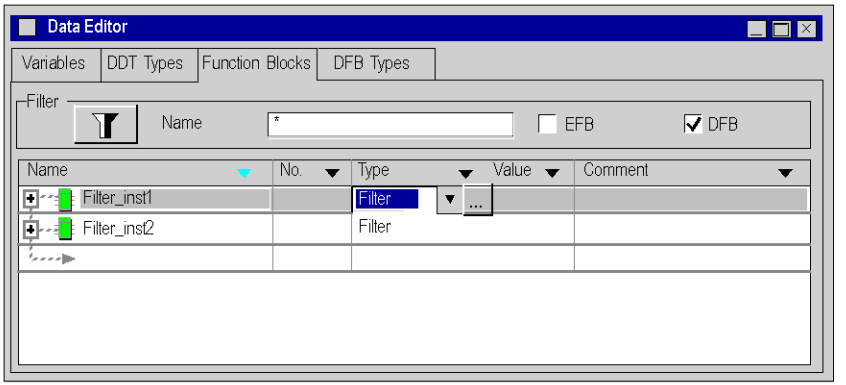
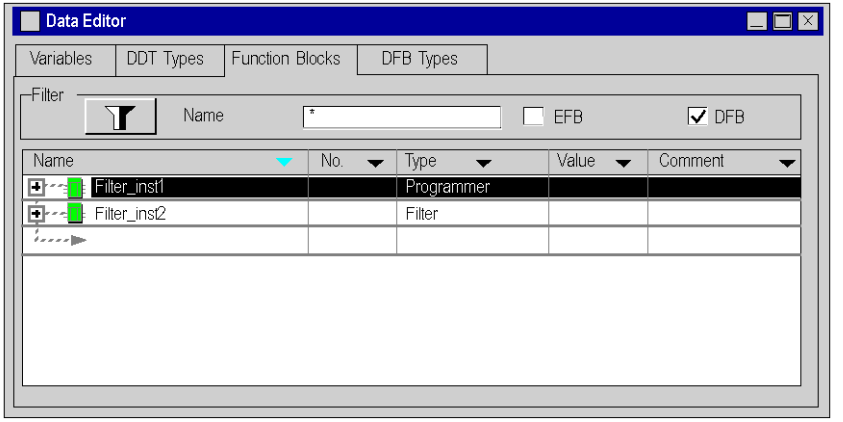
At a Glance

There are 2 different ways to modify the attributes of a DFB instance (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual):

- From the **Function Blocks** register tab, for the attributes that are directly accessible,
- or via the **Data Properties** dialog box for all attributes.

Modifying an attribute using the Function Blocks register tab

To modify the attributes of a DFB instance, you must do the following:

Step	Action
1	Open the **Function Blocks**, page 1273 register tab in the data editor.
2	Double-click on the cell that you wish to change the content of (you can also highlight this cell and press **Enter**). **Result**: A selection list or an entry field is displayed. 
3	Modify the attribute (instance name, instance type, comment) and confirm with **Enter**. **Result**: The DFB instance is displayed with its new attribute. 

Modifying an attribute via the Data Properties dialog box

To modify the attributes of a DFB instance using the **Data Properties** dialog box, you must do the following:

Step	Action
1	Open the **Function Blocks**, page 1273 register tab in the data editor.
2	Right-click on the DFB instance whose attributes you wish to view. **Result**: A shortcut menu is opened.
3	Activate the **Properties** command. **Result**: The related dialog box is opened.
4	In the **Value** column, double-click on the cell of the attribute to modify (you can also highlight this cell and press **Enter**).
5	Modify the attribute (instance name, instance type, comment) and confirm with **Enter**. **Result**: The new attribute is assigned to the DFB instance.

Simultaneous modification of an attribute of several DFB instances

To simultaneously modify an attribute of several DFB instances, you must do the following:

Step	Action
1	Open the **Function Blocks**, page 1273 register tab in the data editor.
2	Select the DFB instances of which you want to change an attribute (adjacent or non-adjacent multi-selection).
3	Right-click on one of the instances that you just selected. **Result**: A shortcut menu is opened.
4	Activate the **Properties** command. **Result**: The related dialog box is opened.
5	Modify the attribute value and confirm with **Enter**. **Result**: The new attribute is assigned to the DFB instance.

Limitations

The following limitations apply to multiple selections:

- All the selected instances must have the attribute and it must be modifiable,
- The new attribute value must be valid for all the selected instances,
- The attribute name cannot be changed (a DFB instance name is unique).

NOTE: In a multiple selection, the selected elements must at the same level of the hierarchy or must be compatible.

Modifying the attributes of the elements of multiple DFB instances

Modification of one or several (multiple selection) attributes of elements of one or several DFB instances. To do this, you must:

Step	Action
1	Open the **Function Blocks**, page 1273 register tab in the data editor.
2	Open the DFB instance(s) to modify.
3	Select the elements for which you want to modify the attributes (by pressing **Shift** if the multiple selections are adjacent or **Ctrl** if the multiple selections are non-adjacent).
4	Right-click on one of the elements that you just selected. **Result**: A shortcut menu is opened.
5	Activate the **Properties** command. **Result**: The related dialog box is opened.
6	Modify the attribute value and confirm with **Enter**. **Result**: The new attribute is assigned to the selected elements.

Authorized DFB Instance Modifications

At a Glance

The modifications possible on a DFB instance (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) are defined by the access rights authorized by your user profile, page 1269.

Authorized DFB instance modifications

The following table shows the operations that can be carried out on the DFB instances in offline mode:

Situation	Possible Actions
Actions when there is no DFB instance selected	Create a DFB instance,
	Paste a DFB instance.
Actions when a DFB instance is selected	Delete a DFB instance,
	Change the name of a DFB instance,
	Change the type of a DFB instance,
	Change the comment for a DFB instance,
	Duplicate a DFB instance.
Actions when several DFB instances are selected	Delete DFB instances,
	Change the type of the DFB instances,
	Change the comment for the DFB instances,
	Duplicate DFB instances.
Actions on all DFB instances	Export all DFB instances,
	Print locally all DFB instances.

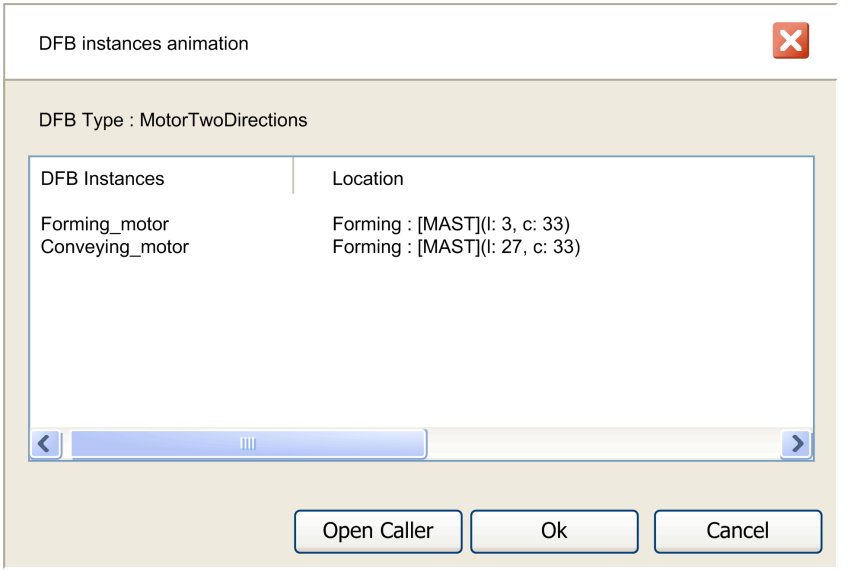
Searching / Animating DFB Instances

Overview

In online mode, you can display the location of instances using a DFB type, select one of the instances, and animate it.

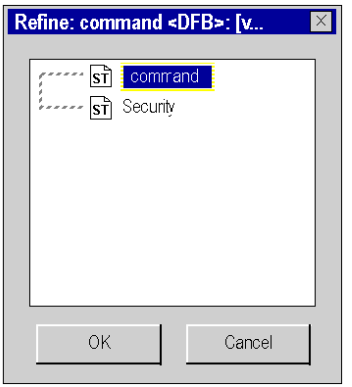
Searching a DFB Instance

To search DFB instances in a program perform the following operations:

Step	Action
1	Select the corresponding DFB type and open one of the code sections.
2	Click the  button in the programming language editor toolbar.
3	The **DFB instances animation** dialog box opens: 
4	Select the DFB instance then click the **Open Caller** button.
5	This opens the DFB instance in your application program.

Animating a DFB Instance

To animate a section of a DFB Instance perform the following operations:

Step	Action
1	Select the corresponding DFB type and open one of the code sections.
2	Click the  button in the programming language editor toolbar.
3	The **DFB instances animation** dialog box opens.
4	Double-click the DFB instance that you want to animate. Or, Select the DFB instance that you want to animate and click **Ok**. A window showing the DFB sections opens:  In this box, choose the section code of the DFB that you want to animate and confirm with **OK**.

How to Create Nested DFBs

Aim of this Section

This subchapter describes the procedure for creating nested DFBs.

How to Create Nested DFBs

At a Glance

When creating nested DFBs (DFBs with DFB instances as private variables), you must not exceed 15 nesting levels (including DDT variables).

You must also respect certain chronological rules. The following table describes the procedure.

Procedure

The steps are as follows.

Step	Action
1	Create the DFB type of the last level (**n**)
2	Create the DFB type of the level **n-1**
3	For this DFB type, create a private variable with a type matching the DFB type of level **n**
4	Create the DFB type of the level **n-2**
5	For this DFB type, create a private variable with a type matching the DFB type of level **n-1**
6	Repeat these actions **without exceeding 15 levels**
7	Create a **first** level DFB type instance, which you will use in your program.

Project Configuration

What's in This Part

Introduction to the configuration 1288
Bus Editors 1301
Modules editor 1348
General functionalities of the editor 1385

Subject of this Part

This part describes how to configure Premium/Atrium, Quantum and Modicon M340 PLC stations.

Its purpose is to guide the user in the configuration:

- of the PLC station,
- of the modules used in the station.

For Modicon M580 PLC stations, refer to chapter *Configuring the CPU in Control Expert* (see Modicon M580, Hardware, Reference Manual).

For Momentum, refer to chapter *Configuring the 171 CBU 78090, 171 CBU 98091 and 171 CBU 98091 Processors in Control Expert* (see Momentum for EcoStruxure™ Control Expert, 171 CBU 78090, 171 CBU 98090, 171 CBU 98091 Processors, User Guide).

Introduction to the configuration

What's in This Chapter

Proposed Services 1288
Services offered in offline mode 1293
Services Offered In Online Mode 1294
Access to project configuration 1295

Aim of this section

This section gives a very general introduction to the functionalities offered by the configuration editor and describes how to access the default configuration.

Proposed Services

Introduction

The configuration tool is used to:

- **create\modify\save** (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) the elements involved in the configuration of the controller,
- **set up** the application-specific modules of the controller,
- **diagnose** the modules configured in controller,
- **assess** the current discharged from the voltages supplied by the power supply module declared in the configuration,
- **control** the number of application-specific channels configured in relation to the capacities of the controller module declared in the configuration.
- **assess** the controller memory (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) usage.

The configuration may be performed before or after the programming of the project; this has the advantage of being able to create generic projects without having to be concerned with the configuration in the initial stage.

The various services are provided through 2 editors which are:

- the bus editor, which allows you to select and position the modules or devices on the bus,

- the input/output module editor, which allows you to set up the modules or devices present on the controller bus(es).

  NOTE: When you configure the different elements of your project (specific application, modules, etc.) you may ensure that there are no conflicts between data areas (overlap) as this may result in downgraded application functionality.

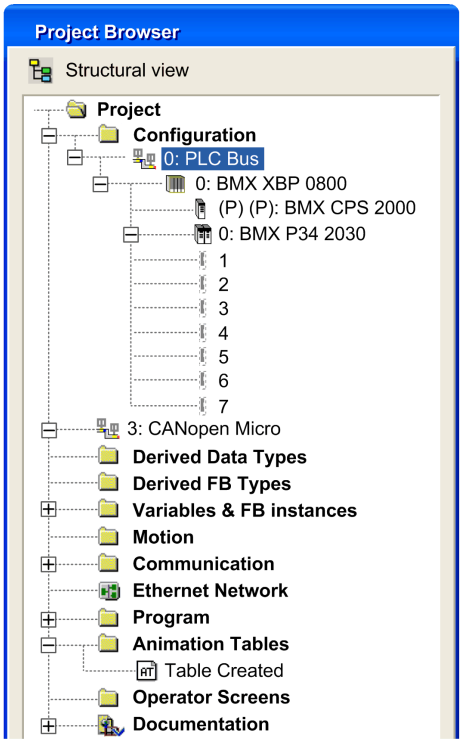
⚠WARNING
UNINTENDED EQUIPMENT OPERATION Before performing a build, ensure that there is no overlapping of data between specific applications within the same project and that the project operates correctly. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

At a Glance

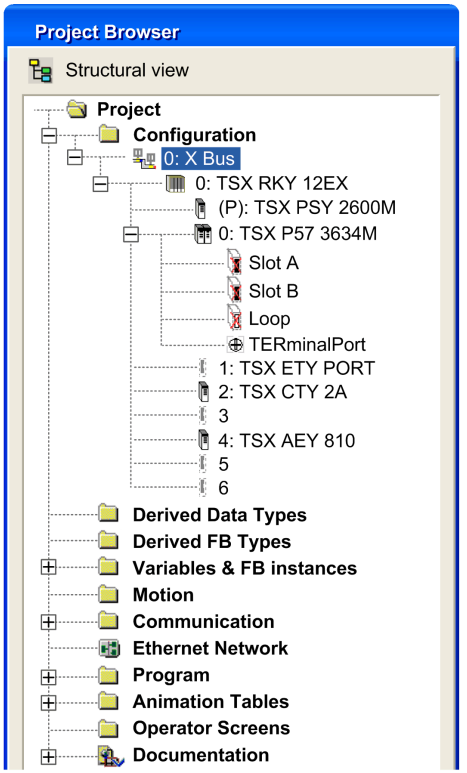
From the project browser, you can view the hardware configuration of the controller by opening the **Configuration** directory.

The hardware configuration of the controller is displayed in a directory tree.

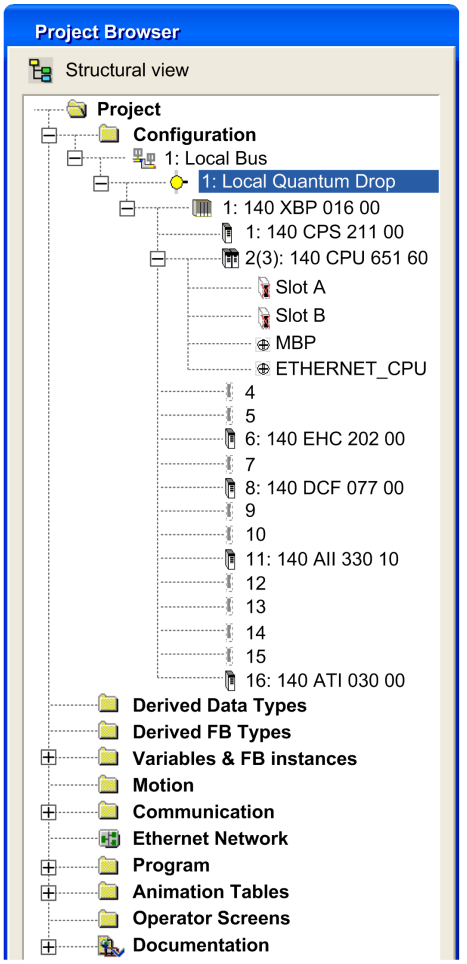
Configuration example for a Modicon M340 controller:



Configuration example for a Premium controller:



Configuration example for a Quantum controller:



NOTE: you may note that each address (slot in the rack) is not necessarily occupied by an input/output module, and that some modules may occupy 2 slots.

Services offered in offline mode

Introduction

These functional options are preset by the bus editor and the input/output module editor, when the PLC is not connected to the programming terminal.

From the bus editor

The available services are:

- selection/replacement of the rack(s),
- selection/replacement of the power supply module,
- selection/replacement/set-up of the processor,
- selection/positioning/replacement of an application-specific module or a device,
- selection/replacement of a PCMCIA card in a module or in the processor,
- the figures for:
  - the consumption of power provided by the power supply module,
  - the number of application-specific channels configured,
  - the memory usage of the PLC.

From the input/output modules editor

The available services are:

- the set-up of the input/output modules,
- the set-up of certain PCMCIA cards,
- the set-up of the application-specific functions.

Services Offered In Online Mode

Introduction

These are configurable functional options when the PLC is connected to the programming terminal.

Certain services are available or not according to whether the type of PLC connected is:

- Modicon M340,
- Premium/Atrium,
- Quantum.

Modicon M340 PLCs

Services offered by the configuration editor:

Service	PLC in RUN mode	PLC in STOP mode
Adding/moving a module	No	No
Modification of configuration parameters	Yes	Yes
Modification of adjustment parameters	Yes	Yes
Display of errors	Yes	Yes
Display of status information	Yes	Yes

Premium\Atrium PLCs

Services offered by the configuration editor:

Service	PLC in RUN mode	PLC in STOP mode
Adding/moving a module	No	No
Modification of configuration parameters	Yes	Yes
Modification of adjustment parameters	Yes	Yes

Service	PLC in RUN mode	PLC in STOP mode
Display of errors	Yes	Yes
Display of status information	Yes	Yes

Quantum PLCs

All Quantum PLCs allow online modification. Services offered by the configuration are:

Service	PLC in RUN mode	PLC in STOP mode
Automatic detection of the PLC configuration	Yes	Yes
Adding/removing a module	Yes	Yes
Modification of configuration parameters	Yes	Yes
Modification of adjustment parameters	Yes	Yes
Display of errors	Yes	Yes
Display of status information	Yes	Yes

NOTE: All modifications made in the bus editor are authorized in online mode. But in order for these to be taken into account, you must generate and transfer the project to the PLC.

NOTE: A feature CCOTF modification (see Quantum using EcoStruxure™ Control Expert, Change Configuration On The Fly, User Guide) has been developped for Quantum and allows I/O configuration changes when the PLC is in RUN mode.

Access to project configuration

Introduction

When a project is created, a default configuration is created automatically according to the choices imposed by the programming software.

These choices concern:

- the PLC family,
- the type of processor used in the station.

These choices determine the power of the configuration, and the available functionalities (number of inputs/outputs, field buses, languages used, types of data, etc.).

NOTE: The choice of PLC family is irreversible, only the type of processor may be changed.

Procedure

Carry out the following actions:

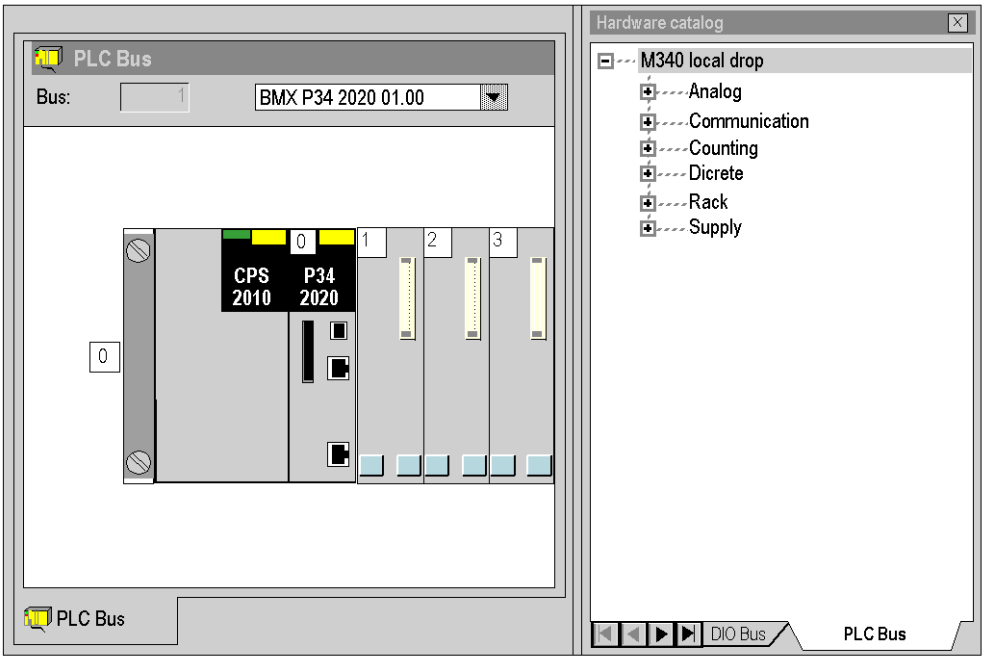
Step	Action
1	From the project browser, open the **Configuration** directory.
2	**For a Modicon M340 station:** From the **PLC Bus** directory or from the directory representing the station, via the contextual menu select the **Open** command; the default configuration is displayed on the screen. **For a Premium station:** From the **X-Bus** directory or from the directory representing the station, via the contextual menu select the **Open** command; the default configuration is displayed on the screen. **For a Quantum station:** From the directory representing the station, via the contextual menu select the **Open** command; the default configuration is displayed on the screen.
3	In the configuration window, via the contextual menu select the **Zoom in** command to determine the size of the station. Click **View > Zoom > Zoom to Fit** to optimize the size of the station in relation to the size of the window.

At this stage the configured physical elements constituting the station are:

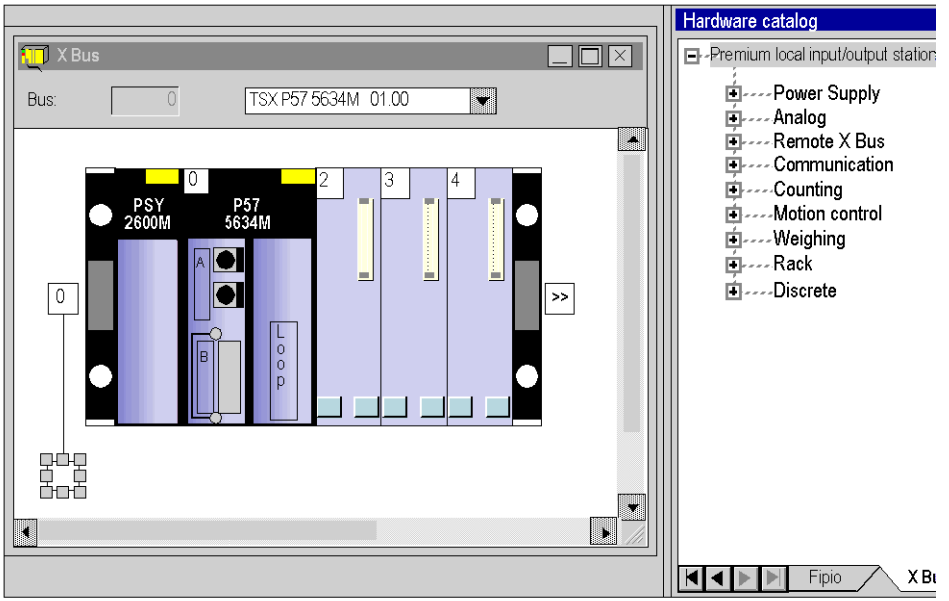
- the rack,
- the power supply module (except for a Quantum station),
- the processor.

Configuration

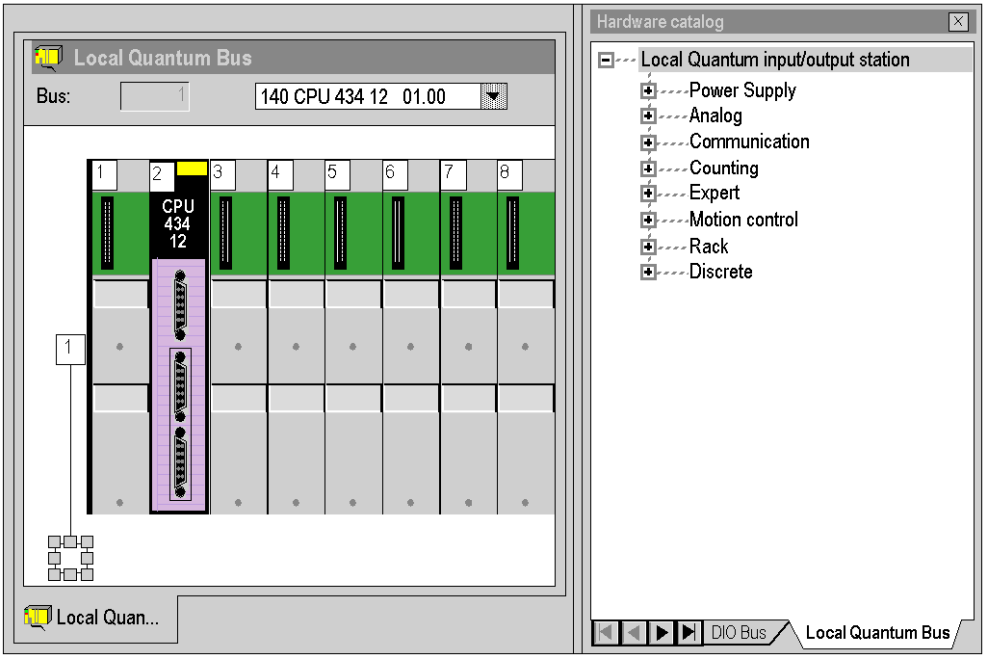
For a Modicon M340 station:



For a Premium station:



For a Quantum station:



Bus Editors

What's in This Chapter

Rack configuration on local bus 1301
Configuration of the power supply modules 1308
Configuration of the processor 1310
Configuration of the modules in the PLC station 1314
Configuration of devices on the field bus 1319
Consumption management 1337
Bus editors in online mode 1344

Aim of this section

This chapter guides the user in the creation of the physical configuration of the PLC station.

Rack configuration on local bus

Aim of this sub-section

This sub-section guides the user in the installation of the rack/s constituting the station.

Organization of the Racks

Introduction

A default rack is selected when the project is created. It has the following address:

- **0** for a PLC from the Premium/Atrium or Modicon M340 family,
- **1** for a PLC from the Quantum family.

This rack contains the type of processor selected when the project is created. This processor can be replaced by a compatible processor.

Processors belonging to the Modicon M340 family.

Type of processors	Number of racks managed
For all M340 PLC Version 01.00.	1 rack
For BMX P34 1000 Version 02.00	2 racks
For BMX P34 20X0	4 racks

Processors belonging to the Premium/Atrium family

Type of processors	Numbers of racks managed
TSX 57 0244	1 rack
TSX 57 1x4	Up to 4 racks
TSX P57 204 TSX PCI 57 204 TSX P57 254 TSX P57 2634 TSX P57 304 TSX P57 354 TSX P57 3634 TSX P57 454 / TSX PCI 57 354 TSX P57 4634 TSX P57 554 TSX P57 5634 TSX P57 6634	Up to 16 racks

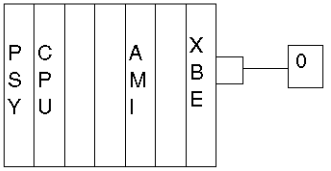
Processors belonging to the Quantum family

Type of processors	Numbers of racks managed
140 CPU 311-10	Does not depend on the type of processor
140 CPU 434-12A/U	
140 CPU 534-14A/U	
140 CPU 651-50\60\60S	
140 CPU 652-60	
140 CPU 658-60	
140 CPU 671-60\60S	
140 CPU 672-60\61	

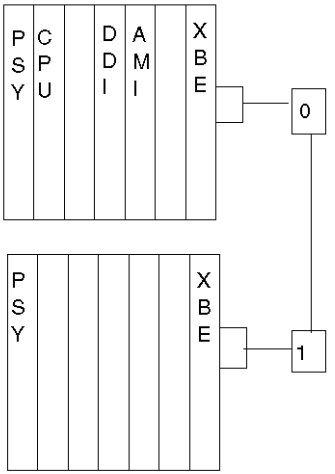
Type of processors	Numbers of racks managed
140 CPU 678-61	

Organization of a Modicon M340 Station on PLC Bus

Station composed of a single rack:

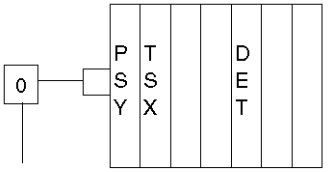


Station composed of several racks:

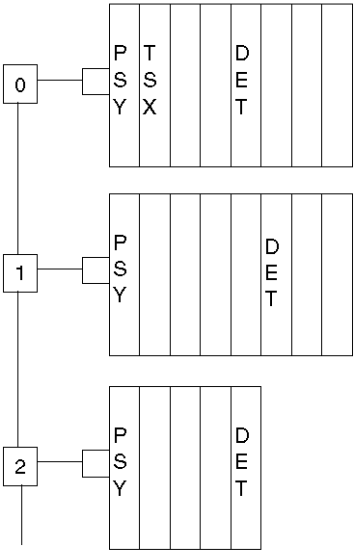


Organization of The Premium/Atrium Station on Bus X

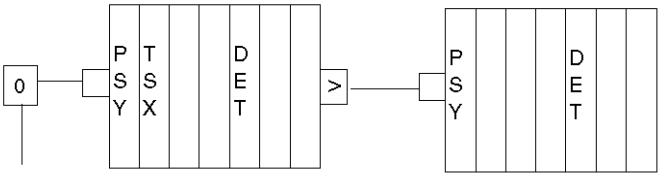
Station composed of a single rack (standard rack):



Station composed of several racks (extendable racks) with different addresses:

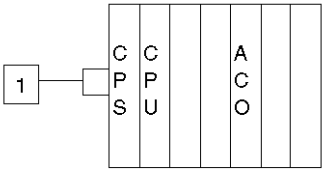


Station composed of several racks (extensible racks) with the same address:

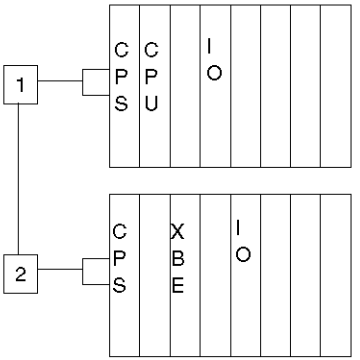


Organization of a Quantum Station on Local Bus

Station composed of several racks:



Station composed of several racks (extendable racks) with different addresses:



How to configure the racks

Introduction

You may manipulate the racks of a PLC station:

- either by using the functions provided by the bus editor,
- or from the browser provided by the bus editor.

Functions offered by the bus editor

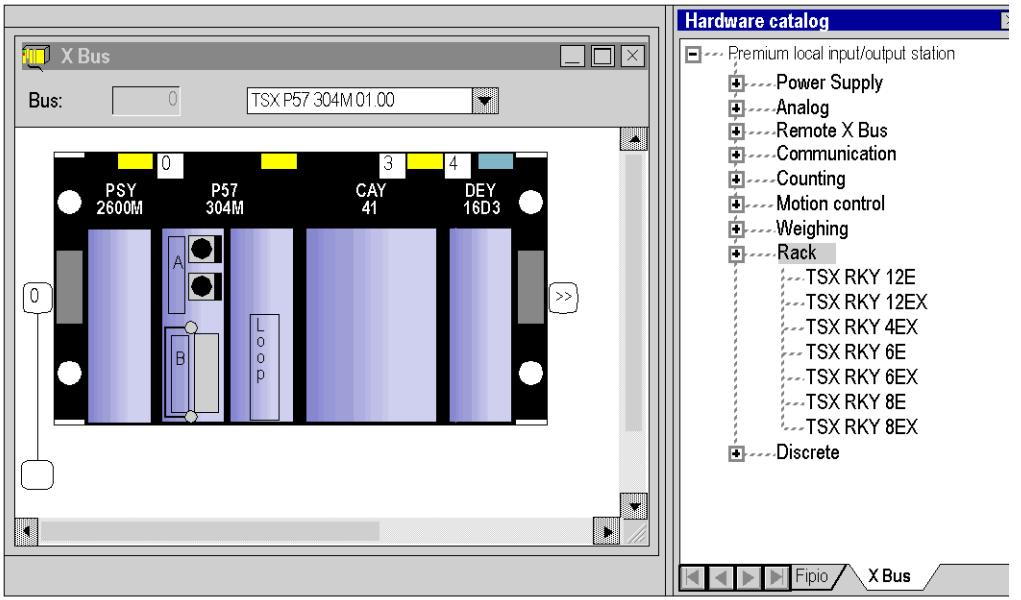
From the bus editor the following functionalites are offered:

If you want to	then	and
select a rack	select the address of the rack  eight handles appear around the selected rack.	
Cut/Paste a rack	select the rack and, from the contextual menu, click **Cut**	select the destination address of the rack and, using the contextual menu, click **Paste**.
Copy/Paste a rack	select the rack and, using the contextual menu, click **Copy**	select the destination address of the rack and, using the contextual menu, click **Paste**.
add a rack	select an empty address  or the extension symbol 	using the contextual menu, click **New Device**,
replace a rack	select the rack and, using the contextual menu, click **Replace Rack**,	select the required rack from the list offered.
delete a rack	select the rack	using the contextual menu, click **Delete Rack**,
empty a rack	select the rack	using the contextual menu, click **Clear Rack**,

Bus editor browser

The browser only allows for the addition of one rack in the PLC station.

Example for a Premium station:



Carry out the following actions:

Step	Action
1	From the browser, select the **Rack** directory and open it by clicking on **+**
2	select the required rack and, holding down right mouse button, move it to the required address point; a dialog box is displayed.
3	Specify in the dialog box the required address and confirm with **OK**.

Configuration of the power supply modules

Aim of this sub-section

This sub-section guides the user in the management of the power supply module(s) of the PLC station.

How to configure the power supply modules

Introduction

When creating an application two boxes are produced according to whether you select a **Premium/Atrium** station, a **Quantum** station or a **Modicon M340** station:

- in a Modicon M340 or Premium station a power supply module is configured by default,
- in a Quantum station, there is no power supply module configured by default.

Rules for a Modicon M340 station

The power supply module must occupy the left-most position of the rack. This position does not have an address.

There is only one power supply module per rack.

Rules for a Premium/Atrium station

The power supply module must occupy the left-most position of the rack. This position does not have an address.

A double format power supply module also occupies the address position 0 (usually occupied by the processor module), in this case the processor module must be configured at the address position 1.

There is only one power supply module per rack.

Rules for a Quantum station

The power supply module can occupy any rack position. It has an address.

The power supply modules are single format.

Several power supply modules may be configured in a rack.

Functions offered by the bus editor

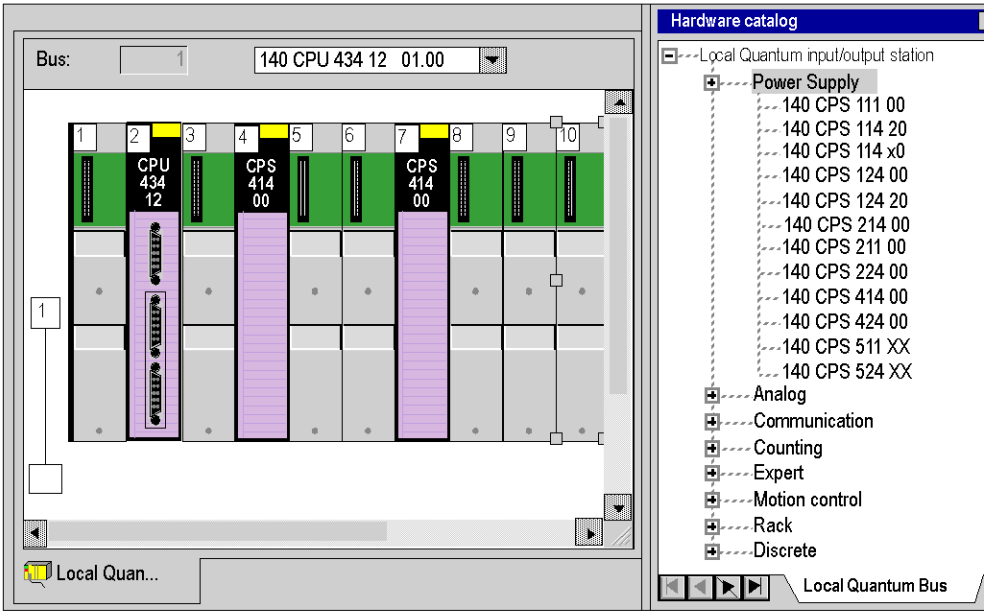
From the bus editor the following functionalites are offered:

If you want to	**then**	**and**
Select a module	click on it. Eight handles appear around the module.	
Copy/Paste the module	select the module and, using the contextual menu, click **Copy**	select the target position and, using the contextual menu, click **Paste**.
Cut/Paste the module	select the module and, using the contextual menu, click **Cut**	select the target position and, using the contextual menu, click **Paste**.
Add a module	select the position in the rack required and, using the contextual menu, click **New Device**	select from the proposed list the module required.
Move a module	select the module	holding down the left mouse button, move it to the required position.
Delete a module	select the module	using the contextual menu, click **Delete Module**.

Bus editor browser

The browser is used to put or to add (for Quantum) a power supply module in the station.

Example of a Quantum station containing 2 power supply modules:



Carry out the following actions:

Step	Action
1	In the browser, select the directory **Supply**, and display it by clicking on **+**.
2	Select the power supply module required and, holding down the left mouse button, move it to the required position.

Configuration of the processor

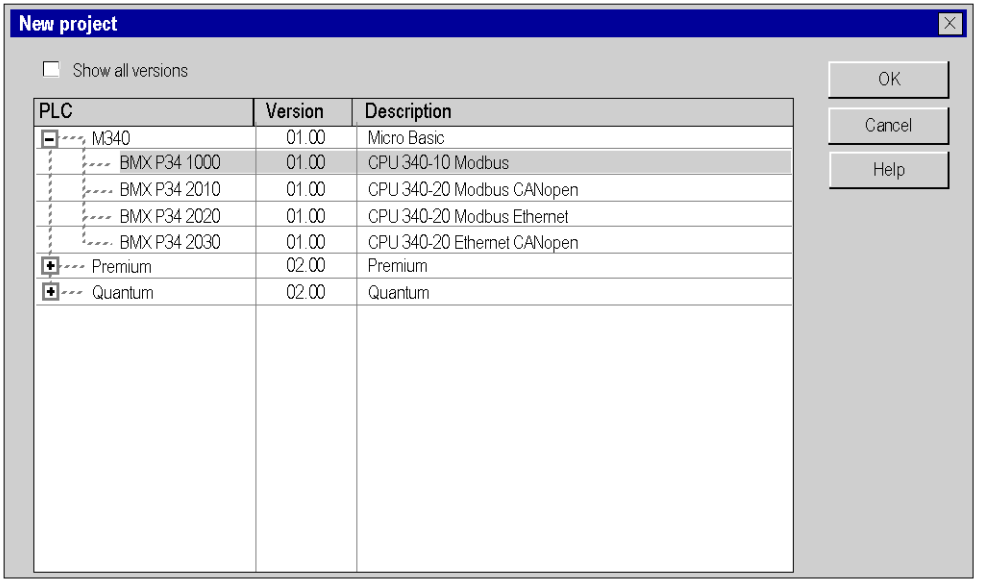
Aim of this sub-section

This sub-section guides the user in the selection and replacement of the PLC station processor.

Choose and Replace the Controller

Select the Controller

Choose a controller when you create the project. (This choice is editable.)

Step	Action
1	From the welcome screen of the software, select the **New** command from the **File** menu.
2	Open the type of controller required. Example: 
3	If you want to see all controller versions, click **Show all versions**.
2	Select the controller.
4	Validate with **OK**.

Rules

For a Modicon M580 station:

- The controller occupies the slots 00 and 01.
- The controller is double format.

For a Modicon M340 station:

- The controller occupies the slot 0.
- The controller is single format.

For a Modicon Premium station:

- The controller occupies the slot 0, it may occupy slot 1 if a double format power supply is configured.
- The controller is single format or double format (in this case it occupies two slots).

For a Modicon Quantum station:

- The controller can occupy any position in the rack.
- The controller is single format or double format (in this case it occupies two slots).

You cannot move or copy a controller in another rack. The controller cannot be deleted from the rack; it can only be replaced.

Modicon M580 Controller Restriction

The following Modicon M580 controller versions cannot be replaced:

- *upgrade:* OS versions earlier than 2.00 cannot be replaced by OS version 2.00 or any comparable subsequent version(s).
- *downgrade:* OS version 2.00 or any comparable subsequent version(s) cannot be replaced by OS version 2.00 or earlier.

If a controller replacement mentioned previously is needed, proceed as follows:

Step	Action
1	Export, page 1686 the original application program and/or elements (I/O configuration, DFB type, animation table, ...).
2	Create a new configuration with the new Modicon M580 controller OS version.
3	Import, page 1686 original application program and/or elements.

Replace the Controller

The configuration editor helps you if you wish to replace the controller. A message is sent if a replacement is not authorized.

The new controller has to belong to the same controller family as the previously configured controller.

If the new controller does not support certain previously configured input/output modules, an on-screen message reports the detection of the error when the project is analyzed. Resolve these incompatibility issues before you continue.

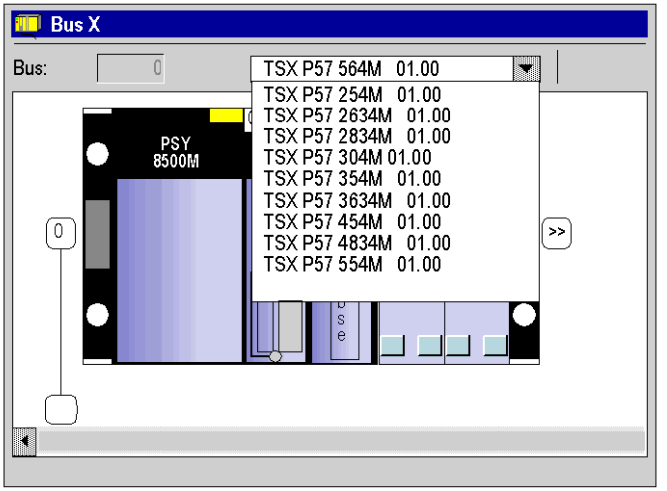
NOTE: This operation is only possible in offline mode (controller not connected).

You may proceed in accordance with two methods:

Method 1:

Step	Action
1	From the bus editor, select the controller.
2	Using the contextual menu, click **Replace Processor...**.
3	Select the required controller.
4	Validate with **OK**.

Method 2:

Step	Action
1	From the bus editor, display the list of controllers. To see all the controller versions, select **Show all versions**. Example: 
2	Select the required controller.
3	Confirm your selection.

NOTE: When making a controller replacement, if an FFB used in the application is not available for the selected controller, you have to remove it from the Application sections and you have to remove it from the Application itself by the **Tools > Types Library Manager > Purge Unused Types** option.

Configuration of the modules in the PLC station

Aim of this sub-section

This sub-section guides the user in the management of the module(s) in the rack(s) of the PLC station.

How to configure the modules in the PLC station

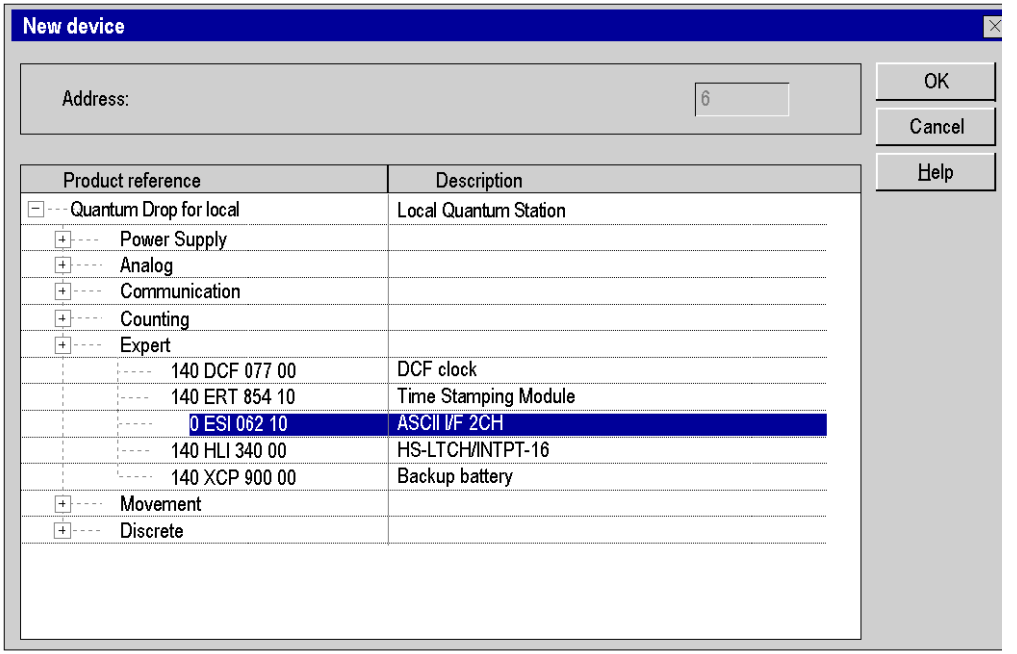
Introduction

You have two options for inserting the modules in the rack:

- either you use the functions provided by the bus editor,
- or you use the hardware catalog provided by the bus editor.

Positioning a module

Carry out the following actions:

Step	Action
1	Use the mouse to select the position of the module to insert; eight handles appear around the module.
2	Using the contextual menu, click **New Device**.
3	Select the required application and module. Example: 
4	Click **OK** to confirm.

NOTE: The double format modules are shown on the list of modules available when the slot preceding or following the selected slot is vacant.

In the case of a Premium or a Modicon M340, the position 0 may only be occupied by a double format power supply or a processor.

Moving a module

A module may be moved:

- into the rack,
- into another rack if the station has several of them.

Carry out the following actions:

Step	Action
1	With the mouse, select the required module.
2	Drag and drop the module into new position.

NOTE: If the module cannot be moved, this is indicated by the bus editor.

Rules:

The objects associated with the module are:

- deleted, then re-created automatically at the new address,
- replaced in:
  - the program,
  - the variable editor after user confirmation,
  - the animation tables,
  - the operator screens.

The **symbols** associated with the moved module objects are attached to the objects at the new address.

The module that has been moved keeps all its parameters.

For a **Modicon M340**,**Premium** or **Atrium** station, if a discrete module with a **RUN/STOP**-configured channel is moved, the **RUN/STOP** bit address remains unchanged. It is necessary to ensure that the **RUN/STOP** input corresponds to a valid discrete input address.

When the addresses of the module are used in an EF (e.g.: Send_REQ, Read_VAR,...), these are not updated automatically.

Copy a module

Carry out the following actions:

Step	Action
1	With the mouse, select the required module.
2	Using the contextual menu, click **Copy**.
3	Use the mouse to select the target position, then using the contextual menu click **Paste**.

Rules:

The objects associated with the module are copied to the new address.

The duplicated module keeps all its parameters.

The duplication of a module associated with an event (at least one module channel is associated with an event) is not possible. An event may be associated with several **channels/modules** at the same time.

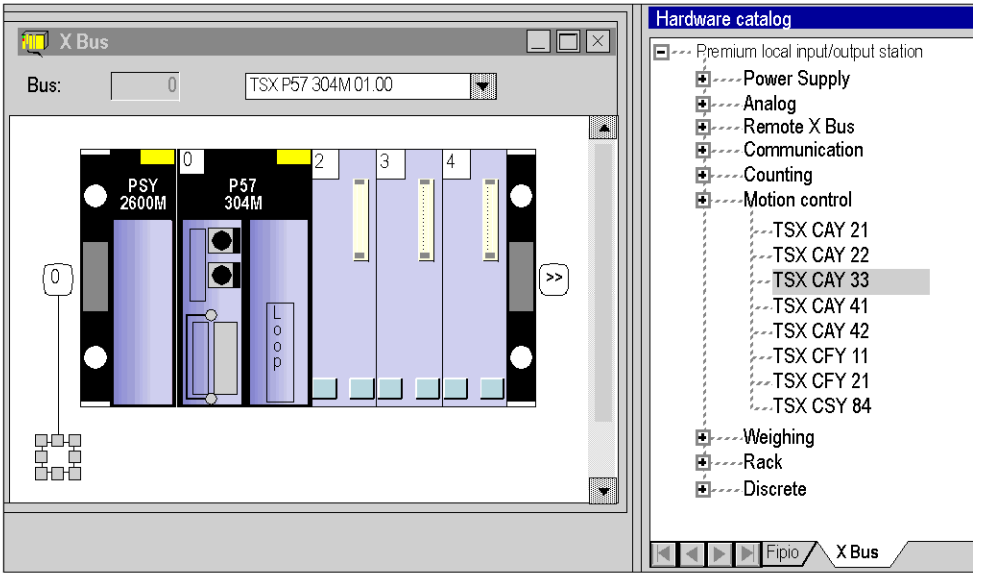
Delete a module

Carry out the following actions:

Step	Action
1	With the mouse, select the required module.
2	Via the popup menu the select the **Delete Module** command.

Hardware catalog

Using the Hardware catalog, you can insert modules in the rack:



Carry out the following actions:

Step	Action
1	From the **Configuration** directory in the project browser, select the subdirectory representing the station. **Result**: the bus editor and hardware catalog appear.
2	Select the directory of the required specific application in the hardware catalog, and expand it by clicking on **+**.
3	Select the module, and, holding down the mouse button, move it to the required position on the rack in the bus editor.

Configuration of devices on the field bus

Aim of this sub-section

This sub-section guides the user in the management of devices on field buses managed by the Modicon M340, Premium and Quantum stations.

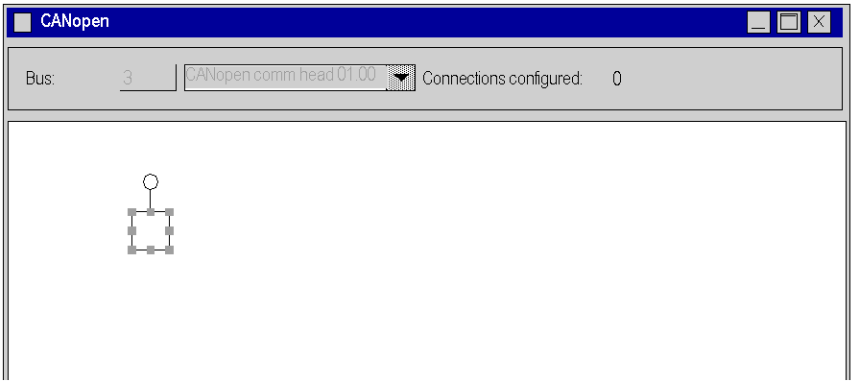
Configuration of the CANopen Slaves for Modicon M340

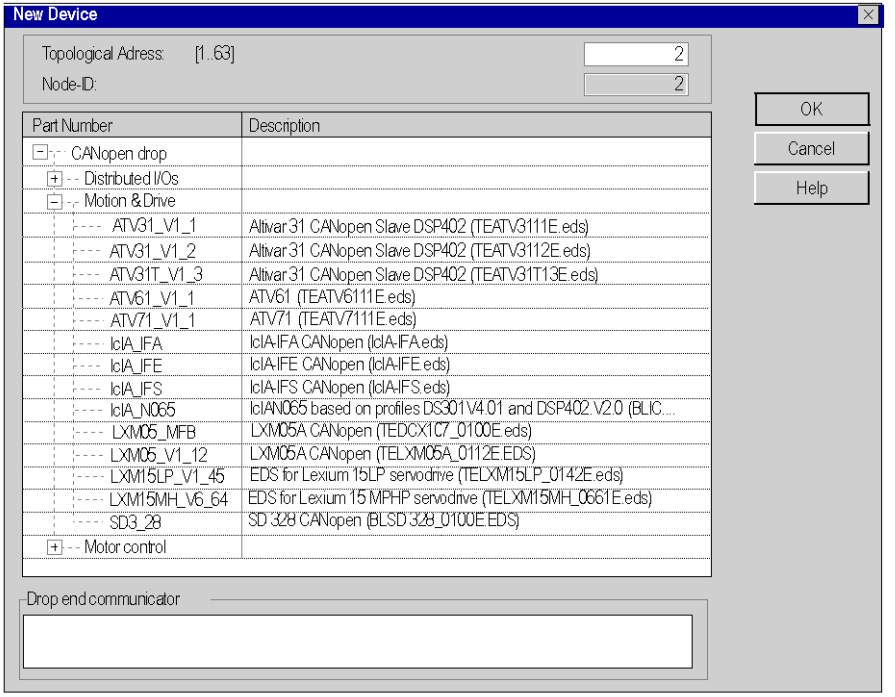
Introduction

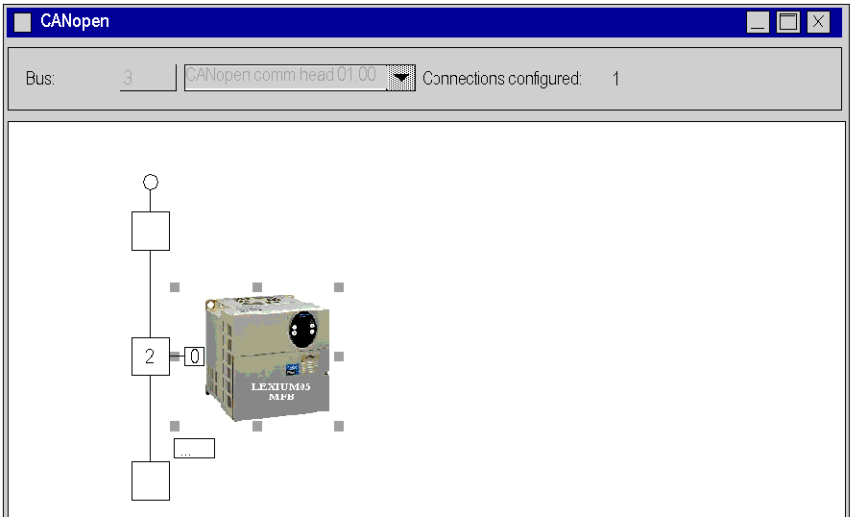
For Modicon M340 PLCs you can configure the CANopen bus with slaves (63 slaves maximum). The configuration is completely carried out with Control Expert, the use of another software is not necessary.

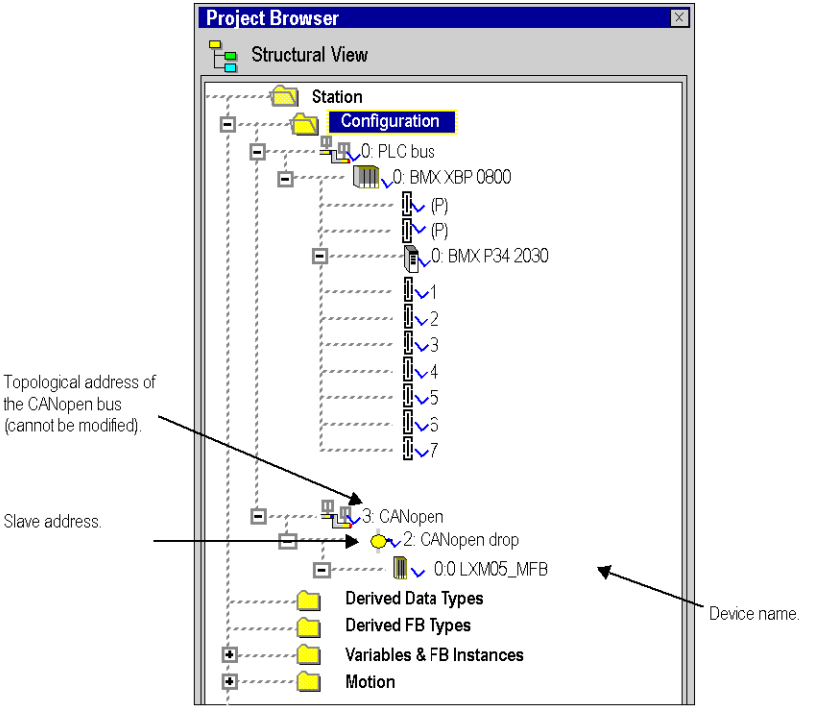
How to Configure the CANopen Slave

This table describes the procedure to configure the CANopen slave.

Step	Action
1	In the Control Expert **Project Browser**, fully expand the **Configuration** directory and then double-click on **CANopen** . **Result:** The CANopen window appears: 
2	Select **Edit** > **New device**.

Step	Action
	Result: The New Device window appears: 
3	Set the address slave in Topological Address. Choose the slave device.

Step	Action
4	Click on OK to confirm the choice. **Result:** The CANopen window appears with the new device selected: 
5	On the project browser the CANopen bus appears with the slaves.

Step	Action
	

Other functions

You can also add, delete, move (see Modicon M340, CANopen, User Manual) a slave on the bus.

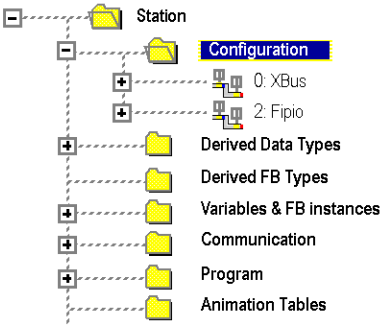
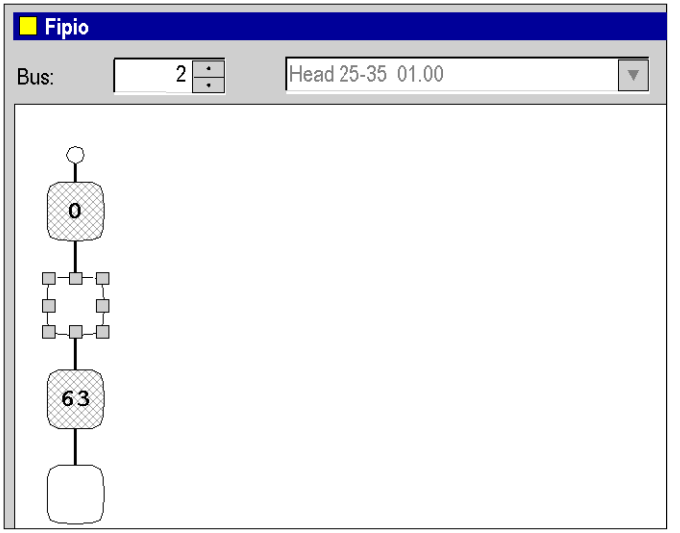
How to access and create a Fipio field bus

Introduction

Premium PLCs offer a decentralized input/output architecture solution with the **Fipio** bus. You can configure (see Premium and Atrium Using EcoStruxure™ Control Expert, Fipio Bus, Setup Manual) 127 devices.

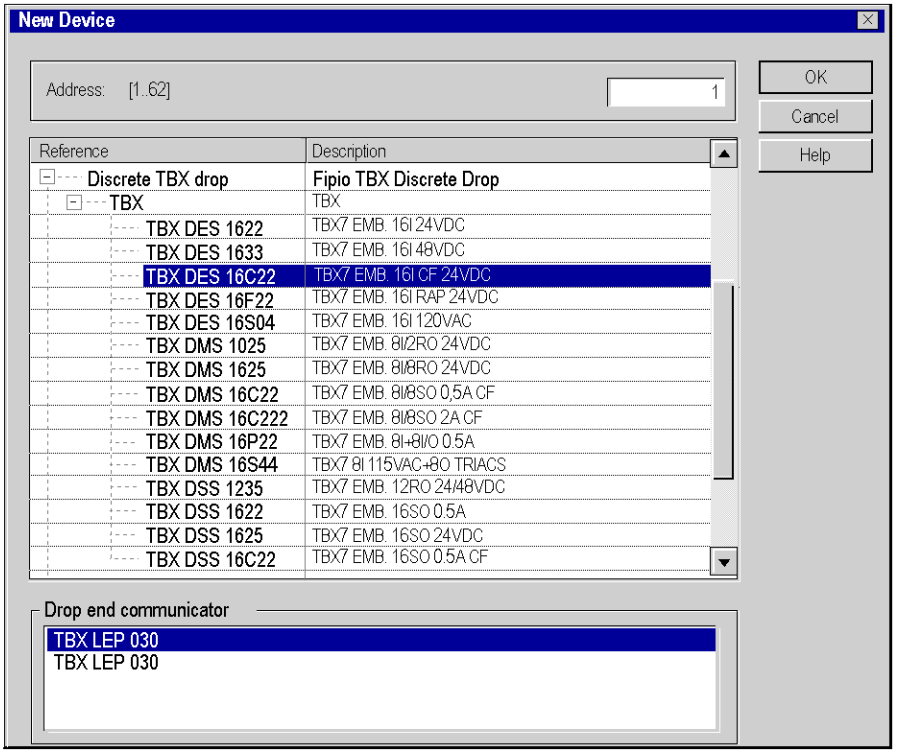
Procedure

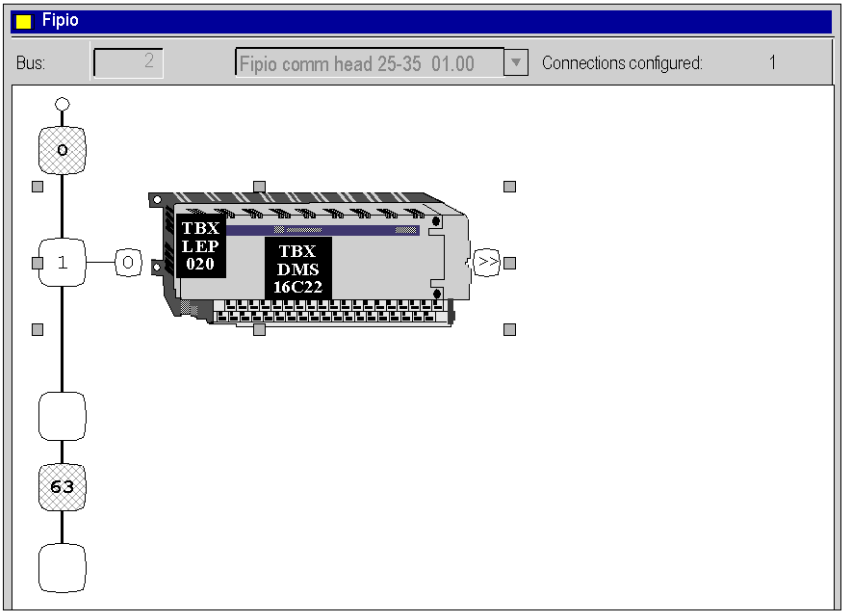
To access the Fipio field bus, perform the following actions:

Step	Action
1	From the project navigator, deploy the **Configuration** directory. **Result**: the following screen appears 
2	Select the Fipio sub-directory and select the **Open**command using the contextual menu. **Result**: the **Fipio** window appears 

Creating a Fipio bus

This operation is used to add, via the software, a device connected to the Fipio bus.

Step	Action
1	Access the Fipio configuration screen:
2	Right-click on the logical address of the connection point, at the location where the module has to be connected (available addresses from 1 to 62 and 64 to 127, addresses 0 and 63 being reserved by the system). **Result**: the **New Device** screen appears. 
3	Enter the number of the connection point corresponding to the address. By default, the Control Expert software offers the first free consecutive address.
4	In the **Reference** field, enter the type of device to connect to the bus.

Step	Action
5	In the **Communicator** field, select the element type enabling communication on the Fipio bus. For modules with built-in communicators, this window does not appear.
6	Confirm with **Ok**. **Result:** the module is declared. 

Other functions

You can also:

- add an extension module (see Premium and Atrium Using EcoStruxure™ Control Expert, Fipio Bus, Setup Manual) to the bus,
- delete, copy, paste (see Premium and Atrium Using EcoStruxure™ Control Expert, Fipio Bus, Setup Manual) bus devices,
- change (see Premium and Atrium Using EcoStruxure™ Control Expert, Fipio Bus, Setup Manual) the Fipio communicator.

Creating and Accessing RIO\DIO Field Buses

Introduction

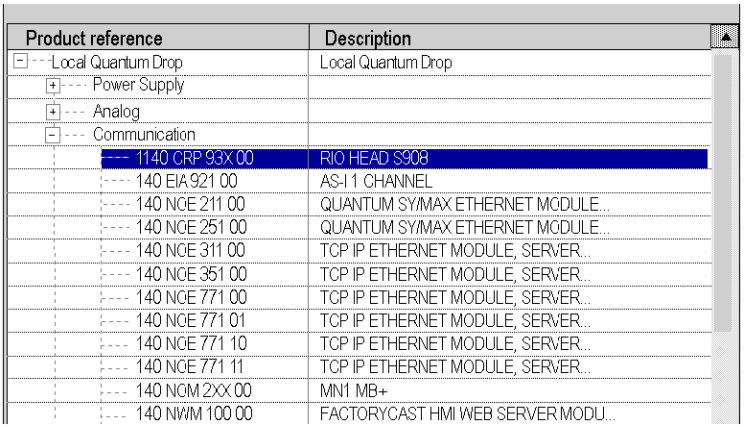
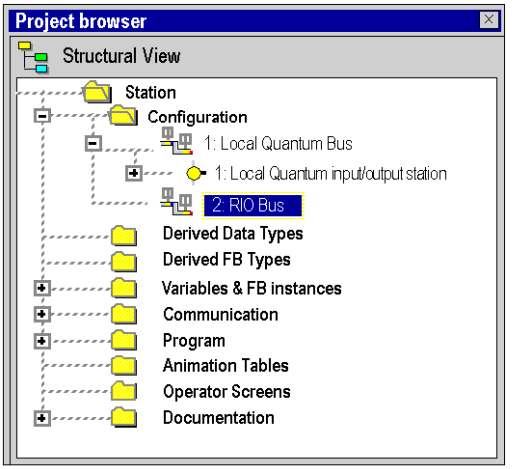
Quantum PLCs offer a decentralized input/output architecture solution:

- **RIO** field bus networks are based on the S908 input/output decentralization network technology. Up to 31 decentralized stations may be configured, with each station capable of supporting up to 128 input/output words.
- **DIO** field bus networks are based on Modbus Plus technology. 32 subscribers may be configured over 500 meters/1640 feet (receiving 64 subscribers over 2000 meters/6560 feet).

Creating a RIO Bus

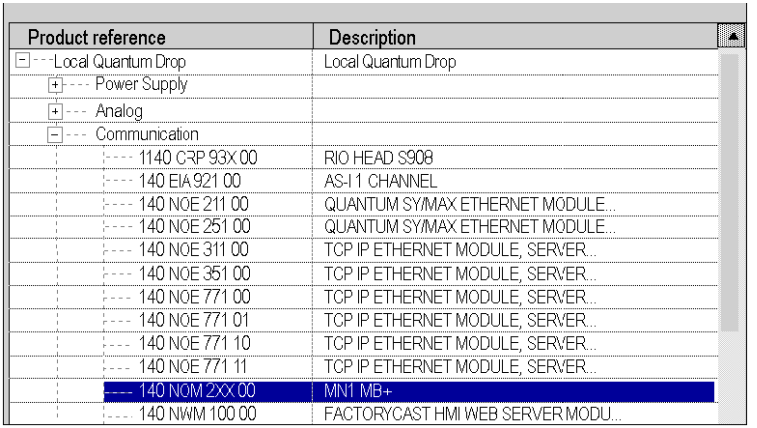
The following table describes the procedure for creating a RIO bus from a communication module:

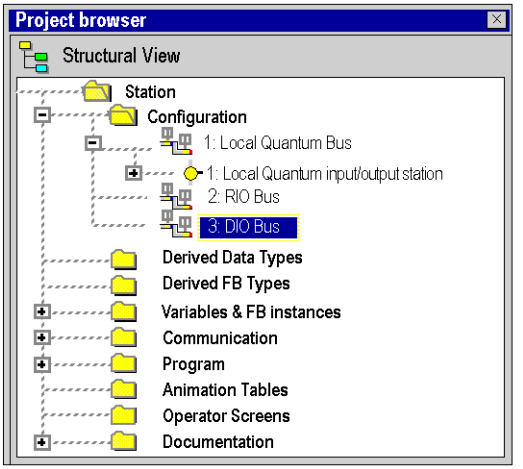
Step	Action
1	In the bus editor, select the slot where you wish to insert the communication module.
2	Select **New Device** in the contextual menu. **Result:** The **New Device** window appears.

Step	Action
3	Expand the *Communication* directory. **Result:** The following window appears: 
4	To create a RIO bus, select a 140 CRP 93x 00 module. **Result:** The bus appears in the project browser: 

Creating a DIO Bus

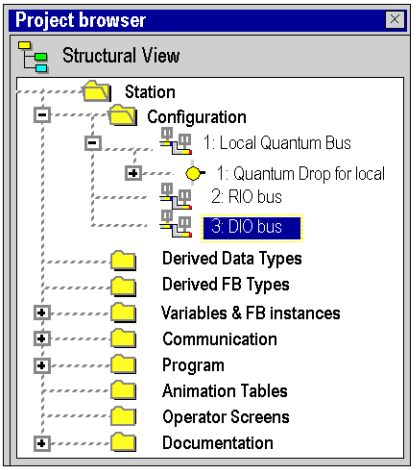
The following table describes the procedure for creating a DIO bus from a communication module:

Step	Action
1	In the bus editor, select the slot where you wish to insert the communication module.
2	Select **New Device** in the contextual menu. **Result:** The **New Device** window appears.
3	Expand the *Communication* directory. **Result:** The following window appears: 
4	To create a bus, select a 140 NOM 2XX 00 module. **Result:** The module appears in the rack.
5	Double-click the 140 NOM 2XX 00 module's Modbus Plus port. **Result:** The bus configuration window appears.

Step	Action
6	Check the box marked **DIO Bus**.
7	Confirm the configuration. **Result:** The DIO bus appears in the project browser. 

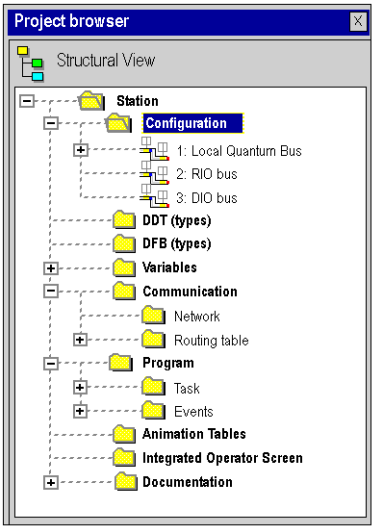
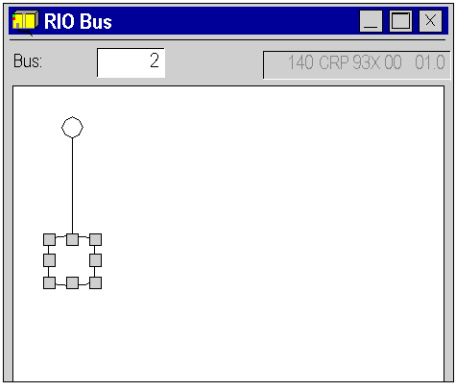
Creating a DIO Bus from the Processor

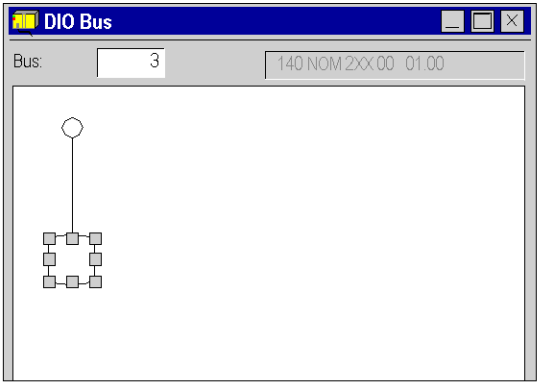
The following table describes the procedure for creating a DIO bus from the processor:

Step	Action
1	In the bus editor, double-click the processor's Modbus Plus port.
2	Check the box marked **DIO Bus**.
3	Confirm the configuration. **Result:** The DIO bus appears in the project browser: 

Accessing a RIO or DIO Bus

To access a bus, carry out the following actions:

Step	Action
1	In the project browser, open the *Configuration* directory. Example: 
2	Select the *RIO bus* or *DIO bus* subdirectory depending on the type of bus you want to open. Next, select the **Open** command in the popup menu. **Result:** The following window is displayed for the RIO bus:  **Result:** The following window is displayed for the DIO bus:

Step	Action
	

How to configure devices on the RIO/DIO bus

Introduction

You can configure devices on field buses using the functions provided by the bus editor.

You can configure modules in the devices on field buses according to two methods:

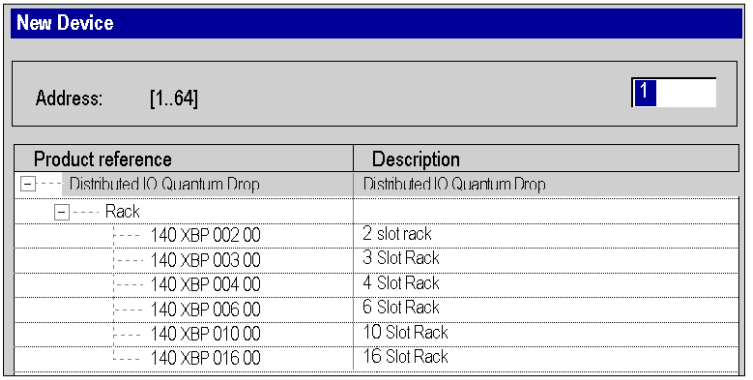
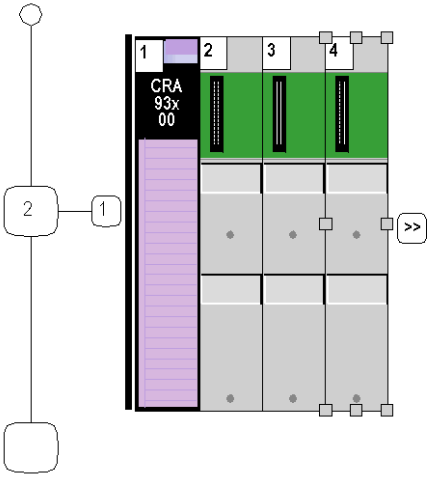
- either by using the functions provided by the bus editor,
- or from the hardware catalog provided by the bus editor.

You can also make copies, perform movements, or delete modules from a field bus device.

Configuration of a device

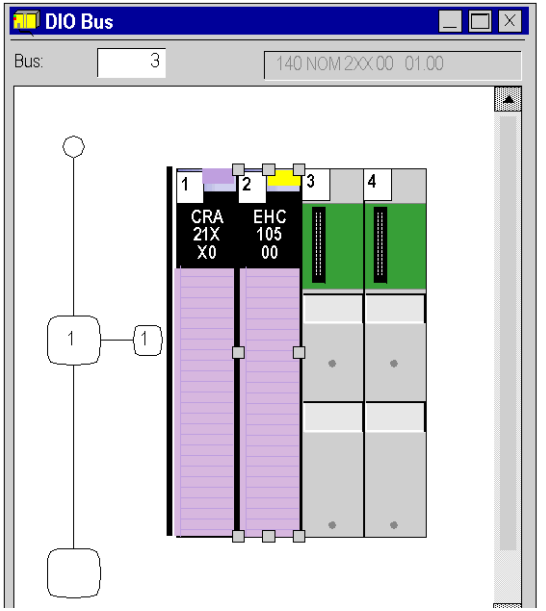
Carry out the following actions:

Step	Action
1	From the **Configuration** directory in the project browser, select the **RIO Bus** or **DIO Bus** subdirectory.
2	Select **Open** in the contextual menu.

Step	Action
3	Select the connection point and via the popup menu select the **New Device** command. **Result**: The **New Device** window appears. 
4	Select the required rack and validate with **OK**. **Result**: The rack appears in the bus. 

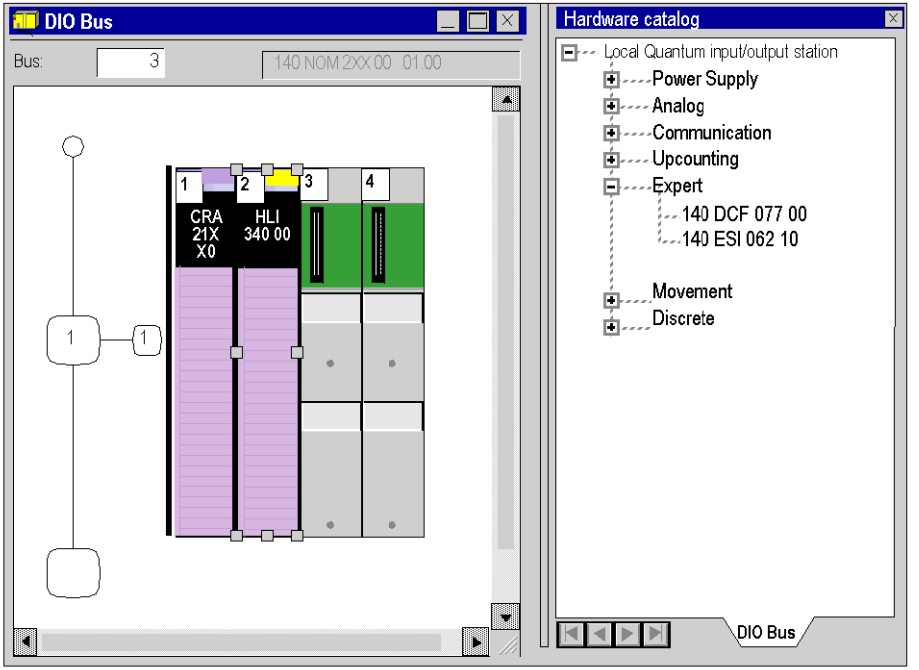
Configuration of a module in a device

Carry out the following actions:

Step	Action
1	Select the required rack (see procedure above: Configuration of a device).
2	To configure a module in the rack, select the module slot.
3	Select **New Device** in the contextual menu. **Result**: The **New Device** window appears.
4	Select the module to be inserted. **Result**: The module appears in the rack. 

Hardware catalog

Using the hardware catalog, you can insert modules into a device on the field bus:



Proceed with the following steps:

Step	Action
1	From the **Configuration** directory in the project browser, select the **DIO Bus** subdirectory. **Result**: The field bus editor and hardware catalog appear.
2	Select the directory of the required specific application in the hardware catalog, and expand it by clicking on **+**.
3	Select the module, and, holding down the mouse button, move it to the required position on the rack in the bus editor.

Copy a module

Proceed with the following steps:

Step	Action
1	With the mouse, select the required module.
2	Using the contextual menu, click **Copy**.
3	With the mouse, select the target position, then via the popup menu select the **Paste** command

Rules:

The duplicated module keeps all its parameters.

The duplication of a module associated with an event (at least one module channel is associated with an event) is not possible. An event may be associated with several **channels/modules** at the same time.

Moving a module

A module may be moved:

- into the rack,
- into another rack if the station has several of them.

Proceed with the following steps:

Step	Action
1	With the mouse, select the required module.
2	Drag and drop the module into new position.

NOTE: If the module cannot be moved, this is indicated by the bus editor.

Rules:

The objects associated with the module are:

- deleted, then re-created automatically at the new address,
- replaced in:
  - the program,
  - the variable editor after user confirmation,
  - the animation tables,
  - The operator screens.

The **symbols** associated with the moved module objects are attached to the objects at the new address.

The module that has been moved keeps all its parameters.

Deleting a module

Proceed with the following steps:

Step	Action
1	With the mouse, select the required module.
2	Via the popup menu, select the **Delete Module** command.

Consumption management

Aim of this sub-section

This sub-section guides the user during the selection of modules so as not to exceed the limits:

- for consuming power,
- for consuming application-specific channels.

Power consumption budget

At a Glance

A power consumption budget is made for:

- the rack power supply module,
- each module (processor, input/output module) depending on the rack power supply module.

This budget is presented in the form of a bar chart where each color has a particular significance. For each voltage, it indicates:

- the current power flow rate: **green**,
- the quantity of power still available: **white**,
- a power overload: **red**, when an excess occurs, a message is displayed.
- the total power (same color code).

Power supply module budget

This consumption budget shows the amount of power discharged **by the power supply** for each voltage it supplies, as well as the total power.

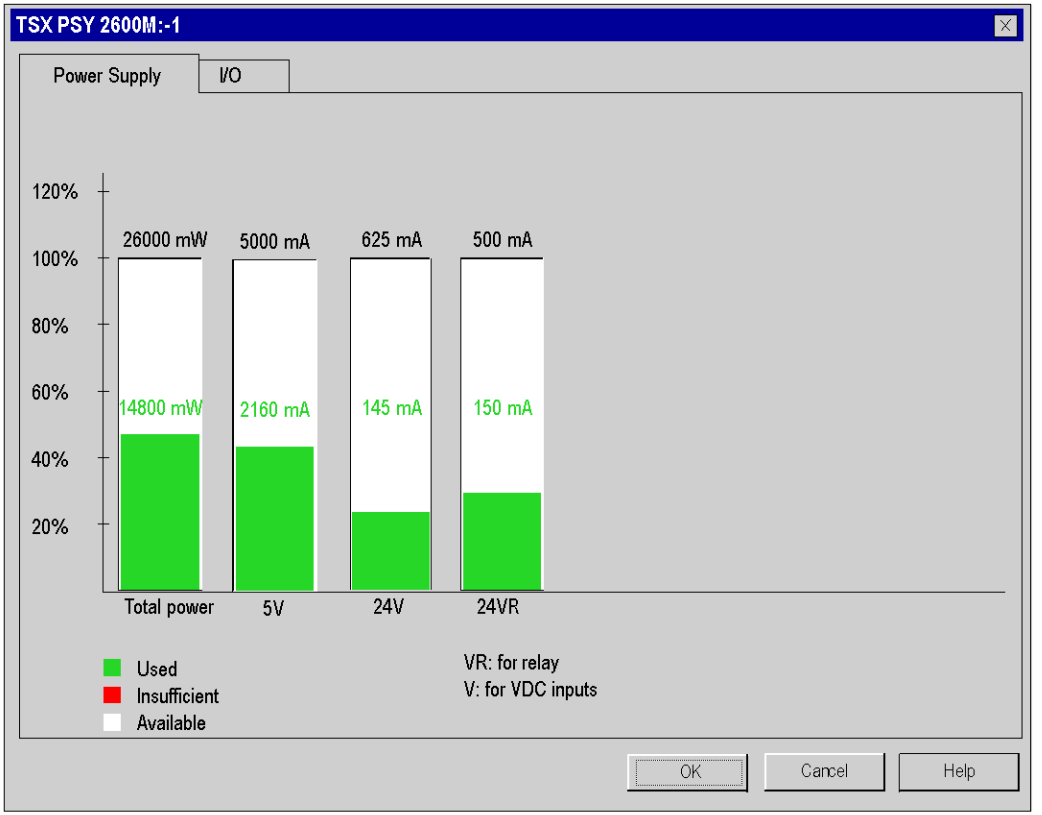
When you add or remove a module, the budget is adjusted on opening the **Power Supply and IO Budget** window.

NOTE: The Power Supply and IO Budget window must be closed to be able to delete or add a module.

How to access a consumption budget:

Steps	Action
1	With the mouse, select the power supply module.
2	Using the contextual menu, click **Power Supply and IO Budget**. The bar chart is displayed.
3	Select the **Power Supply** tab.

Budget:



NOTE: The ASY 800 24 V power consumption is always included in 24 VR (Internal Rack Power) even if configured with an external power supply.

Budget of the other modules

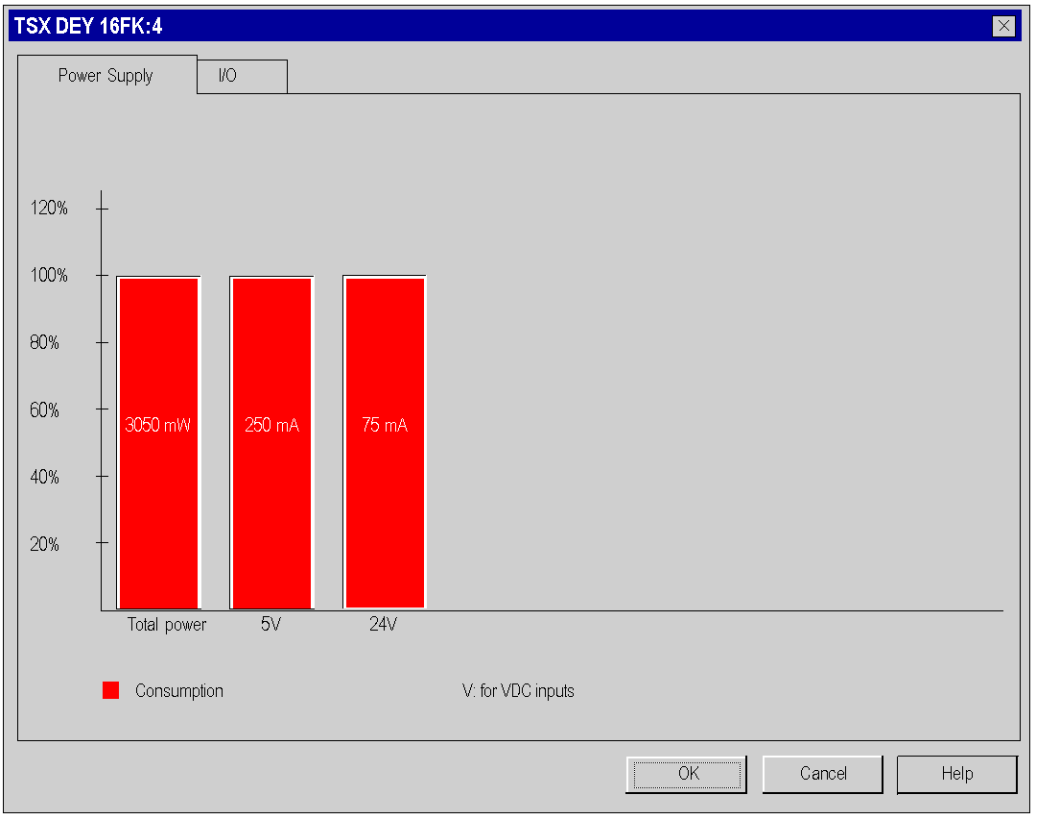
This consumption budget shows the amount of power discharged in **the module** for each voltage it uses, as well as the total power.

NOTE: The Power Supply and IO Budget window must be closed to be able to delete or add a module.

How to access a consumption budget:

Steps	Action
1	With the mouse, select the module of your choice.
2	Using the contextual menu, click **Power Supply and IO Budget**. The bar chart is displayed.
3	Select the **Power Supply** tab.

Budget:



Budget on the number of application-specific channels

At a Glance

A budget on the number of application-specific channels used is made for:

- the processor module of the station
- each module (processor, adapter, I/O module) of the station

This budget is presented in the form of a bar chart where each color has a particular significance, it indicates for each application:

- the number of application-specific channels configured: **green**,
- the number of application-specific channels still available: **white**,
- the excess application-specific channels (not managed by the processor): **red**, when an excess occurs, a message is displayed.

Processor and adapter budget

This budget shows the number of channels configured by applications **in the station**.

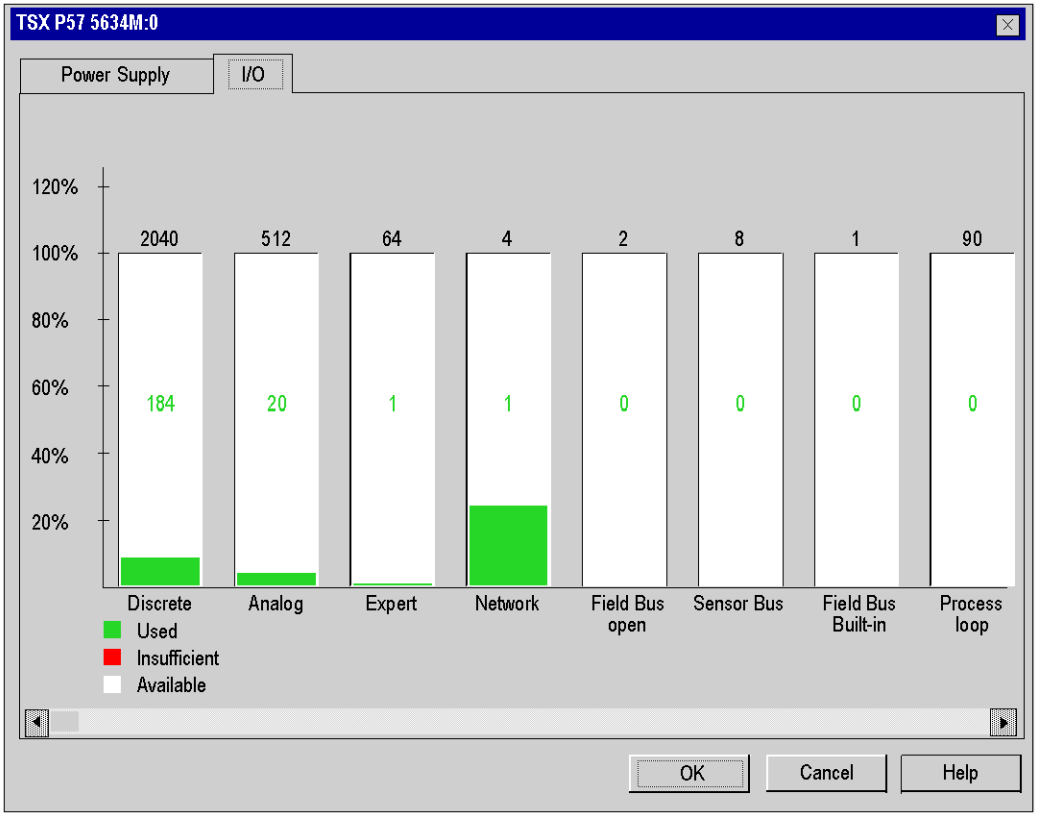
When you add or remove a module, the budget is adjusted on opening the **Power Supply and IO Budget** window.

NOTE: The **Power Supply and IO Budget** window must be closed to be able to delete or add a module.

How to access the budget:

Steps	Action
1	With the mouse, select the required processor or adapter.
2	Using the contextual menu, click **Power Supply and IO Budget**. The bar chart is displayed.
3	Select the **I/O** tab.

Example of IO budget content:



Input/output modules budget

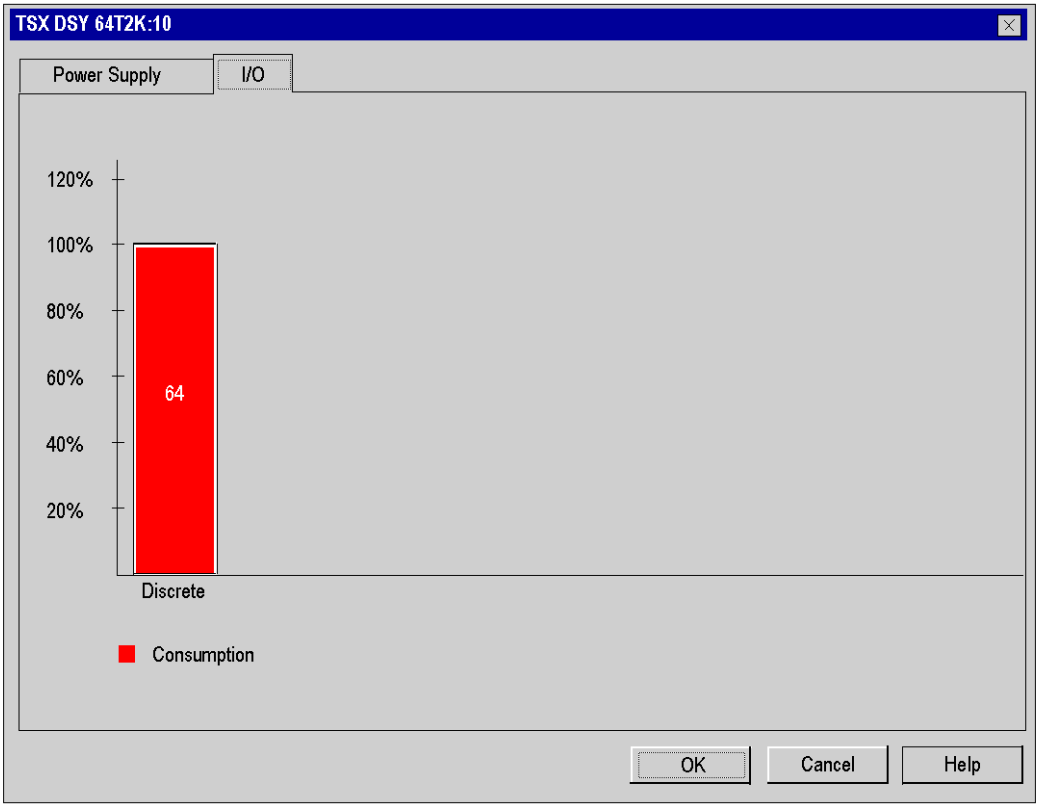
This budget shows the number of application-specific channels configured in the module.

NOTE: The **Power Supply and IO Budget** window must be closed to be able to delete or add a module.

How to access the budget of the number of application-specific channels:

Steps	Action
1	Using the mouse select the input/output module of your choice.
2	Using the contextual menu, click **Power Supply and IO Budget**. The bar chart is displayed.
3	Select the **I/O** tab.

Budget:


TSX DSY 64T2K:10
Power Supply
I/O
120%
100%
80%
60%
40%
20%
64
Discrete
Consumption
OK
Cancel
Help

Bus editors in online mode

Aim of this sub-section

This sub-section guides the user in diagnosing problems linked to the PLC station configuration.

Online Information

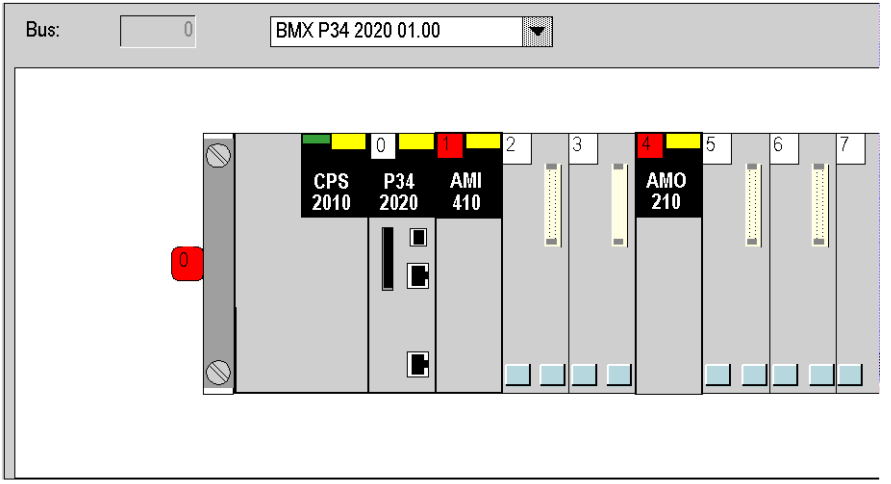
Animation

The bus editor uses color animation to inform you about the status of a module on the bus.

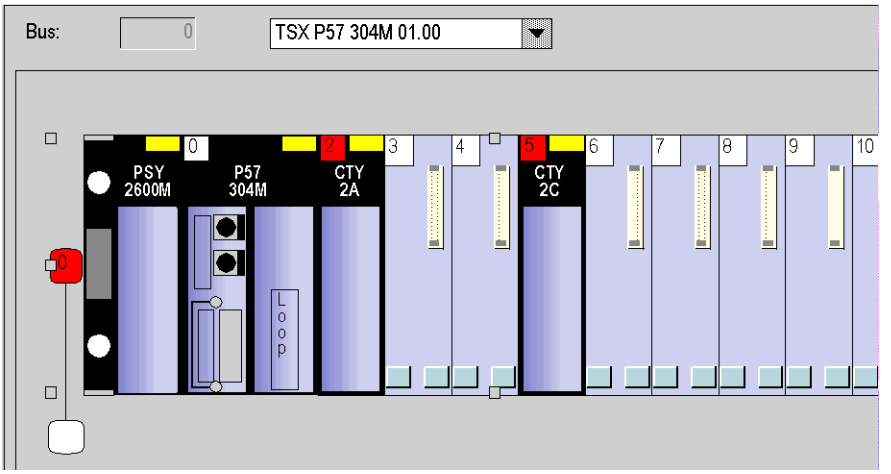
The module's slot number in the rack appears in red if the module is:

- missing
- detecting an error condition
- incorrectly configured

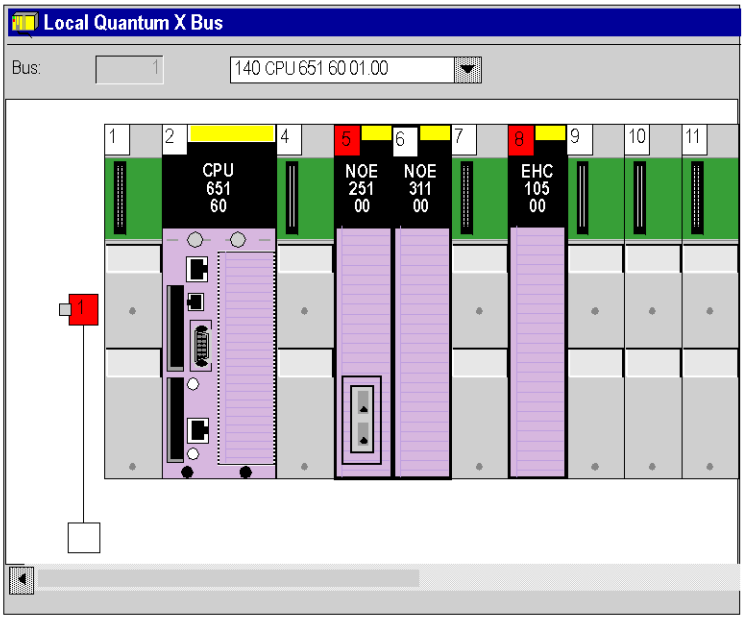
Modicon M340 controllers:



Premium controllers:



Quantum controllers:



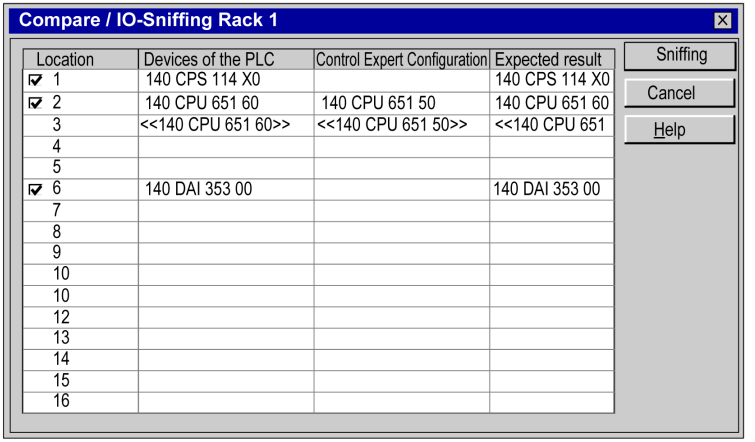
If an error is detected on a rack module, the rack number appears in red.

Automatic Detection of the Configuration

When connected to a **Quantum** station, the programming software is able to detect the different modules that make up the configuration. This operation can be performed before programming the application, or after the application has been programmed.

Procedure

The following table describes the procedure for detecting the different elements of a controller configuration.

Step	Action
1	In online mode (controller in Stop), open the configuration editor.
2	Select the rack.
3	From the **Services** menu, select **Compare/IO-Sniffing...**. **Result**: The following window opens. 
4	Select the **Slot** check box if you want: • to add the module detected on initial configuration, • to replace the configured module by the one detected. If the detected module is identical to the configured module, you can keep this module by selecting the **Slot** check box. If you leave the **Slot** check box cleared, the module specified on initial configuration remains.
5	Validate the new configuration by clicking **Sniff**.
6	Click **OK** to confirm your choice. **Result**: The new configuration appears in the configuration editor.

Modules editor

What's in This Chapter

Configuration of Modicon M340 processors 1348
Configuration of Premium processors..... 1361
Configuration of the Quantum processors 1371
Configuration of the input\output modules for a Premium \Atrium, Quantum or Modicon M340 station..... 1380

Aim of this section

This section guides the user in the configuration of the modules comprising a Premium/ Atrium, Quantum or Modicon M340 station.

For Modicon M580 PLC stations, refer to chapter *Configuring the CPU in Control Expert* (see Modicon M580, Hardware, Reference Manual).

For Momentum, refer to chapter *Configuring the 171 CBU 78090, 171 CBU 98090 and 171 CBU 98091 Processors in Control Expert* (see Momentum for EcoStruxure™ Control Expert, 171 CBU 78090, 171 CBU 98090, 171 CBU 98091 Processors, User Guide).

Configuration of Modicon M340 processors

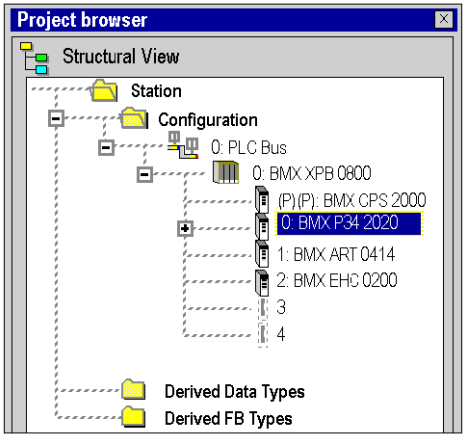
Aim of this sub-section

This sub-section guides the user in the configuration of processor modules for Modicon M340 stations.

Configuration of Modicon M340 Controllers

Access to the Configuration Screen

Proceed with the following steps:

Step	Action
1	Select the controller. Example: 
2	Using the contextual menu, click **Open**.
3	Select the **Configuration** tab.

Memory Management

The configuration screen differs, depending on the type of memory management to be used.

The option you select here must comply with the type of memory management you select for the input and output modules, page 1226.

The following settings are possible:

If...	Then...
Topological memory management is configured for the controller	you can only select **Topological** memory management for the discrete and analog modules.
Mixed topological and State RAM memory management is configured for the controller	you can select either **Topological** or **Mixed topological and State RAM** memory management for the discrete and analog modules.

The memory management settings of the controller are compared with the memory management settings of the modules during **Analyze**. If any inconsistencies are detected, a appropriate message is displayed in the output window.

The type of memory management you selected for the controller is used as default setting for the **Memory Management** parameter in the **Memory** tab for input / output configuration, page 1225.

NOTE:

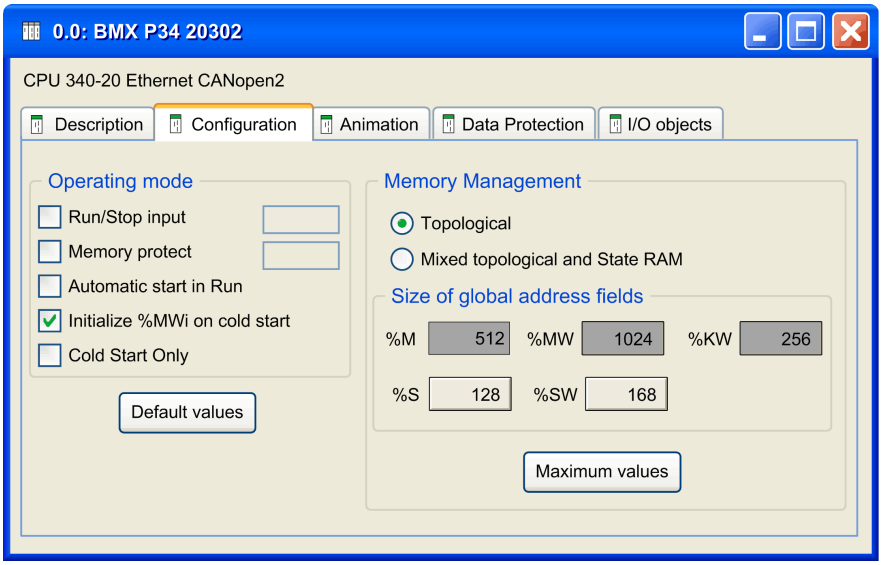
To import a legacy LL984 Compact application that uses Modbus requests to communicate with an HMI device, use State RAM addressing to preserve the Modbus exchange between the controller and the HMI.

NOTE: If you use the fallback configuration in your hardware configuration, take care that the fallback values can only be applied if the **Topological** option is selected.

Configuration of the Controller

NOTE: You can modify the parameters of this dialog box only in offline mode.

Configuration screen for topological memory management:



Proceed with the following steps:

Step	Action
1	If you wish, enable the **Run/Stop input** option, page 1354. Do not enable this option if the associated discrete input is mapped in State RAM because this inhibits the controller start-up.
2	If you wish, enable the **Memory protect** option. The protection is activated by an input bit. The protection level depends on the Firmware versions, see detailed information, page 1354.
3	If you wish, enable the **Automatic start in Run** option, page 1355.
4	Confirm whether the internal words %MWi, page 1355 are to be initialized on cold start (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) triggered by software (application download, initialize command, restore command, `%S0` activation, cold start button on the controller screen) or not. By selecting the check box, the controller writes the initial values to `%MWI`.
5	If you wish, enable the **Cold Start Only** option, page 1355.
6	Define the type of **Memory Management**: • select the option **Topological** for topological addressing • select the option **Mixed topological and State RAM** for a mixture of topological and defined State RAM addressing
7	Proceed with address configuration for the 2 different addressing types as described in the following separate descriptions.

NOTE: The State RAM address starts at address 0, but the first available address for discrete and analog modules is `%I1`, `%M1`, `%IW1`, and `%MW1`.

Topological Addressing

Proceed with address configuration for topological addressing as follows:

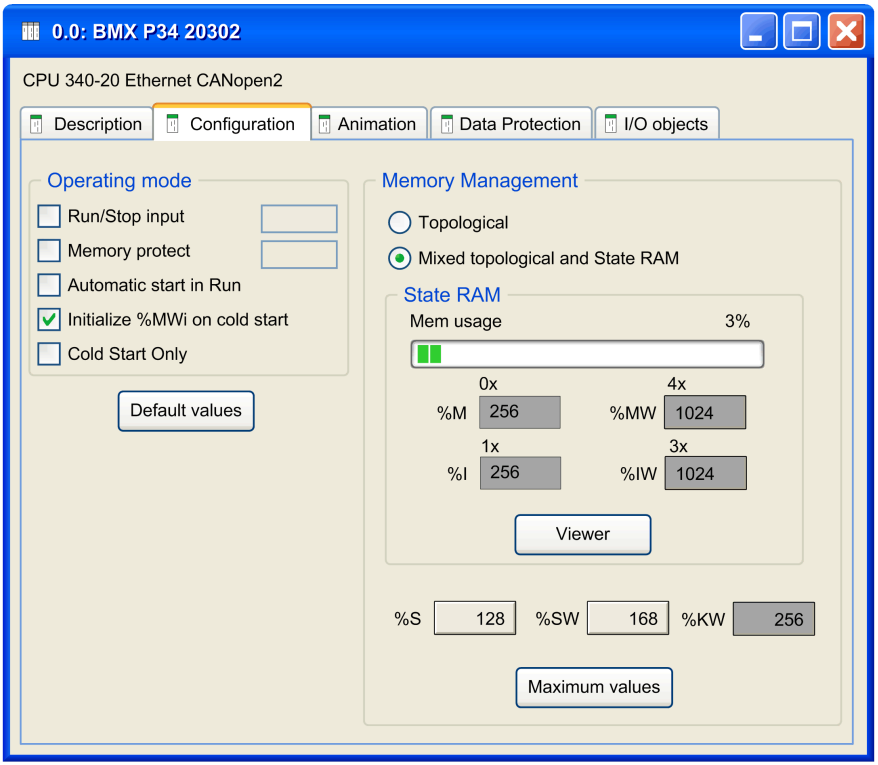
Step	Action
1	In the **Size of global address fields** of the dialog box define for the application (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) the number of internal bits `%M`.
2	Define the number of internal words `%MW`.
3	Define the number of constants `%KW`.

The system bits and system words are fixed by the manufacturer.

M340 applications of version 6.0 and earlier are automatically configured with this address configuration for topological addressing.

Mixed Topological and State RAM Addressing

Configuration screen for mixed topological and State RAM memory management:



For mixed topological and State RAM addressing, the following options are additionally available in the configuration screen:

Option	Value	Description
Mem usage	The value (expressed as a percentage and displayed on the scale bar) depends on the memory usage of the Modicon M340 configuration.	A bar graph displays the percentage of the memory already in use.
%M-0x	Enter the appropriate values. The values depend on the configuration.	Size of the different memory areas. **NOTE:** The values for **%IW** and **%MW** have to be divisible by 8.
%MW-4x		
%I-1x		
%IW-3x		
Viewer	–	Opens the State RAM Viewer tab, page 1357 which displays the allocation of used memory.

NOTE: The State RAM address starts at address 0, but the first available address for discrete and analog modules is `%I1`, `%M1`, `%IW1`, and `%MW1`.

NOTE: State RAM addresses are not verified during input. If any range overlapping is detected or any address exceeds the range of the State RAM, an appropriate message is displayed at the validation screen or during **Analyze**.

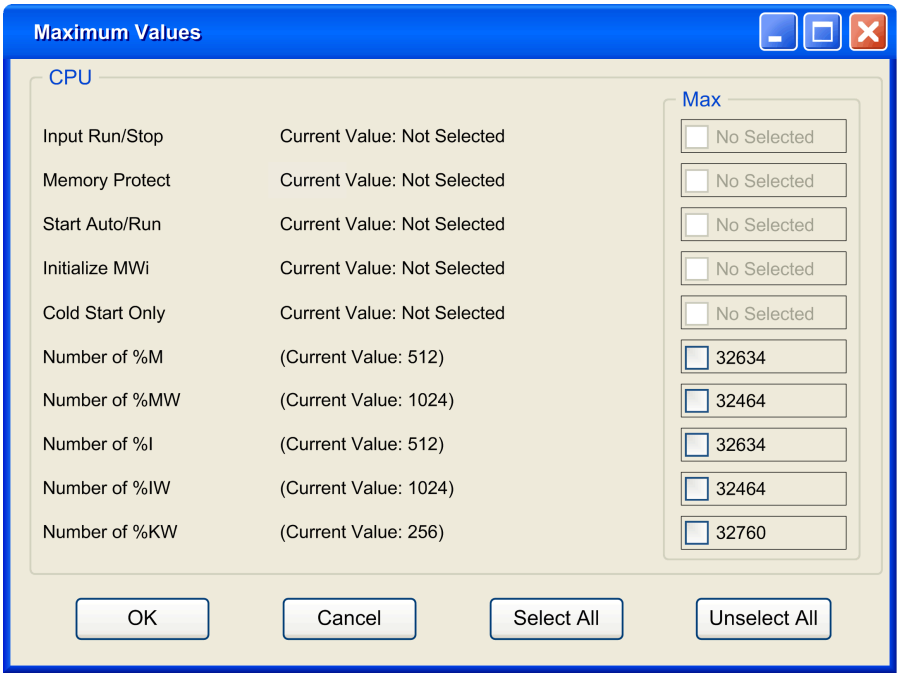
NOTE: With this type of memory management, all channels are associated to one task (MAST or FAST).

Pre-set Values

To select the:

- **default** values, press the **Default values** button,
- **maximum** values, press the **Maximum values** button,

Window displayed by pressing the **Default values** button:



Use of the command buttons:

Button	Role
Unselect All	Is used to invalidate all the default values posted on the right of the check boxes if they have been selected.
Select All	Is used to select all the default values posted on the right of the check boxes.
Cancel	Is used to exit.
OK	Is used to exit taking the values into account.

NOTE: With the **Maximum values** button, the window displayed is the same, only the values on the right of the check boxes are different.

Input RUN/STOP

The input **%Ir.m.c** can be parameterized to switch the controller to **RUN/STOP** mode in the following way:

- `%Ir.m.c` to 1 -> the controller switches to RUN (execution of the program),
- `%Ir.m.c` to 0 -> the controller switches to STOP mode (stop program execution).

NOTE: A STOP command always takes priority over a RUN command. A STOP command sent from a terminal or via the network has priority over the `%Ir.m.c` input.

An error on the RUN/STOP input causes a switch to STOP.

Do not enable this option if the associated discrete input is mapped in State RAM because this inhibits the start-up of the controller.

Memory Protect

The input **%Ir.m.c** can be parameterized to protect the internal application ram and the memory card in the following way:

- `%Ir.m.c` to 0 -> the internal application and the memory card are not protected,
- `%Ir.m.c` to 1 -> the internal application and the memory card are protected.

For Firmware versions < 2.60, the Memory Protect function prohibits the transfer of a project into the controller and modifications in online mode, but RUN/STOP commands are available.

For firmware version 2.60 and any subsequent supporting version(s), the Memory Protect function prohibits the transfer of a project into the controller and modifications in online mode, **for cybersecurity improvement, the connection to the controller is forbidden then the RUN/STOP command are not available**.

NOTE: If the input is in error (Input Module not physically plugged, incorrect wiring, input not powered...), %Ir.m.c is considered at 1 (memory is protected). To remove this protection in this configuration screen, the input should not be in error.

Automatic Start in Run

The enabling of this option automatically changes the controller to RUN mode at the time of a cold start (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).

Initialize %MWi

On a cold start (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) or on download if you select the check box (default state):

- the %MWi are handled like other global variables (initialized to 0 or initial value, according to current application) in all cold start cases,

On cold start or on download if you clear the check box:

- if %MW were previously saved in internal flash memory (using the %SW96 word) they are restored from internal flash memory,
- if not,
  - if cold start (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) is linked to a power-off or of a push on the reset button, the %MW are initialized:
  - if not, the current values of %MW are maintained.

NOTE: if the new (or restored) application has more %MW than the previous one, the added %MW are set to 0 (non-zero initial values are not applied).

Cold Start Only

If selected, this option forces the cold start (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) of the application, instead of the normal warm start (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).

By default, the Cold Start Only option is cleared.

The Cold Start Only option is only supported for controllers with OS version V2.1 or any subsequent supporting version(s).

An application using this functionality is not:

- downloadable on a controller with a previous version,
- executable on a controller with a previous version,
- usable with Unity Pro V4.0 or earlier.

  Unity Pro is the former name of Control Expert for version 13.1 or earlier.

  NOTE: The Cold Start Only check box is present only if the selected controller can support it.

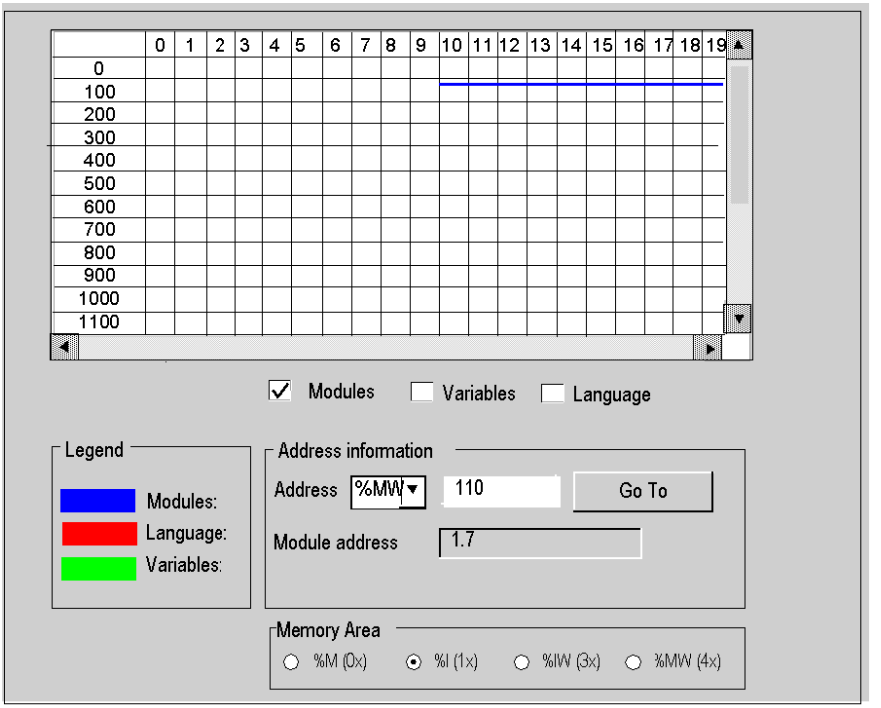
Memory Management

This area of the dialog box allows you to define the type of addressing (topological or mixed addressing).

For mixed addressing it contains a bar graph indicating the size of the **State RAM** memory used in your project in relation to the maximum memory size and a **Viewer** button opening the State RAM memory viewer.

State RAM Memory Viewer

The following screen displays the content of the State RAM memory areas of the M340 controller:



NOTE: The State RAM viewer is directly accessible via the menu:

- **PLC > State Ram Viewer**

Description of the screen zones

Zone	Description
Table	This table represents the mapping of the memory area selected with the **Memory Area** selection buttons. The occupied memory addresses are marked by dashes: • blue: for module type objects • red: for objects entered in the program • green: for objects defined in the variable editor The scroll bars can be used to access the entire memory area.
Modules, **Variables** and **Language** check boxes	These check boxes are used to filter the information displayed in the table. If, for example, **Modules** is the only check box that is selected, only module-type information is displayed.

Zone	Description
Address information	By entering the object and its address in the **Address** fields and clicking on the **Go To** button, you display directly the address in the table (without using the scroll bars). This zone also has a display function; it displays the object and address of the box selected in the table. For module-type objects, it also gives the topological address of the module concerned.
Memory Area selection buttons	These buttons are used to select the memory area to display in the table: • **%M (0x)** output bits and internal bits • **%I (1x)** input bits • **%IW (3x)** input words • **%MW (4x)** output words and internal words

Size of Located Data in Case of State RAM

NOTE: Modicon M340 firmware version 2.4 (and any subsequent supporting version(s)) facilitates the State RAM configuration.

The following table shows maximum and default size of located data in case of State RAM configuration according to the type of controller.

Type of Objects	Address	BMX P34 1000 controller		BMX P34 2000, 20102, 2020, 20302 controllers	
		Maximum Size	Default Size	Maximum Size	Default Size
output bits and internal bits	%M (0x)	32765	752	65530	1504
input bits and internal bits	%I (1x)	32765	752	65530	1504
input words and internal words	%IW (3x)	32765	256	65530	512
output words and internal words	%MW (4x)	32765	256	65530	512

Impact of a Cold Start on %I and %IW Areas

After a Cold Start or %S0 setting, the %I and %IW areas are set to 0 and forcing state is reset.

Controller Modbus Server Uses %I and %IW Areas

- When the State RAM **is** configured in a controller, the Modbus server access to %I area for all input bits operations and to %IW area for all input words operations.

  Base address %SW138 and %SW140 are not used in this case.

  The first object address in a Modbus request (0000) corresponds to %I1 or %IW1.
- When State RAM is *not* configured for a Modicon M340 controller that uses firmware version 2.4 or any subsequent supporting version(s), the behavior of the Modbus server is the same as that in firmware version 2.3.

Base address for %M and %MW in system words %SW139 and %SW141 are used whether State RAM is configured or not.

Protecting Located Data

Introduction

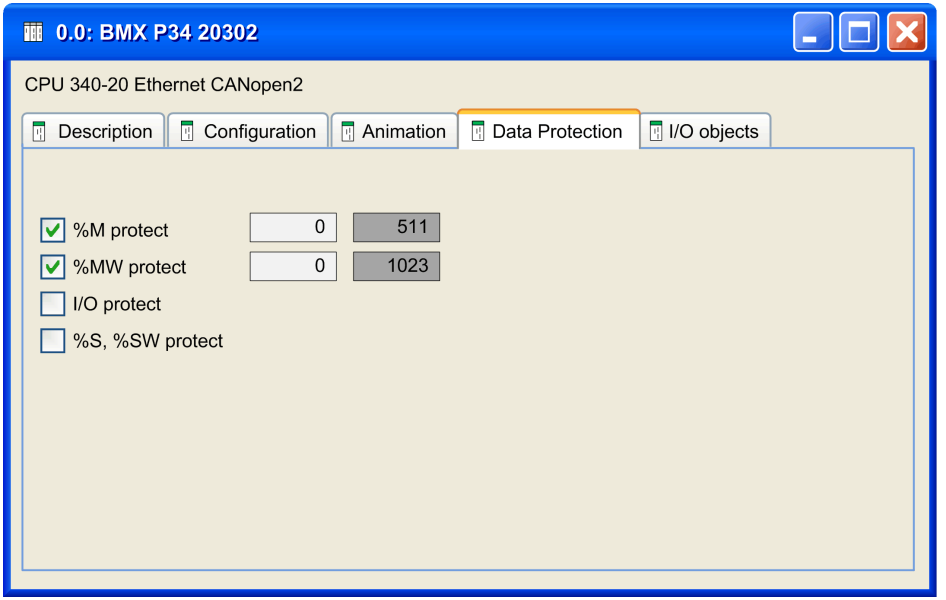
Before any action on the data memory protection, you must activate this feature in your project settings.

In the Control Expert main window, click **Tools > Project Setting > PLC embedded data**. Then select the **Data memory protect** box and click **Apply**.

The data memory protection feature is supported by M340 controllers with firmware version 3.30 or any subsequent supporting version(s). For details, refer to chapter *Data Memory Protection*, page 146.

Procedure of Protecting Located Data

Follow the procedure below to define the located data to protect:

Step	Action	
1	In the Control Expert **Project Browser**, double-click **PLC Bus** to display the main rack. Then double-click on the M580 controller (but not on the Ethernet connectors) to view its properties.	
2	Select the **Data Protection** tab: 	
3	Select the check boxes to enable the data protection:	
	%M protect	The protected area is always located at the end of the %M area. Only the starting address of the protected area can be set. The end address of the protected area is not configurable (grayed). The end address of the protected area equals to n-1 where n is the number of available %M defined by the controller abilities and set in the **Configuration** tab. If **%M protect** is selected, you can enter the starting address or the %M data to protect. By default, the starting address is 0. Clearing the %M protection reset the starting address.
	%MW protect	The protected area is always located at the end of the %M area. Only the starting address of the protected area can be set. The end address of the protected area is not configurable (grayed). The end address of the protected area equals to n-1 where n is the number of available %MW defined by the controller abilities and set in the **Configuration** tab. If **%MW protect** is selected, you can enter the starting address or the %M data to protect. By default, the starting address is 0. Clearing the %MW protection reset the starting address.

Step	Action	
		NOTE: Array variables which are mapped on a %MW range must be entirely inside or entirely outside of the protected %MW range.
	I/O protect	If selected, all I/O objects (including DTM-objects) are protected. **NOTE:** except state RAM objects.
	%S, %SW protect	If selected, all system bits and system words are protected.
4	Select **Edit** > **Validate** (or click the  toolbar button) to save the configuration.	

Configuration of Premium processors

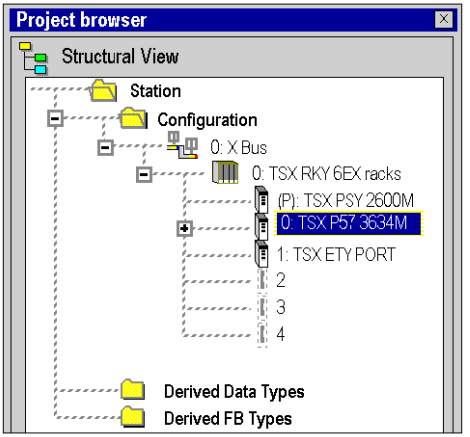
Aim of this sub-section

This sub-section guides the user in the configuration of processor modules for Premium stations.

Configuration of Premium\Atrium processors

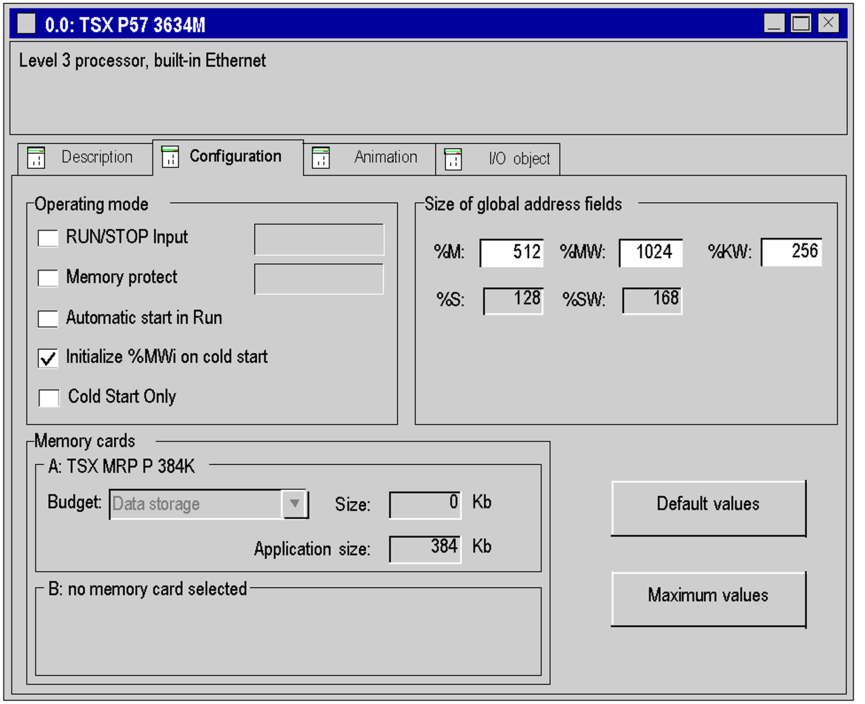
Access to the configuration screen

Proceed with the following steps:

Step	Action
1	Select the processor. Example: 
2	Using the contextual menu, click **Open**.
3	Select the **Configuration** tab.

Configuration of the processor

Configuration Screen:



Proceed with the following steps:

Step	Action
1	If you wish, enable the RUN/STOP input, page 1366.
2	If you wish, enable the Memory Protect. The protection is activated by an input bit. It prohibits the transfer of a project into the PLC and modifications in online mode, regardless of the communication channel. The Run and Stop commands are authorized.
3	If you wish, enable the Automatic start in Run, page 1366.
4	Confirm whether the internal words %MWi, page 1366 are to be initialized (on cold restart, on download, after the insert PCMCIA memory card) or not.
5	If you wish, enable the Cold Start Only, page 1367 feature.

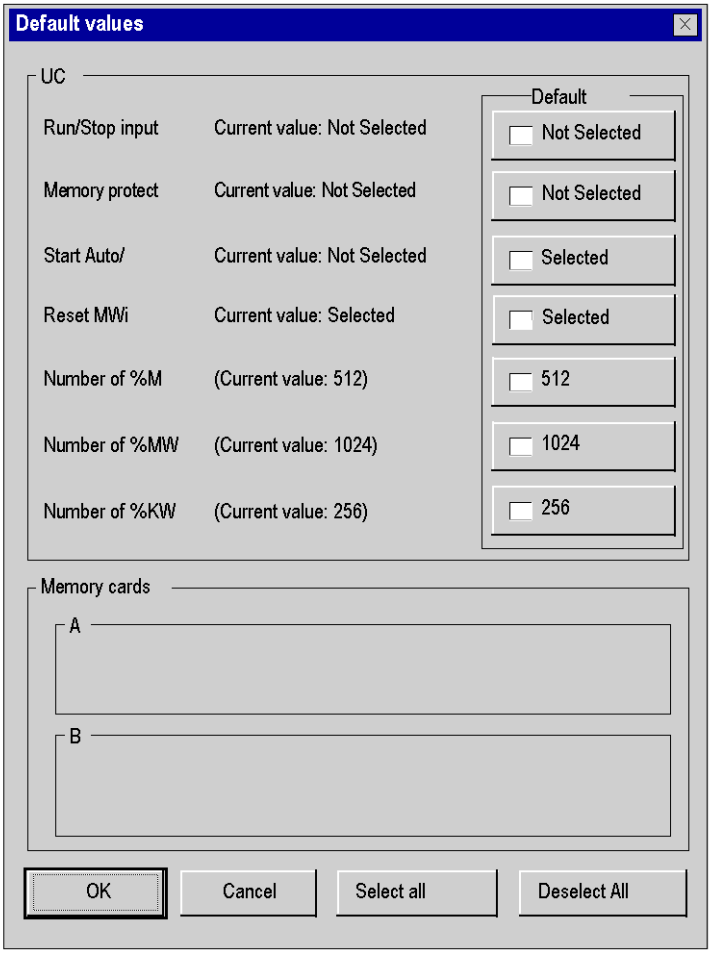
Step	Action
6	Define for the application: (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) • the number of internal bits %M, • the number of internal words %MW, • the number of constants %KW, The bits and system words are fixed by the manufacturer.
7	The **Memory cards** field is completed according to the PCMCIA memory card selected, page 1368. The data storage area is only used by EFBs such as `READ_U_PCMCIA` and `WRITE_U_PCMCIA`. The application area is used for the program. However, some parts of the program (located and unlocated variables, FB instances) are always present in the CPU internal memory, page 139.

Pre-set values

To select the:

- **default** values, press the **Default values** button,
- **maximum** values, press the **Maximum values** button,

Window displayed by pressing the **Default values** button:



Use of the command buttons:

Button	Role
Deselect All	Is used to invalidate all the default values posted on the right of the check boxes if they have been selected.
Select All	Is used to select all the default values posted on the right of the check boxes.
Cancel	Is used to exit.
OK	Is used to exit taking the values into account.

NOTE: With the **Maximum values** button, the window displayed is the same, only the values on the right of the check boxes are different.

RUN/STOP input

The input **%Ir.m.c** can be parameterized to switch the PLC to **RUN/STOP** mode in the following way:

- %Ir.m.c to 1 -> the PLC switches to RUN (execution of the program),
- %Ir.m.cl to 0 -> the PLC switches to STOP mode (stop program execution).

NOTE: A STOP command by the input %Ir.m.c is priority compared to a RUN operation by terminal or by network command.

An error on the RUN/STOP input causes a switch to STOP.

Memory protect

The input **%Ir.m.c** can be parameterized to protect the internal application ram and the memory card in the following way:

- %Ir.m.c to 0 -> the internal application and the memory card are not protected,
- %Ir.m.c to 1 -> the internal application and the memory card are protected.

NOTE: If the input is in error, %Ir.m.c is considered at 1 (memory is protected). To remove this protection in this configuration screen, the input should not be in error.

Automatic start in Run

The enabling of this option automatically changes the PLC to RUN mode at the time of a cold start (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).

Two types of start:

- in the absence of a PCMCIA memory card, the PLC starts on the contents of the internal RAM of the processor,
- in the presence of a PCMCIA memory card it is its content which fixes the start.

Initialize %MWi

On cold start (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual), on download or after the insert of PCMCIA memory card:

Depending of the flag:

		At the end of the transfer : ==> The initial values are assigned to the variables.	On setting of %S0 : ==> The initial values are assigned to the variables.
Not checked	OS >= V2.3	**NO**	**NO**
	OS < V2.3	**YES**	**NO**
Checked	OS >= V2.3	**YES**	**YES**
	OS < V2.3	**YES**	**YES**

the %MWi values will be initialized to 0 or initial value even if the box is unchecked,

- if you change the %MW area on the field **Size of global address fields**,
- if after a power restoration, the OS detects that the contents of the internal RAM has not been saved (PLC powered off and Back-up battery non-operational).

Cold Start Only

If checked, this option forces the cold start (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) of the application, instead of the normal warm start (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).

By default, the Cold Start Only option is unchecked.

The Cold Start Only option is only supported since PLC V2.7.

An application using this functionality will not be:

- downloadable on a PLC with a previous version,
- executable on a PLC with a previous version,
- usable with Unity Pro V4.0 or lower.

  Unity Pro is the former name of Control Expert for version 13.1 or earlier.

  NOTE: The Cold Start Only check box is present only if the current selected PLC can support it.

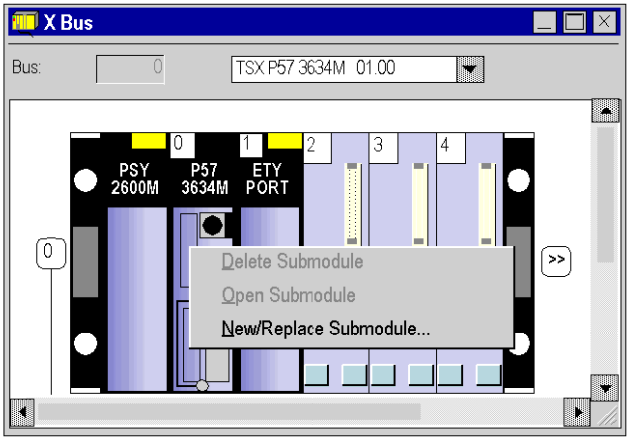
Access to PCMCIA memory cards of Premium processors

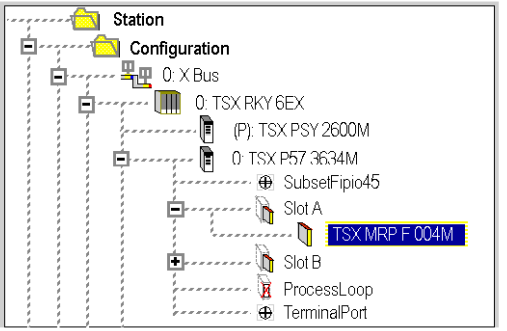
Introduction

This operation describes how to choose a PCMCIA memory card.

Selection of a PCMCIA card

Carry out the following actions:

Step	Action
1	Access the rack configuration screen.
2	Select the PCMCIA location and from the popup menu execute the **New/Replace Submodule...** command. Example: 

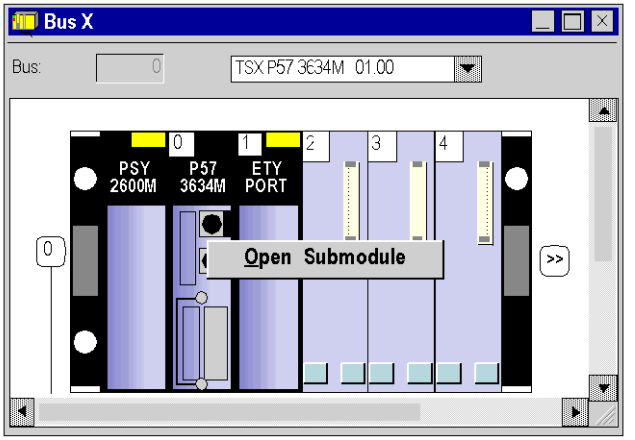
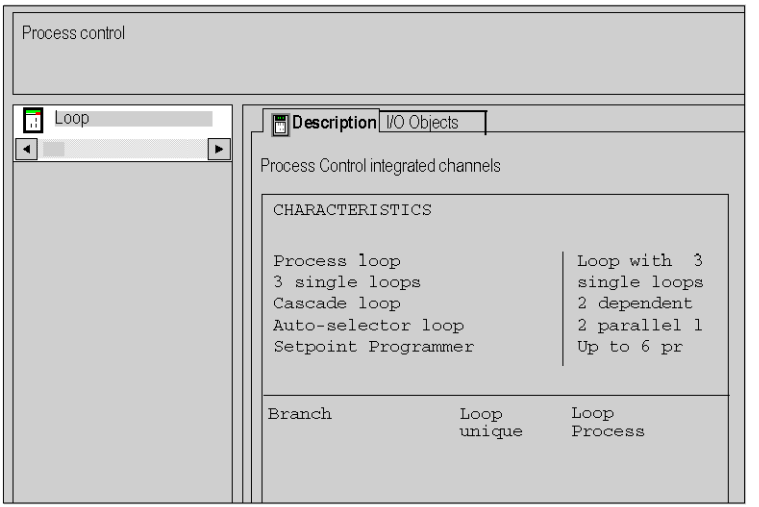
Step	Action
3	Select the PCMCIA family, and the card within the family.
4	Confirm with **OK** The card is added to the directory tree of the processor: 

To delete a PCMCIA memory card, select its location and from the popup menu execute the **Delete Submodule** command.

Access to control loop configuration for Premium processors

Instructions

Carry out the following actions:

Step	Action
1	Access the rack configuration screen.
2	Select the location of the process control loops and from the popup menu execute the **Open Subobject** command. Example:  The following screen is displayed: 
3	Configure the process control loops (see Premium and Atrium using EcoStruxure™ Control Expert, Process Control, User Manual).

Configuration of the Quantum processors

Aim of this sub-section

This sub-section guides the user in the configuration of the processor modules for the Quantum stations.

Configuration of Quantum Controllers

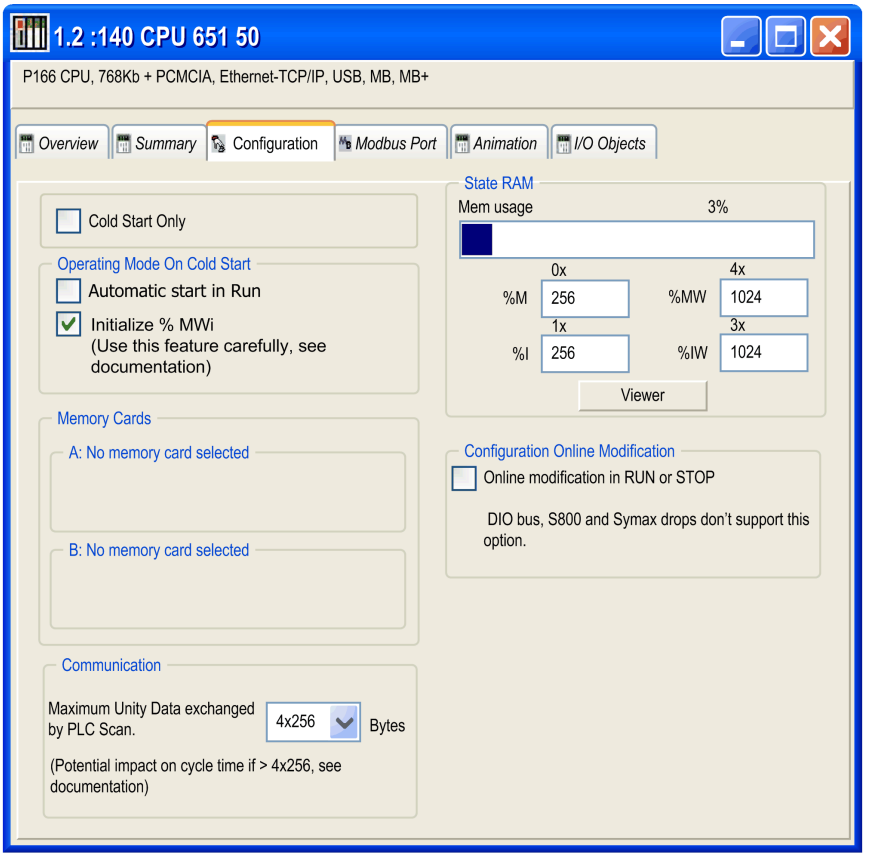
Accessing the Configuration Screen

Carry out the following actions:

Step	Action
1	Access the rack configuration screen.
2	Select the controller.
3	Using the contextual menu, click **Open Module**.
4	Select the **Configuration** tab.

Configuring the Controller

The following figure shows an example of the Quantum controller configuration screen:



Description of the Configuration Screen:

Item	Option	Value	Description
Cold Start Only		x	If you wish, enable the Cold Start Only, page 1373 feature.
Operating Mode On Cold Start	Automatic start in Run	x	Determines the operating condition during Cold Start
	Initialize `%MWi`	x	
Memory Cards	A:	N/A	Displays the configuration in the PCMCIA Slots
	B:	N/A	
Communication	By default, the bandwidth is 4x256 bytes, supported by the OS versions prior		The maximum data volume exchanged each cycle between the NOE and controller modules.

Item	Option	Value	Description
	to V2.80 for the controller and V4.60 for the NOE.		
	For Quantum controllers: • 140 CPU 311 10 • 140 CPU 534 14 • 140 CPU 434 12	4x256 4x1024	
	For Quantum controllers: • 140 CPU 651 50 • 140 CPU 651 60 • 140 CPU 652 60 • 140 CPU 670 60 • 140 CPU 671 60 • 140 CPU 672 60 • 140 CPU 672 61 • 140 CPU 678 61	4x256 4x1024 8x1024 12x1024	
State RAM	Mem usage	1.	A bar displays percent of memory used.
	%M-0x	2.	Size of the different memory areas **Note:** The values for %IW and %MW have to be divisible by 8.
	%MW-4x	2.	
	%I-1x	2.	
	%IW-3x	2.	
	Viewer	N/A	Opens the State RAM Viewer tab, which displays the allocation of used memory. (See the illustration following.)
Configuration Online Modification	Online modification in RUN or STOP	x	This check box allows you to: • Add or delete discrete or analog modules, • Modify parameters **NOTE:** These modifications can be done in RUN.
1. The value (expressed as a percentage and displayed on the scale) depends on the memory usage of the Hot Standby configuration. 2. Enter the appropriate values. All values depend on Hot Standby configuration.			

Cold Start Only

If selected, this option forces the cold start (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) of the application, instead of the normal warm

start (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).

By default, the Cold Start Only option is cleared.

The Cold Start Only option is only supported on High End controllers since V2.7.

An application using this functionality will not be:

- downloadable on a controller with a previous version,
- executable on a controller with a previous version,
- usable with Unity Pro V4.0 or lower.

  Unity Pro is the former name of Control Expert for version 13.1 or earlier.

  NOTE: The Cold Start Only check box is present only if the selected controller can support it.

Automatic Start in RUN

Enabling this option automatically changes the controller to Run mode (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual)on cold start.

Two types of start:

- in the absence of a PCMCIA memory card, the controller starts on the contents of its internal RAM,
- in the presence of a PCMCIA memory card it is its content which fixes the start.

With the **Automatic start in RUN** option enabled, the following events will trigger the run of the application on cold start:

- Inserting the PCMCIA card when the controller is powered
- Replacing the controller while powered
- Unintentional use of the reset button
- Powering up a controller with an inoperable battery after a power outage

⚠ WARNING
UNINTENDED COLD START OF THE CONTROLLER Use the STOP input (on Premium controllers) or use the switch on the front panel of the controller (for Quantum controllers) to prevent automatic RUN of the controller. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

Initialize %MWi

On application download:

- if you select the check box, the %MWi values will be re initialized or set to 0,
- if you clear the check box, the %MWi values will set to 0.

On cold start or after inserting the PCMCIA memory card:

- if you select the check box, the %MWi values will be re initialized or set to 0,
- if you clear the check box, the %MWi values will retain their current value.

Communication

When the protocol under TCP/IP is used (OPC Factory Server or EcoStruxure Control Expert), it is possible to configure the maximum volume of data that can be exchanged each cycle between the controller and the NOE modules using the **Maximum Unity Data exchanged by Plc Scan** option.

This functionality is supported only by controller modules as of OS version 2.80 and NOE modules as of OS version 4.60.

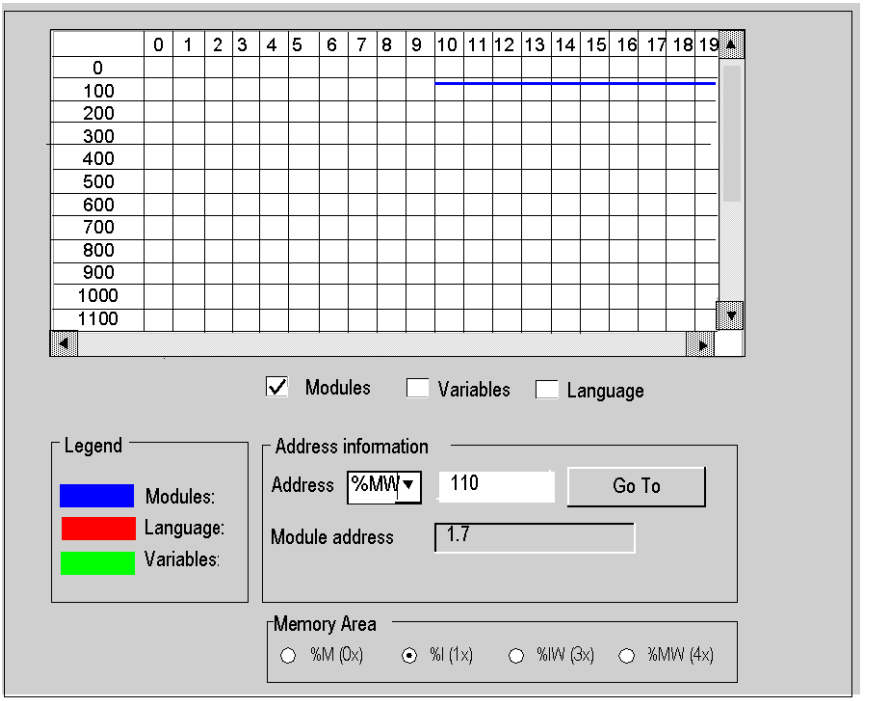
Increasing this bandwidth has an impact on the cycle time of the controller (2ms per kbytes exchanged). This impact is proportional to the amount of data actually exchanged and not the bandwidth configured. So, if the channel is set to the maximum, but not used, the impact on cycle time will be negligible.

State RAM Memory

The **State RAM** bar chart allows you to know the size of the **State RAM** memory used in your project in relation to the maximum memory size.

State RAM Memory Viewer

The following screen displays the content of the State RAM memory areas of the Quantum controller:



NOTE: The state ram viewer can be directly accessible via the menu:

- **PLC > State Ram Viewer**

Description of the screen zones

Zone	Description
Table	This table represents the mapping of the memory area selected in the **Memory Area** field. The occupied memory addresses are marked by dashes: • blue: for module type objects • red: for objects entered in the program • green: for objects defined in the variable editor The scroll bars can be used to access the entire memory area.
Modules, **Variables** and **Language** check boxes	These check boxes are used to filter the information displayed in the table. If, for example, only 'module' is selected, only module-type information is displayed.

Zone	Description
Address information	By entering the object and its address in the **Address** fields and clicking the **Go to** button, you display directly the address in the table (without using the scroll bars). This zone also has a display function; it displays the object and address of the box selected in the table. For module-type objects, it also gives the topological address of the module concerned.
Memory Area selection buttons	These buttons are used to select the memory area to display in the table: • **%M (0x)** output module bits and internal bits • **%I (1x)** internal module bits • **%IW (3x)** input/output register words and internal words • **%MW (4x)** I/O module output words and internal words

Configuration Online Modification

On controllers supporting this functionality, a check box is activated and appears in the controller Editor tab.

If the **Online modification in RUN or STOP** check box is selected, the user can do online RUN modification like addition / deletion, and modification of the module parameters.

Configuration of Modbus ports on Quantum

Assignment of Modbus ports

The Modbus port configuration screen can be used to select the Modbus communication parameters for 3 ports: ports No. 1, 2 and 3.

Port No. 1 is assigned:

- to the built-in Modbus port for 140 CPU 6•• processors
- to the first built-in Modbus port (MODBUS COMM1) for other Quantum processors

Port No. 2 is assigned:

- either to the second built-in port (MODBUS COMM2), for processors with 2 built-in ports
- or to one or several NOM modules.

Port No. 3 is assigned: to one or several NOM modules

Rule for assignment of ports 2 and 3 to NOM modules:

Starting with the lowest slot numbers:

- the first NOM module will be assigned to port 2,
- the second will be assigned to port 3,

- the third will be assigned to port 2,
- the fourth will be assigned to port 3,
- the fifth will be assigned to port 2,
- the sixth will be assigned to port 3.

Note: It is possible to change the assignment order of 2 of them by entering

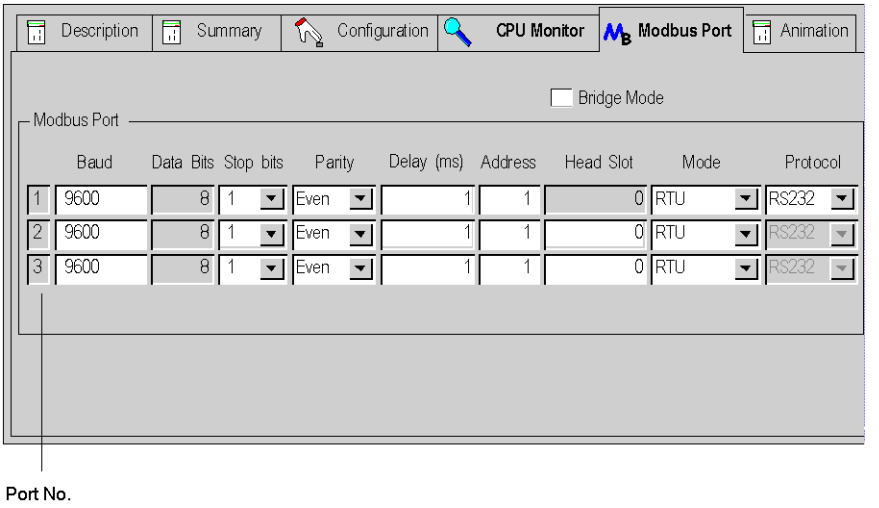
their slot number in the **Head Slot** column (see table below).

Important:

The port parameters are only taken into account if the switches on the front panels of Quantum 140 CPU 3••/4••/5•• processors (used with first built-in port MODBUS COMM1) and of NOM modules are in the **Mem** position.

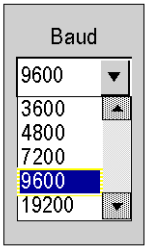
Access to the configuration editor

Carry out the following actions:

Step	Action
1	Select the processor and then select the **Open module** command from the popup menu.
2	Select the **Modbus Port** tab. **Result**: The following screen appears: 

Presentation of configuration parameters

Carry out the following actions:

Step	Action
1	Choose the operating mode: **Bridge Mode** checkbox
2	Choose the transmission speed: 
3	Choose the number of Stop bits: 
4	Choose the parity: 
5	Choose the Delay which corresponds to the minimum time tolerated between two frames, also called delay between frames.
6	Choose the Modbus port slave address.
7	By default, if the **Head Slot** fields are left set to 0, the NOM modules observe the normal assignment rule described above. Entering the slot number of a NOM module assigns it to the corresponding port (the assignment rule is no longer applied for this module but continues to apply to other modules).

Step	Action
8	Choose the communication mode: 
9	Choose the communication protocol: 

Configuration of the input\output modules for a Premium\Atrium, Quantum or Modicon M340 station

Subject of this Section

This section guides the user through the configuration procedure for input\output modules dedicated to a Premium\Atrium, Quantum or Modicon M340 station.

Access to the Modicon, Premium or Atrium I/O Module Configuration Editor

Procedure

Carry out the following actions:

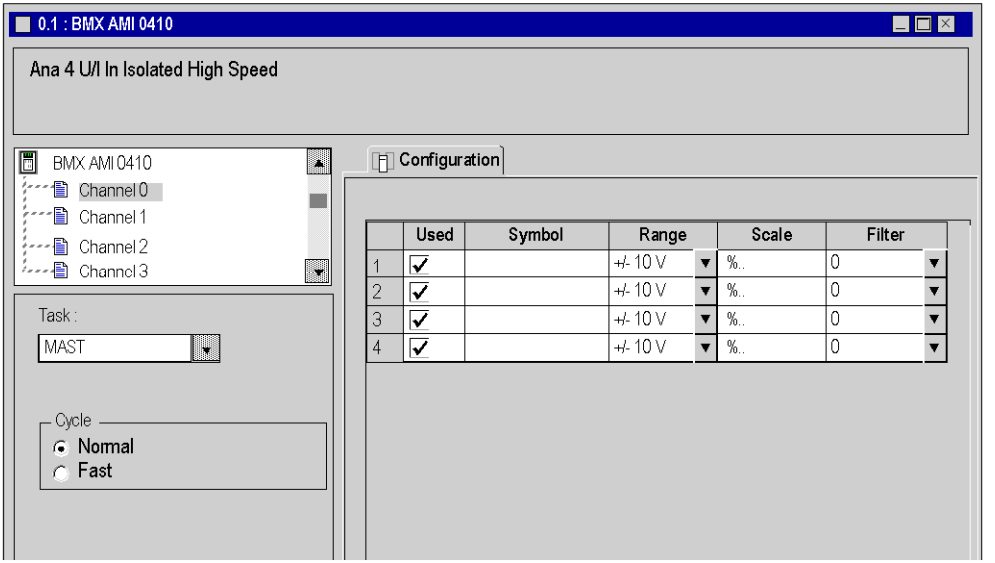
Step	Action
1	With the mouse, select the module to be configured.
2	Using the contextual menu, click **Open Module**.

Configuration in Offline Mode

The configuration window is an editor from which you configure the module, some of your entries are checked, and in some cases you are assisted.

To complete the various fields, **refer to the application-specific manual corresponding to the module** in which you will find the procedure and information necessary for the configuration.

Screen example for a discrete module:



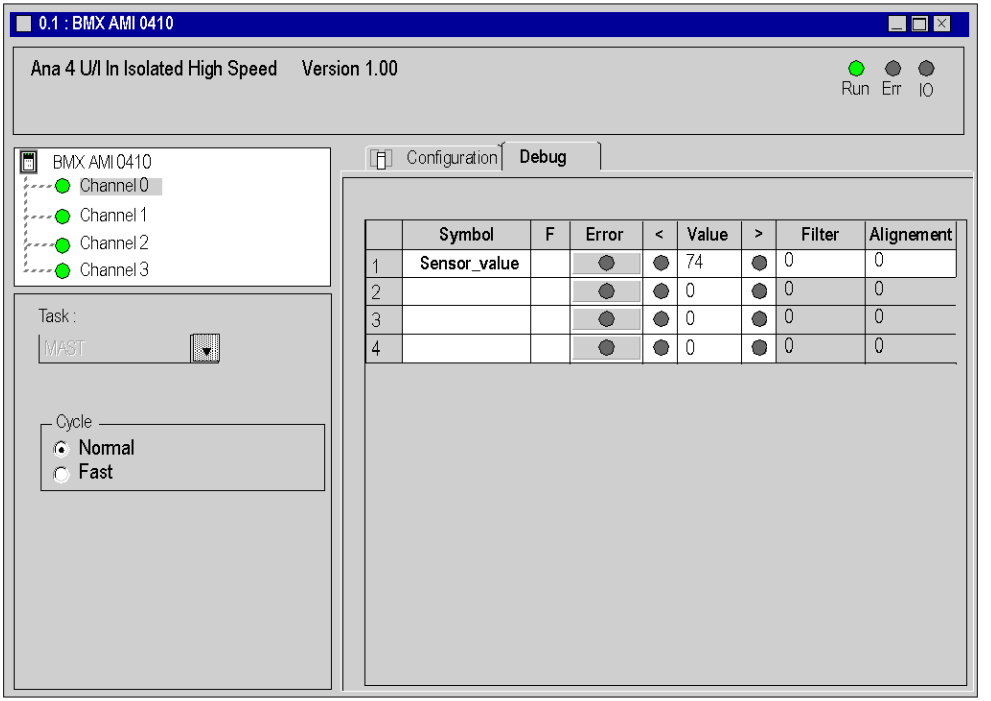
Configuration in Online Mode

In online mode two additional tabs are accessible:

- The **Debug** tab, used to:
  - display the current values of the module data,
  - modify/force the data values,
  - diagnose the erroneous channels.
- The **Fault** tab, used to display errors occurring in the module.

  NOTE: The content of these tabs is described in the **"Debugging and adjustment"** chapter of the corresponding application-specific manuals.

Screen example for a counting module:



Access the Quantum Controller Input/Output Module Configuration Editor

Instructions

Proceed with the following steps:

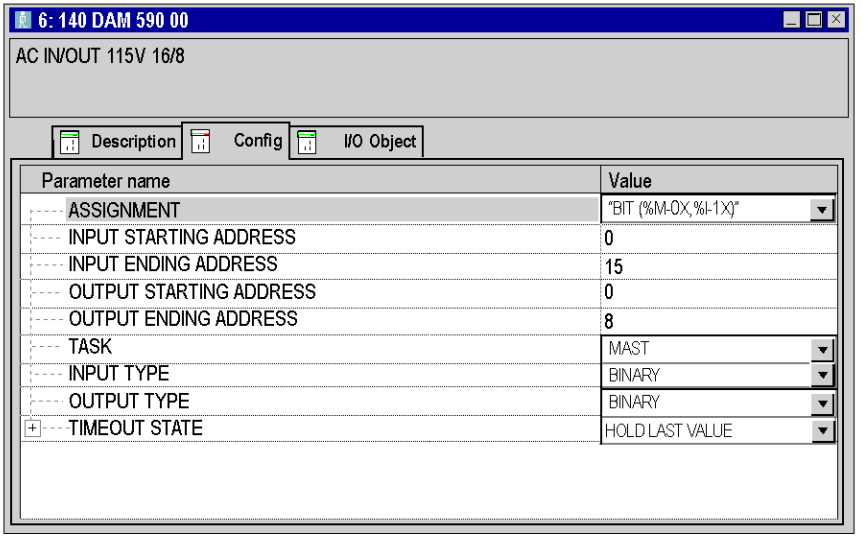
Step	Action
1	With the mouse, select the module to be configured.
2	Using the contextual menu, click **Open Module**.

Configuration in Offline Mode

The configuration window is an editor from which you configure the module, some of your entries are checked, and in some cases you are assisted.

To complete the various fields, **refer to the application-specific manual corresponding to the module** in which you will find the procedure and information necessary for the configuration.

Screen example for a discrete module



Configuration in Online Mode

In online mode, for certain modules, two additional tabs are accessible:

- The **Debug** tab, used to:
  - display the current values of the module data,
  - modify/force the data values,
  - diagnose the erroneous channels.
- the **Fault** tab which is used to display errors occurring in the module.

  NOTE: The content of these tabs is described in the corresponding application-specific manuals.

General functionalities of the editor

What's in This Chapter

Analysis of the configuration 1385
Export\Import of the configuration..... 1386
Printing the configuration 1386

Aim of this section

This section describes the general functions available in the configuration editor.

Analysis of the configuration

At a Glance

The configuration analysis is not a separate functionality, it is started during the global analysis of the project.

The configuration analysis checks:

- that all the declared modules are compatible,
- that the maximum number of one type of module in the configuration is not exceeded,
- that the limits of the various types of inputs/outputs managed are not exceeded
- that all the inputs/outputs declared and used in the application are properly configured.

How to start the global analysis of the application

Carry out the following actions:

Step	Action
1	Select the **Build -> Analyse project** command from the menu.
2	Check the result of the analysis in the information window at the bottom of the screen.

Export\Import of the configuration

Export

The description of the procedure is given in the section "Import/Export, page 1696".

Import

The description of the procedure is given in the section "Import/Export, page 1698".

Import SIS

This Import SIS command ensures the gateway with the SIS Automation software version greater than or equal to Schneider Electric 3.0.

This import ensures the automatic generation of the configuration: racks, processor, input/output modules... from the .XML file resulting from an export made by this tool.

The description of the procedure is given in the section "Import/Export, page 1698".

Printing the configuration

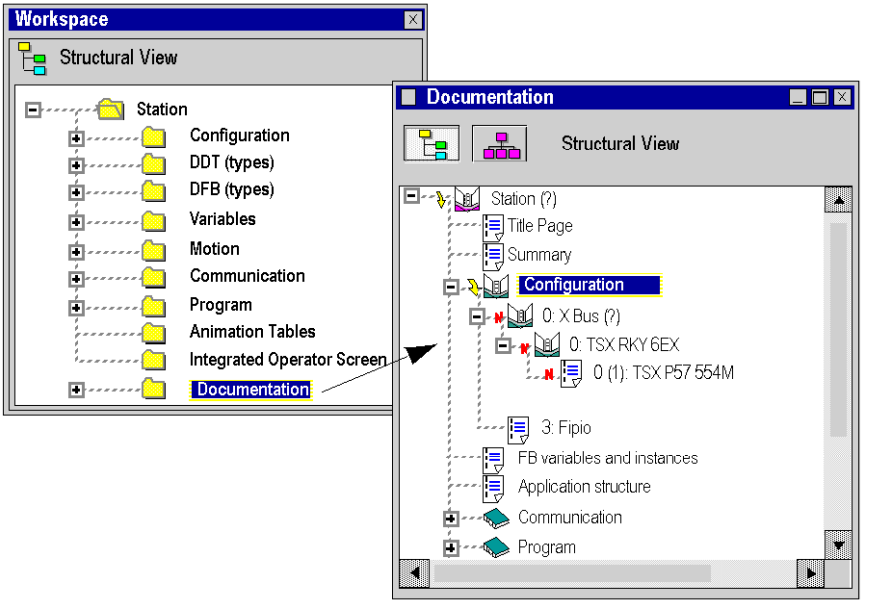
Introduction

The programming software allows you to generate and print the complete project documentation file. You can choose to print the configuration alone from this file.

In the project documentation file, the configuration part is made up:

- of the bus editor,
- of the input/output module editor.

The documentation file can be accessed from the project browser in the **"Documentation"** directory.



How to print

A description of the procedure is given in the chapter on "Documentation, page 1660"

Debugging and adjustment

What's in This Part

Debugging the program 1390
Viewing and adjusting variables 1466
Machine-Process Simulator 1508

Subject of this Part

This part describes the debug and adjustment functions and tools offered by Control Expert: application debug, variable adjustment, simulator, etc.

Debugging the program

What's in This Chapter

Safety precautions 1390
Program Animation 1397
Debugging in Ladder Language 1413
Debugging in Textual Languages (Structured Text, Instruction List) 1421
Debugging in Function Block Diagram (FBD) Language 1430
Debugging User Function Block (DFB) Instances 1439
Debugging in Sequential Function Chart Language (SFC) 1443
Information on Debugging 1450
PLC debug screen 1453

Subject of this Chapter

This chapter guides the user through how to debug the program depending on the language used in its component sections:

- Ladder language,
- textual languages (Structured Text or Instruction List),
- Function Block Diagram language,
- Sequential Function Chart language.

Safety precautions

Subject of this Section

This section presents the different safety precautions to respect when debugging a project with Control Expert.

Modification in RUN

At a Glance

This functionality is used to modify the controller program where its inputs/outputs control an industrial process.

RUN Modification

Modifications in In RUN mode are immediately applied to the process.

⚠WARNING
UNINTENDED EQUIPMENT OPERATION When using the in RUN modification function, ensure that the modifications will not have adverse effects on the running process. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

Switch to STOP of a Controller in Debug mode

At a Glance

You can switch a controller to STOP using:

- the command **PLC > STOP** when running Control Expert
- the RUN/STOP input for Modicon M340, Modicon M580, and Premium/Atrium controllers
- the switch on the front panel of Quantum controllers (except 140CPU6••)
- the LCD and buttons for Quantum controller 140CPU6••

Precaution

In all the above cases, the STOP is only effective at the end of the MAST task cycle.

⚠WARNING
UNINTENDED EQUIPMENT OPERATION In Debug mode, verify that neither any breakpoint nor the Step by Step mode are active before switching the controller to STOP. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

NOTE: When the controller is in STOP, all tasks are in STOP. Verify the behavior of the corresponding Inputs/Outputs.

Setting a Task to STOP

At a Glance

A task can be set to STOP in different ways:

- Use of a Breakpoint or the Step by Step mode.
- Use of the **PLC screen** , page 1454 (controller animation screen).

Precautions

When a task is set to STOP, Inputs/Outputs behave differently to whena task is deactivated, page 1393.

The behavior of outputs differs depending on the type of controller:

- **Modicon M580, M340 and Premium / Atrium**: when a task is set to STOP, the associated outputs automatically switch to the configured mode (fallback or maintain). The outputs associated with the other tasks continue to be updated as normal.
- **Quantum**: when a task is set to STOP, the associated outputs continue to be updated with the value that preceded the STOP action. The outputs associated with the other tasks continue to be updated as normal.

  NOTE: When a task is set to STOP with a Quantum controller, the outputs continue to be updated by the controller. In two cases, this can lead to unforeseen behavior:

  - when inputs are also updated by another task that is still in RUN
  - when Inputs/Outputs are mapped in the same memory zone, and associated with different tasks.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

When a task is set to STOP with a Quantum controller, ensure that no other running task is managing the same outputs.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Task Deactivation

At a Glance

A task can be deactivated in different ways:

- By using system bits (%S30 to %S35 (see *EcoStruxure™ Control Expert, System Bits and Words, Reference Manual*)).
- By using the **PLC screen** , page 1454 (controller animation screen).

Precautions

When a task is deactivated, Inputs/Outputs behave differently to when a task is set to STOP., page 1392

The behavior of outputs differs depending on the type of controller:

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

When deactivating a task, take into account the fact that Inputs and Outputs continue to be active.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Deactivation of Inputs/Outputs

At a Glance

The inputs associated with a task can be deactivated using the system word (see *EcoStruxure™ Control Expert, System Bits and Words, Reference Manual*).

The outputs associated with a task can be deactivated using the system word (see *EcoStruxure™ Control Expert, System Bits and Words, Reference Manual*).

Precautions

NOTE: Deactivating the Inputs/Outputs does not deactivate the task, page 1393

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

Before deactivating an Output, take into account its deactivation mode (fallback or maintain).

Failure to follow these instructions can result in death, serious injury, or equipment damage.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

Before setting the %SW9 value to 1, ensure that the output behavior remains appropriate.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

On M580, the remote and distributed outputs are not assigned by the system word %SW9.

On M340, the inputs/outputs distributed via CANopen bus are not assigned by the system word %SW9.

On Premium/Atrium, module outputs located on the X Bus automatically switch to the configured mode (fallback or maintain). On the Fipio bus, certain devices do not manage fallback mode; then only maintain mode is possible.

On Quantum, all outputs, as well as the local or remote rack (RIO) are maintained in the state that preceded the switch to 1 of the %SW9 bit corresponding to the task.

NOTE: The Distributed Inputs/Outputs (DIO) are not assigned by the system word % SW9.

Use of the Step by Step Mode and Breakpoint

At a Glance

The use of the Step by Step mode and Breakpoint influence the controller operating modes. You must be careful when using them.

Precautions

In Debug mode, the use of a breakpoint (except for an SFC breakpoint, page 1443) or of the Step by Step mode forces the task to STOP.

⚠WARNING
UNINTENDED EQUIPMENT OPERATION Check the consequences on the operation of the outputs when the task is set to STOP. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

For more information, refer to setting a task to STOP, page 1392.

Unwanted Disconnection in Debug Mode for Premium Controllers

Precaution

If the connection is interrupted between the EcoStruxure Control Expert programming software and the Premium controller, the breakpoint is automatically deleted. If the Premium controller was stopped at breakpoint level, it automatically switches to STOP after the disconnection Timeout.

⚠WARNING
UNINTENDED EQUIPMENT OPERATION With a Premium controller in debug mode, ensure that the consequences of a controller STOP on the application are acceptable. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

NOTE: When the controller is in STOP, all tasks are in STOP. Verify the operation of the corresponding Inputs/Outputs.

Debug Screen Via a Fipway, Fipio or Uni-Telway Network with a PCMCIA Card

Restriction

If you use debug screens to access a PLC via a Fipway network, a Fipio fieldbus or a Uni-Telway bus with a PCMCIA card, you may, on certain modules, obtain a message explaining that it is not possible to access debugging.

The solution is to connect directly to the PLC with the Uni-Telway, USB or XIP link (depending on the configuration of the hardware used).

Behavior of Forced Variables Between Modsoft/NxT/Concept and EcoStruxure Control Expert

Forced Variable Behavior

The behavior of forced variables has been changed between Modsoft/NxT/Concept and EcoStruxure Control Expert.

With Modsoft/NxT/Concept you cannot force variables (0x, 1X) with the Quantum controller memory protect switch in the ON position.

With EcoStruxure Control Expert you can force variables (%M, %I, %Q) with the Quantum controller memory protect switch in the ON position.

When using a basic Quantum controller, if the memory protect switch is in the OFF position, some variables are forced and the application is saved in flash memory:

- with Modsoft/NxT/Concept, forced variables keep their state after a cold start,
- with EcoStruxure Control Expert, forced variables lose their state after a cold start.

⚠WARNING
UNINTENDED EQUIPMENT OPERATION Verify the forced variables and memory protection switch when shifting between Modsoft/ NxT/Concept and EcoStruxure Control Expert. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

Warning Message when Forcing Variables

A warning message will display in EcoStruxure Control Expert each time that variables are forced on the following Quantum controllers:

- 140 CPU 311 10
- 140 CPU 434 12A/U
- 140 CPU 534 14A/U

This warning message includes a "Do not warn me again" option so that, when selected, the message will not be redisplayed during the current EcoStruxure Control Expert session. Starting a new session of EcoStruxure Control Expert means that the option must be checked once more in order to no longer see it when forcing a variable.

Program Animation

Subject of this Section

This section describes the animation used in the different language editors.

Introduction to animation

General

Program **animation** is possible if the project that has been opened using the software is identical to the project in the PLC.

The two connection modes are as follows:

- **integral online mode** (projects identical),
- **degraded online mode** (projects different).

In integral online mode, as long as it is authorized for your user profile and the **Use Programming mode** option is checked (**Tools ->Options** menu then **Connection** tab) you can:

- debug the project (setting a breakpoint or a watchpoint),
- modify the project, in this case:
  - either you generate the project, in which case the modifications are automatically transferred to the PLC and animation resumes,
  - or you do not generate the project, in which case animation of the modified sections is stopped.

NOTE: Modification and debugging of the project are not permitted in degraded mode, or in integral online mode with the **"Use monitoring mode"** option checked ("Tools ->Options" menu then "Connection" tab).

NOTE: In Monitoring mode links between EFs are not animated.

Animation is managed section by section. You can stop it or restart it section by section.

NOTE: The screen is refreshed immediately after the execution of each section. As a result, for variables which change very quickly it is possible that there may be differences from time to time between what is displayed and the actual values in the PLC.

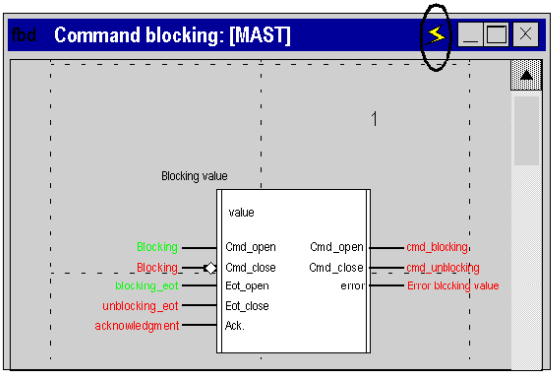
NOTE: For all languages, the animation of strings is limited to 16 characters no matter what the string size is. In order to animate more than 16 characters use the  button in the toolbar.

Specific Features of Animation

Two types of animation are available:

- **standard** animation, in which the variables of the active section are refreshed upon completion of the master task (MAST),
- **synchronized** animation in which the variables of the active section are refreshed at the same time as a program element (of the section) containing the **Watchpoint** (useful if one of the variables is used in several program sections and you want to know its value at a specific location)

Example of an animated section with Watchpoint:

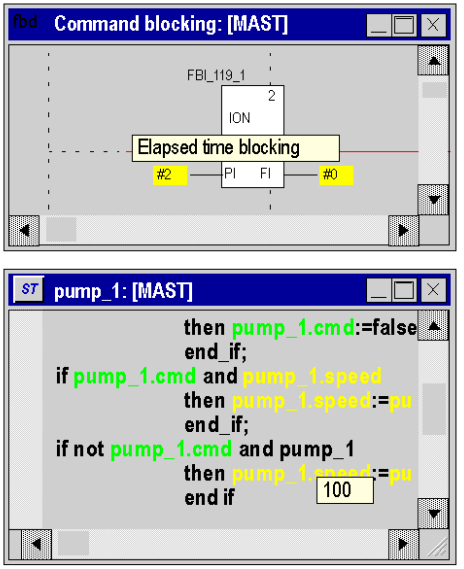


Depending on which language editor is open, it may not be possible to view the name of the variable at the same time as it content. The **Tool tip** function is provided to compensate for this.

This is a help bubble which is displayed when you move the cursor over the variables. This help bubble information is then displayed:

- the value of the variable if only its name is visible in the editor,
- the type, name, address and comment of the variable if only its value is visible in the editor.

Examples of help bubbles in an FBD editor and in a Structured Text editor



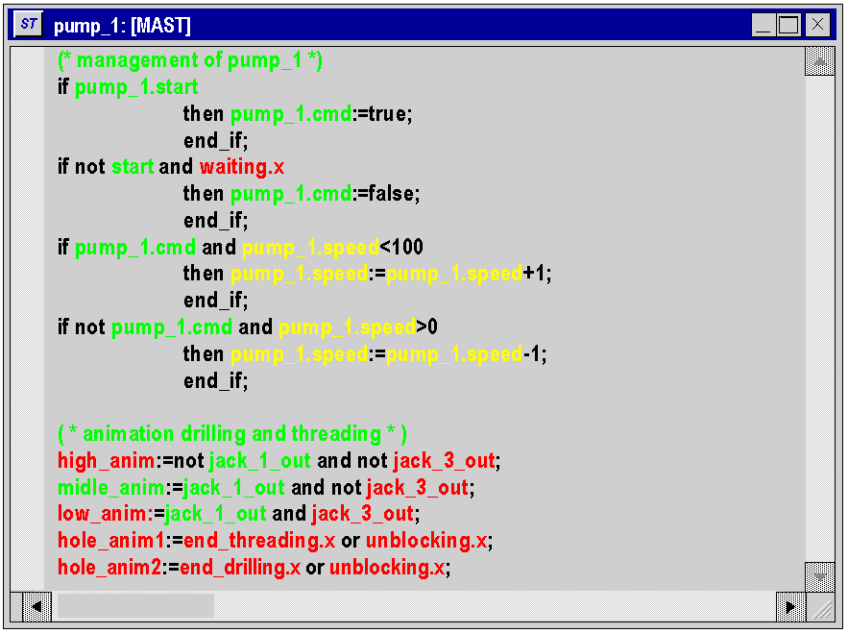
Starting and Stopping Animation

By default, the sections are animated. To stop the animation of a section click on the  button in the toolbar. To restart the animation, click the button a second time.

Section in textual languages (Structured Text or Instruction List)

Animation colors

Animation of a Section in Structured Text language:



The background color of the editor is **gray**.

All text is shown in **black**.

All comments are shown in **green**.

Black color used in offline mode for variables and instructions is replaced.

Variables and instructions during animation are shown in the following colors:

- for Boolean variables:
  - **green** if the variable is TRUE (1)
  - **red** if the variable is FALSE (0)
- **yellow** for most types of variable

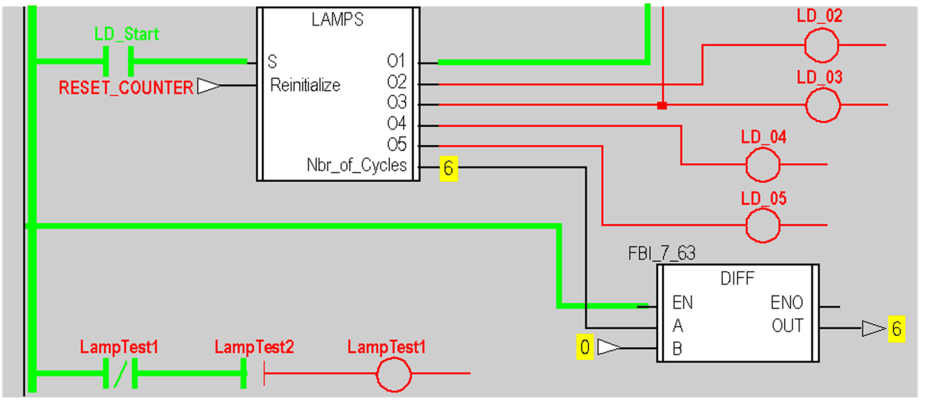
NOTE: Data types **other than Booleans**, are animated in a display window and not in the editor. However, it is possible to view their value in a help bubble (Tool tip) when you position the mouse pointer over the value.

NOTE: Multi-dimension array variable are not animated in the language editors and are not shown in yellow.

Section in a Ladder Diagram language

At a Glance

Animation of a section in Ladder Diagram language:



The background color of the editor is **gray**.

The colors used in offline mode for the various elements (instructions, variables, comments) are replaced:

- for Boolean variables:
  - **green** if the variable is TRUE (1),
  - **red** if the variable is FALSE (0).
- **yellow** for numerical types of variable.

  NOTE: Derived data **(DDT)** instances and function block **(EFB\DFB)** instances are animated in a display window and not in the language editor. You can view type, name, address and comment of a numerical variable (yellow) in a help bubble (Tool tip) when you position the mouse pointer over the variable.

NOTE: In a LD section:

- the unconnected parameters of elementary functions are not animated,
- the unconnected parameters of function blocks are animated.

The result of textual expressions is also animated:

- in green and red for Boolean expressions,
- numerical expressions are replaced by their result, the associated help bubble (Tool tip) displays type, name, address and comment of the expression.

Animations

There are 3 types of link:

- Boolean links between contacts and coils,
- Boolean links between function blocks,
- numerical links between function blocks.

Two sorts of animation are possible depending on the selected option:

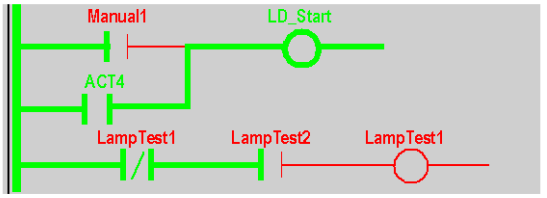
- **with animation of links** for which:
  - Boolean links between contacts and coils are shown in **green** or **red** depending on whether the evaluation of the ladder upstream yields the value TRUE (1) or FALSE (0),
  - Boolean links between function blocks are shown in **green** or **red** depending on whether the output parameter to which they are connected is TRUE (1) or FALSE (0),
  - numerical links between the function blocks are shown in **black**. They display the value of the output parameters to which they are connected.
- **without animation of links** in which closed contacts and triggered coils are shown highlighted.

To select an animation, perform the following actions:

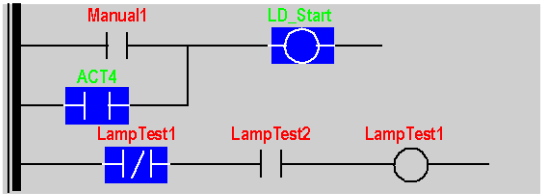
Step	Action
1	From the menu, select the **Tools->Project Settings...** command. **Result**: The **Project Settings**, page 551 window is displayed.
2	In the **Code generation** zone either: • check the **Generate with LD link animation** box for animation of links, • uncheck the **Generate with LD link animation** box to disable animation of links.
3	Confirm with **OK**

Example:

With **link animation**



Without link animation

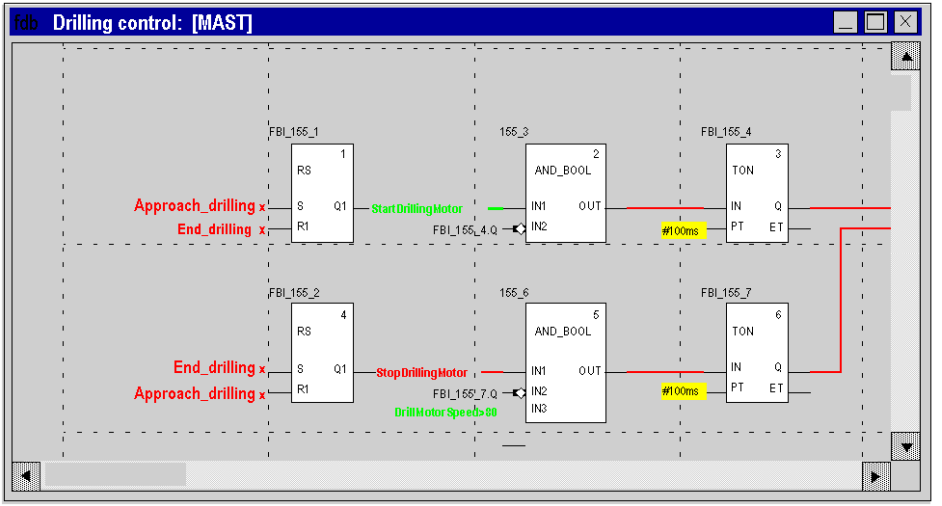


NOTE: When switching from one option to the other the project must be regenerated and reloaded into the PLC. Animation of links has an adverse effect on the performance of the project in run.

Section in Function Block Diagram (FBD) language

At a Glance

Animation of a section in Function Block Diagram language:



The background color of the editor is **gray**.

The colors used in offline mode for the various elements (instructions, variables, comments) are:

- for Boolean variables:
  - **green** if the variable is TRUE (1),
  - **red** if the variable is FALSE (0).
- **yellow** for numerical types of variable.

  NOTE: Derived data **(DDT)** instances and function block **(EFB\DFB)** instances are animated in a display window and not in the language editor. You can view type, name, address and comment of a numerical variable (yellow) in a help bubble (Tool tip) when you position the mouse pointer over the variable.

  NOTE: In an FBD section:
  - the unconnected parameters of elementary functions are not animated,
  - the unconnected parameters of function blocks are animated.

The result of textual expressions is also animated:

- in green and red for Boolean expressions,

- numerical expressions are replaced by their result, the associated help bubble (Tool tip) displays type, name, address and comment of the expression.

Animating the Links

There are 2 types of link:

- Boolean links between function blocks,
- numerical links between function blocks.

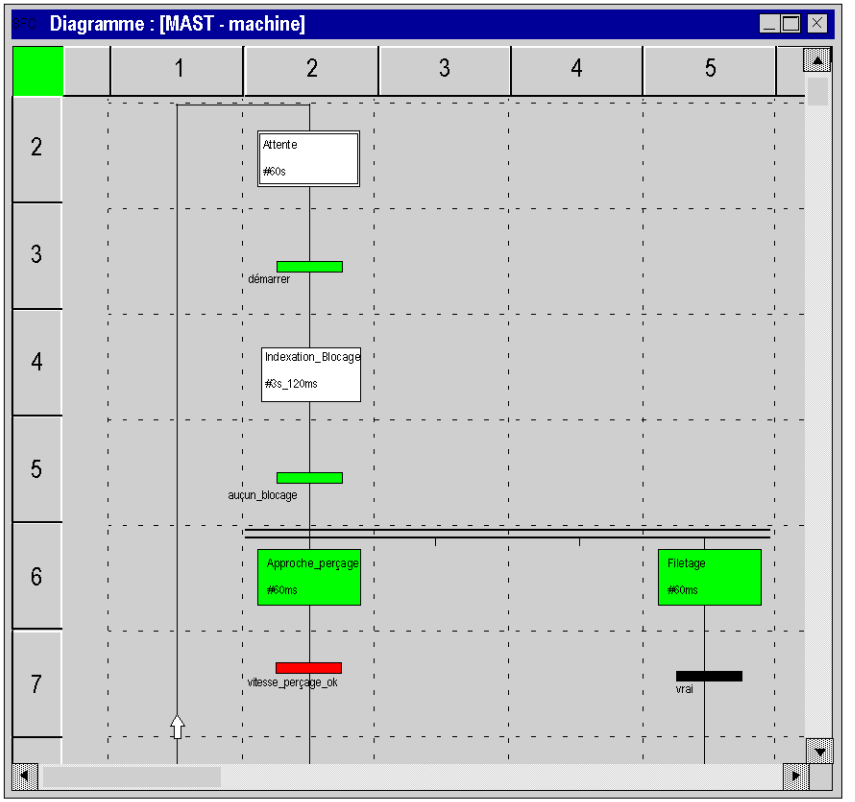
These 2 types of links are animated in the following way:

- Boolean links between function blocks are shown in **green** or **red** depending on whether the output parameters to which they are connected are TRUE (1) or FALSE (0),
- numerical links between the function blocks are shown in **black**. They display the value of the output parameter to which they are connected.

Section in Sequential Function Chart Language (SFC)

At a Glance

Animation of a section in Sequential Function Chart language:



The background color of the editor is **gray**.

The colors used for the different elements are:

- for steps:
  - **green** if the step is active,
  - **white** if the step is inactive,
  - **yellow** if the activity time of the step is less than the minimum programmed time,
  - **pink** if the activity time of the step is greater than the minimum programmed time.

- for macro-steps:
  - when a macro-step becomes active the upper half is shown in **green**,
  - when the OUT step of the macro-step is active the whole of the macro-step is shown in **green**,
  - when the macro-step becomes inactive it is then shown in **white**.
- for transitions associated with a Boolean element or a simple Boolean expression:
  - **green** if the element or the expression is TRUE,
  - **red** if the element or the expression is FALSE.
- for transitions associated with a section:
  - **black** as long as the previous step remains inactive,
  - **green** if the conditions in the section are TRUE,
  - **red** if the conditions in the section are FALSE.

In online mode with the animation activated, if you modify the program and build the project, the animation is not automatically restarted.

NOTE: To return to the animation, click **Services > Animation**.

Display of Prepositioned steps, after an on-line modification

At a Glance

As part of the on-line modifications associated with a prepositioning of the SFC, Control Expert behaves in a specific way when:

- An SFC section uses macro sections
- Sections are animated.
- A prepositioning is performed for this section and for the sections of the macro-steps.

Procedure

Procedure for obtaining the scenario that may cause a problem.

Step	Action
1	Perform a modification in one of the sections. Result: • The animation of the section is stopped. • The prepositionings of the SFC and its macro-sections are reset to zero.
2	Use the command **Build > Build Changes** to generate the modifications.
3	Use the command **PLC > Transfer project to PLC** to load the modifications in the PLC. Result: • The animation of the modified section is relaunched and displays the current state (the prepositioning is reset to zero). • The animation of the sections belonging to the modified section is not updated. In fact, these still display the prepositionings despite the fact they have been deleted.

Solution:

Stop the animation of the sections concerned, then relaunch this animation again.

DFB instances (user function blocks)

At a Glance

A DFB instance may be made up of one section (IEC 1131-3) or several sections.

Animation of a DFB instance consists in **animating a section of the instance** written in one of the programming languages.

The rules governing animation of the DFB sections are the same as for conventional sections according to the language used such as:

- textual languages, page 1401,
- Ladder Diagram language, page 1402,
- Function Block Diagram language, page 1405.

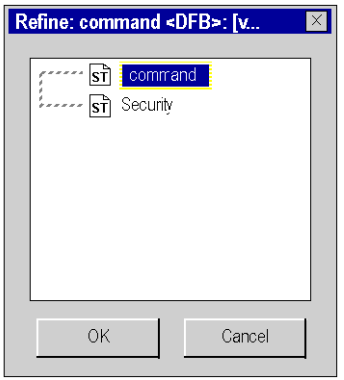
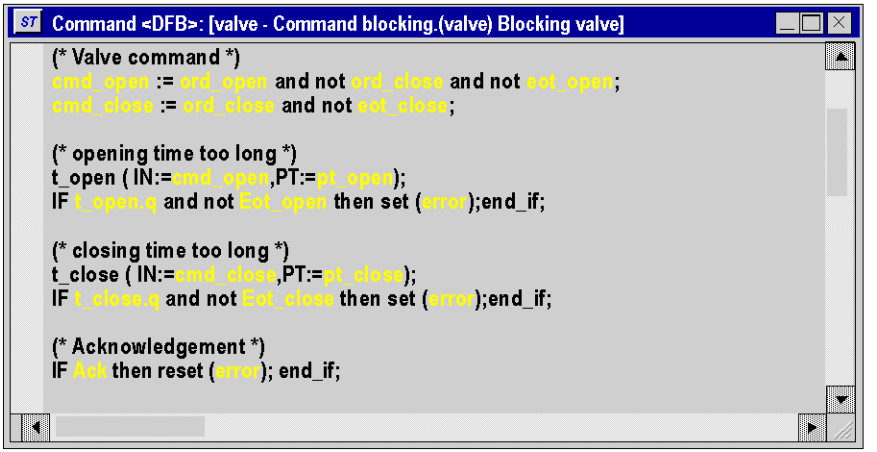
The animated variables are the variables belonging to the DFB instance that has a section which is currently being animated, in other words:

- the input, input\output, output parameters,
- the public variables,
- the private variables.

NOTE: in a DFB section, the unconnected parameters of function blocks are not animated.

How to Animate a Section of a DFB

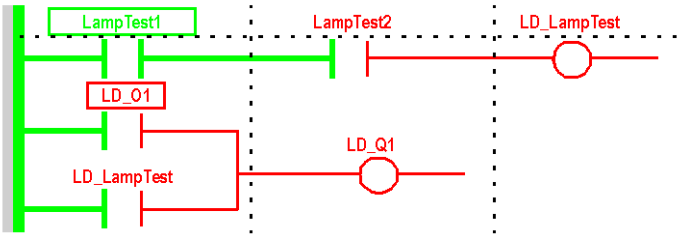
To go to the relevant section of a DFB instance, perform the following operations:

Step	Action
1	In the **animated** section of the program, select the DFB instance and from the contextual menu execute the **Refine** command. A window showing the DFB section(s) opens: 
2	In this box, choose the section of the DFB that you wish to animate and confirm with **OK**. Section in Structured Text: 
3	If the DFB section which is currently animated is a **nested instance** and you wish to animate one of its sections, restart the operation.

Representation of forced variables

At a Glance

Animation of forced variables:



NOTE: Forcing only concerns located Boolean variables of EBOOL type.

NOTE: For Modicon M580 and M340 PLCs analog I/O variables can also be forced.

In **animated mode**, these variables are represented in the following way:

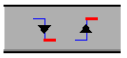
- framed by a **green** rectangle if the value is TRUE (1),
- framed by a **red** rectangle if the value is FALSE (0).

NOTE: Analog I/O variables don't have any special representation when forced. (They are not framed by rectangles)

Implementation

The table below describes the actions to be performed to force variables from a language editor:

Step	Action
1	Open a section.
2	From the **Edit** menu, select all variables using the **Select all** command or Select a variable.
3	Select the **Initialize Animation Table** command from the **Services** menu. **Result**: An animation table opens with the variable(s) selected in the section.
4	Click on **Force** in the animation table.

Step	Action
5	Select the variable to be forced in the animation table.
6	In the animation table click on one of the  buttons alongside the desired value or execute the **Force to 0** or **Force to 1** commands from the contextual menu.

Debugging in Ladder Language

Subject of this Section

This section introduces the different tools available and explains how to implement them to debug the program. These tools are:

- the breakpoint,
- step by step execution,
- the watchpoint.

Inserting a breakpoint in Ladder Diagram language (LD)

At a Glance

A breakpoint can be used to **stop the execution** of the task at the point at which it was set.

During debugging it can be used:

- to examine the behavior of the code,
- to view the value of variables.

There is a single breakpoint at a given point in the project. This is not saved and is lost on disconnection from the PLC.

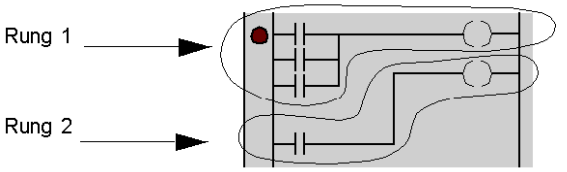
It is implemented in online mode, regardless of whether the PLC is in **Run** or in **Stop**.

NOTE: It is not possible to set a breakpoint in an event task.

Element Used for Setting a Breakpoint

The program element which can be used to set a breakpoint is the **rung** or a Sub-Routine (SR) or user function block (DFB) call within a rung.

In other words, the following program section:

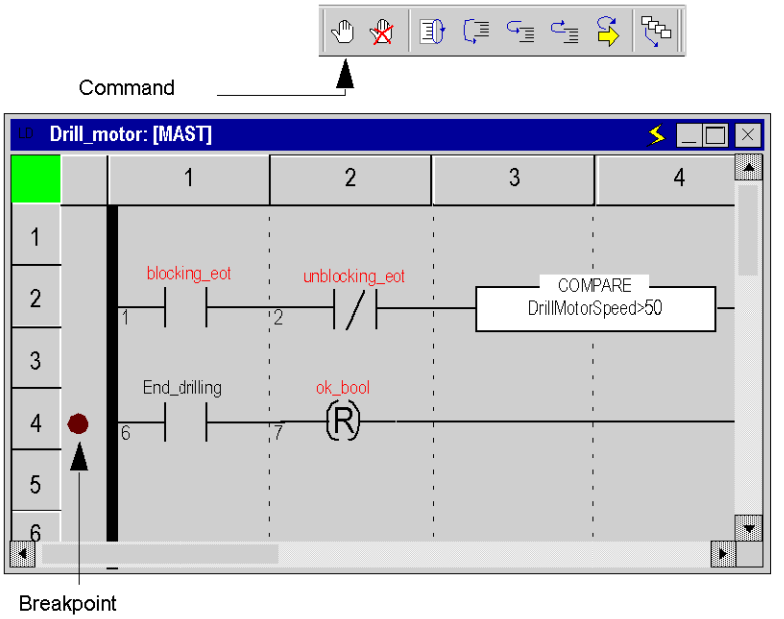


How to Insert a Breakpoint

Carry out the following actions:

Step	Action
1	For example, select a contact in the rung.
2	Set the breakpoint: • by selecting the **Debug->Set Breakpoint** command from the menu, • by selecting the **Set Breakpoint** command from the contextual menu, • or by selecting the  button in the debug toolbar.

Example:



NOTE: Inserting a new breakpoint automatically clears the old one.

From the menu, select the **Debug->Show Breakpoint** command to locate the breakpoint by displaying the section where it has been set in the language editor.

How to Delete a Breakpoint

Either you can:

- select the **Debug->Clear Breakpoint** command from the menu,
- or select the button in the debug toolbar.

Erasing a breakpoint does not restart the task. To do this, you have to press .

Step by step execution mode in Ladder Diagram language (LD)

At a Glance

Step by step mode consists in executing the application program **rung by rung**. For rungs containing sub-routine (SR) or user function block (DFB) calls, step by step can be written into the code used to implement them.

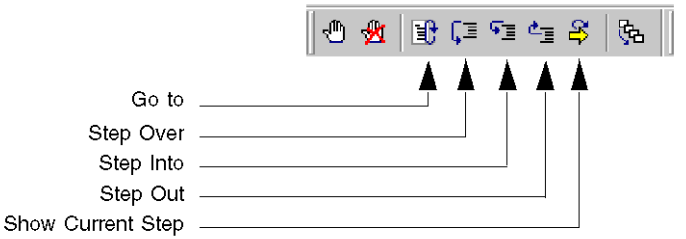
This mode is launched by a **breakpoint** which will have been set in advance. It is used to examine the behavior of the code, and the value of the variables.

It is implemented in online mode. The section executed in step by step mode stops the corresponding task, and the links are no longer animated.

Three commands can be used for step by step mode. These are:

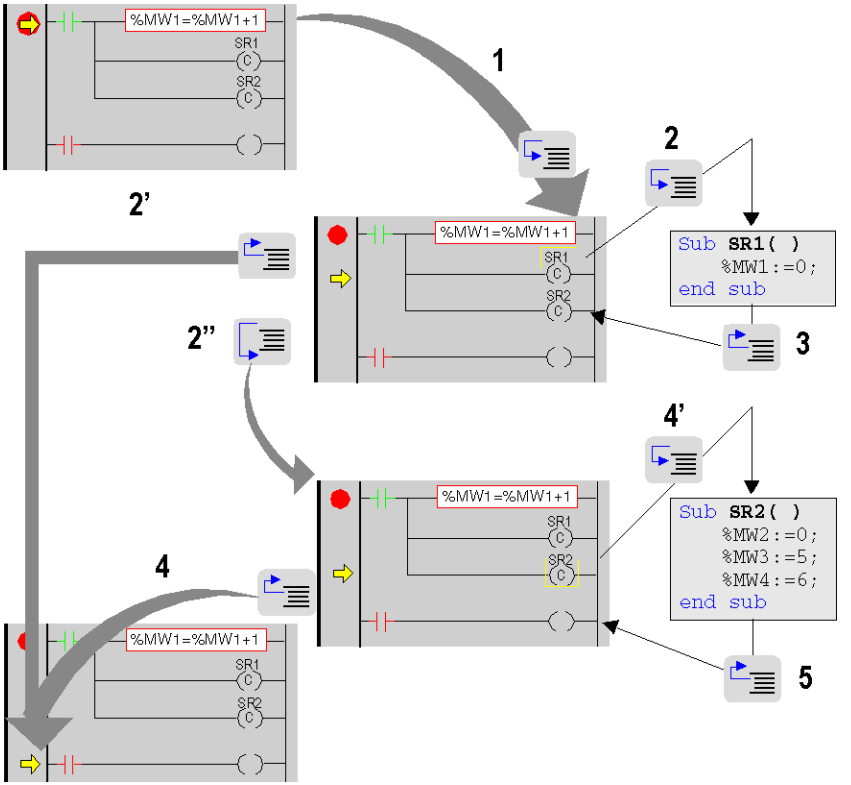
- the **Step Into** command,
  - if the current element is a rung which does not contain an SR call or a DFB instance call, this command executes the rung and moves onto the next rung,
  - if the current element is a rung which does contain an SR call or a DFB instance call, this command executes the start of the rung and moves to the first SR or DFB call.
  - if the current element is an SR or DFB instance call, this command can be used to step into the code and go to the first element.
- the **Step Over** command,
  - if the current element is a rung, this command executes it in its entirety and moves onto the next rung,
  - if the current element is an SR or DFB instance call, this command executes it in its entirety as if it were a straightforward element and moves onto the next element.
- the **Step Out** command.
  - if the current element is part of the code of an SR or DFB instance, this command can be used to execute all the elements of the SR or the DFB and to move onto the next element after the SR or DFB call,
  - if the current element is an SR call or a DFB instance call, this command executes the remainder of the current rung in its entirety and moves onto the next rung,
  - if the current element is a rung, this command executes the current section in its entirety and moves onto the start of the next section.

From step by step mode, if you want to relaunch the execution of the task in order to return to the **breakpoint** you set earlier, select the **Debug->Go** command from the menu, or click on **Go** in the toolbar:



Step Into, Step Out and Step Over

Example:



Description of step by step from the breakpoint reached  :

Step	Action
1	Clicking on the **Step Into** button moves execution of the program to the SR1 call.
2 or 2' or 2''	If you click on: • **Step Into** (2) execution of the program moves onto `%MW1:=0` • **Step Out** (2') the rung is executed and execution of the program moves onto the next rung. • **Step Over** (2'') SR1 is executed and execution of the program moves on to call SR2.
3	Clicking on **Step Out** while in SR1 executes SR1 in its entirety and moves execution of the program on to call SR2.
4 or 4' or	If you click on: • **Step Into** (4) execution of the program moves onto `%MW2:=0` Clicking on **Step Into** moves execution of the program onto `%MW3:=5`, and so on if you want to use step by step in SR2. • **Step Out** (4) the rung is executed and execution of the program moves onto the next rung,
5	Clicking on **Step Out** while in SR2 executes SR2 in its entirety and moves execution of the program onto the next rung.

NOTE: When program execution is over an instruction, it will not yet have been executed. It will be executed after a command is pressed.

Specific Case of Step Out

If you want to perform step by step **section by section**, execute the **Step Out** command from the first element of each section.

NOTE: If the next section is an **SFC section or a section implementing an action or a transition** (mandatory in the Mast task), this is skipped unless a language element has a breakpoint.

Rule

In step by step mode the task manager will not detect a **watchdog overflow** for the task currently being debugged. It will, however, detect infinite loops.

Inserting a Watchpoint in Ladder Diagram Language

At a Glance

When there is no watchpoint, the values of animated variables are displayed at the end of MAST task processing.

The limitation of this mode of operation is that it does not permit the value of a variable at a given point in the program to be known if this value is used in different sections.

The watchpoint is used to **synchronize display of animated variables** with **execution of a program element** (rung) in order to know their exact value at this specific point in the program.

Variables which are synchronous with the watchpoint **must belong to the section** in which the watchpoint is set; display of other variables is synchronous with the end of the MAST task.

These variables are displayed:

- in the section ladder editor,
- in the display window, page 1468,
- in a single animation table at a given moment when requested by the user.

NOTE: It is not possible to set a watchpoint in an event task.

Watchpoint Properties

These properties are:

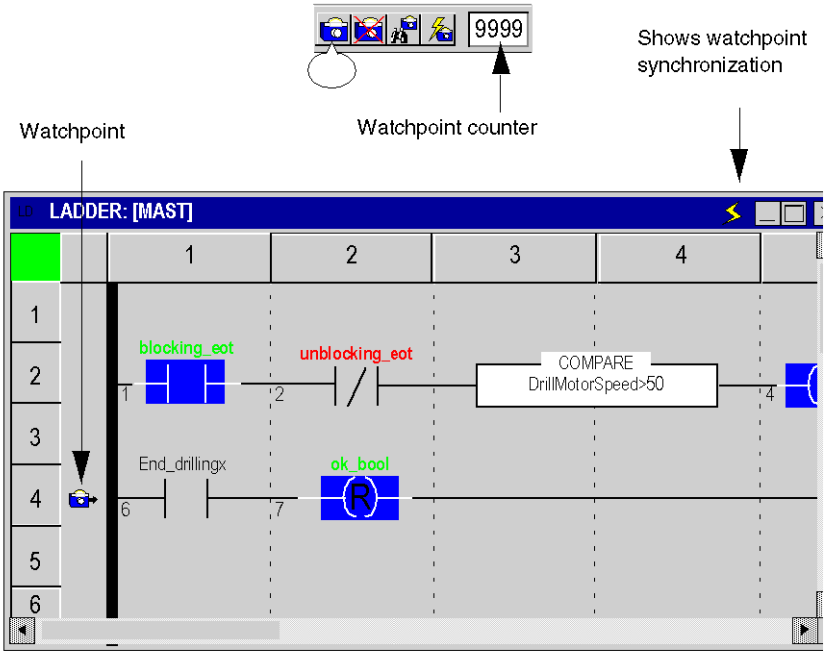
- the watchpoint can be set only in online mode; if the connection is broken, the watchpoint is lost as well,
- a single watchpoint is allowed at a given moment; the watchpoint is exclusive with the breakpoint,
- display of animated variables is effective before execution of the rung on which the watchpoint has been set,
- a counter is incremented each time the rung with the watchpoint is executed, up to a maximum value of 9999, then reset to 0,
- modification of a section is not authorized if a watchpoint is set.

How to Insert a Watchpoint

Carry out the following actions:

Step	Action
1	For example, select a contact in the rung.
2	Set the watchpoint in one of the following ways: • by selecting the following command from the menu: **Debug->Set Watchpoint**, • by selecting the following command from the contextual menu: **Set Watchpoint**, • or select the  button in the toolbar.

Example:



NOTE: Inserting a new watchpoint automatically clears the old one.

How to Locate an Existing Watchpoint

Carry out the following actions:

Step	Action
1	Either: • by selecting the following command from the menu: **Debug->Show Watchpoint,** • or select the  button in the toolbar.
2	The part of the language editor containing the watchpoint is displayed.

How to Synchronize with an Animation Table

Carry out the following actions:

Step	Action
1	Select the animation table desired.
2	either you can: • select the following command from the menu: **Debug->Synchronize Animation Table**. • or select the  button in the toolbar.

How to Delete a Watchpoint

Either you can:

- select the **Debug->Clear Watchpoint** command from the menu,
- or select the  button in the toolbar.

Debugging in Textual Languages (Structured Text, Instruction List)

Subject of this Section

This section introduces the different tools available and explains how to implement them to debug the program. These tools are:

- the breakpoint,
- step by step execution,
- the watchpoint.

Insertion of a breakpoint in textual languages (Structured Text or Instruction List)

At a Glance

A breakpoint can be used to **stop the execution** of the task at the point at which it was set.

During debugging it can be used:

- to examine the behavior of the code,
- to view the value of variables.

There is a single breakpoint at a given point in the project. This is not saved and is lost on disconnection from the PLC.

It is implemented in online mode regardless of whether the PLC is in **Run** or in **Stop**.

NOTE: It is not possible to set a breakpoint in an event task.

Element Used for Setting a Breakpoint

The program element used to set a breakpoint is an **instruction**.

In other words, the following program section:

```
Line 1 (*Example*)
Line 2 IF(%MW10=14) THEN  (*Test condition*)
Line 3     %MW45:=68;     (*Execution if condition true*)
Line 4 ELSE
Line 5     %MW45:=24;     (*Execution if condition false*)
Line 6 END_IF;
Line 7 %MW10:=12; DFB1_3(prms); %MW0:=%MW0+1;
Line 8
```

Lines 2, 3, 5 and 7 can hold a breakpoint.

Lines 1, 4, 6 and 8 cannot hold a breakpoint.

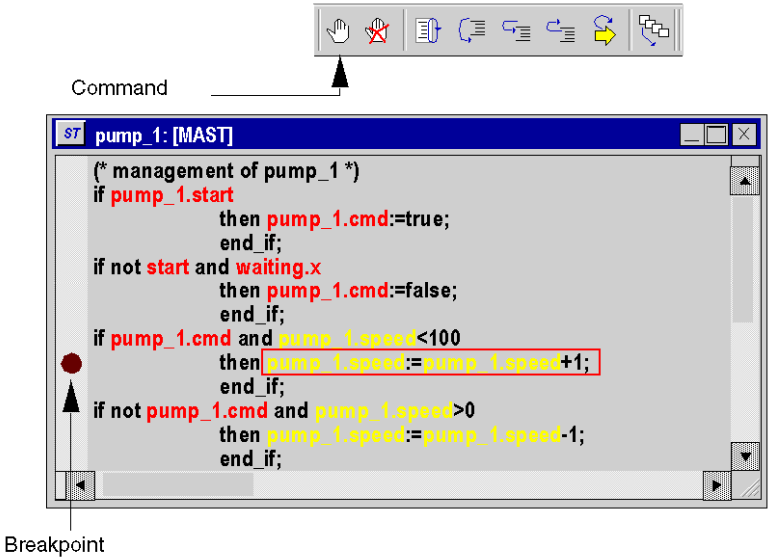
NOTE: On line 7 (several instructions), the breakpoint can be set on the first, second or third instruction. **The instruction with the breakpoint is framed.**

How to Insert a Breakpoint

Carry out the following actions:

Step	Action
1	Select the desired program element.
2	Set the breakpoint: • by selecting the following command from the menu: **Debug->Set Breakpoint**, • or by selecting the button in the debug toolbar.

Example:



NOTE: Inserting a new breakpoint automatically clears the old one.

From the menu, select the **Debug->Show Breakpoint** command to locate the breakpoint by displaying the section where it has been set in the language editor.

How to Delete a Breakpoint

Either you can:

- select the **Debug->Clear Breakpoint** command from the menu,

- or select the button in the debug toolbar.

Erasing a breakpoint does not restart the task. To do this, you have to press .

Step by step execution mode in textual languages (Structured Text or Instruction List)

At a Glance

Step by step mode consists in executing the application program **instruction by instruction**. A line may contain several instructions.

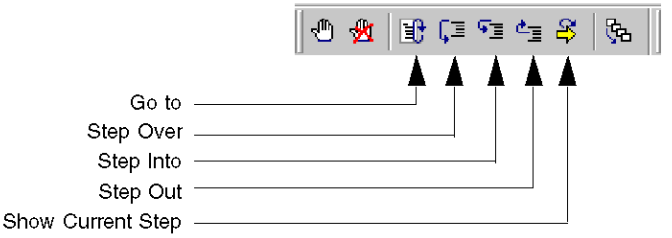
This mode is launched by a **breakpoint** which will have been set in advance. It is used to examine the behavior of the code, and the value of the variables.

It is implemented in online mode. The section executed in step by step mode stops the corresponding task.

Three commands can be used for step by step mode. These are:

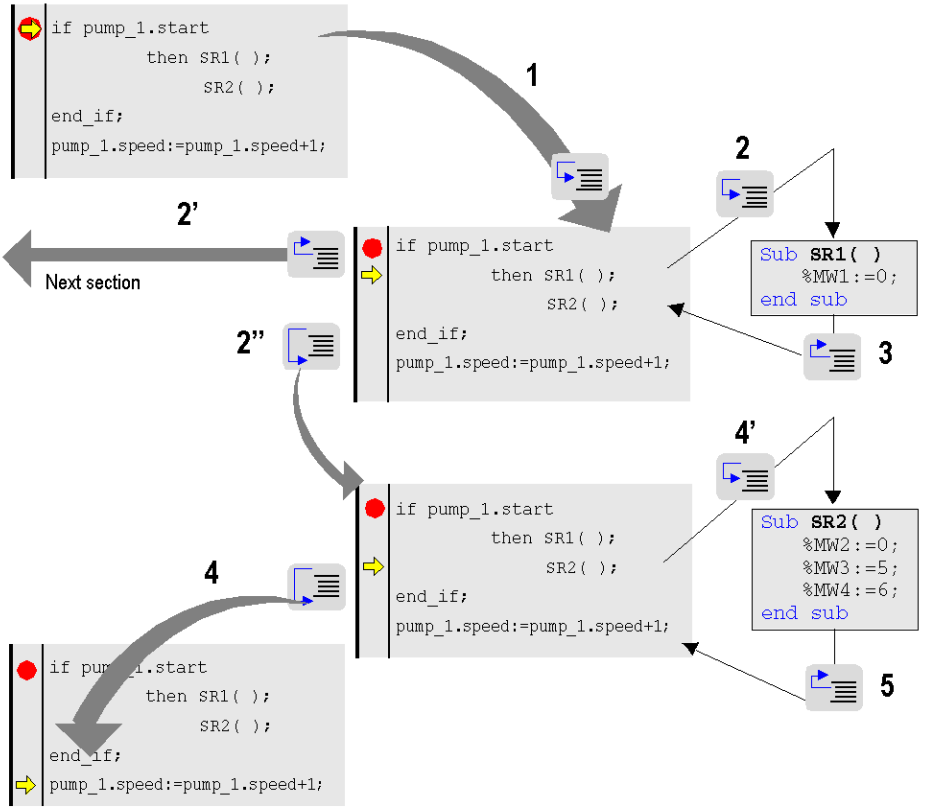
- the **Step Into** command,
  - if the current element is an SR (sub-routine) or DFB (user function block) instance call, this command can be used to step into the code and go to the first element of the SR or DFB,
  - if the current element is an instruction, this command executes it and moves onto the next instruction.
- the **Step Over** command,
  - if the current element is an SR or DFB instance call, this command executes it in its entirety as if it were a straightforward element and moves onto the next instruction,
  - if the current element is an instruction, this command executes it and moves onto the next instruction.
- the **Step Out** command.
  - if the current element is part of the code of an SR or DFB, this command can be used to execute all the elements of the SR or the DFB and to move onto the next element of the SR or DFB.
  - if the current element is an instruction, this command executes the current section in its entirety and moves onto the start of the next section.

From step by step mode, if you want to relaunch the execution of the task in order to return to the **breakpoint** you set earlier, select the **Debug->Go** command from the menu, or click on **Go** in the toolbar:



Step Into, Step Out and Step Over

Example:



Description of step by step from the breakpoint reached  :

Step	Action
1	Clicking on the **Step Into** button moves execution of the program to the SR1 call.
2 or 2' or 2''	If you click on: • **Step Into** (2) execution of the program moves onto `%MW1:=0` • **Step Out** (2') the section is executed and execution of the program moves onto the first element of the next section. • **Step Over** (2'') SR1 is executed and execution of the program moves on to call SR2.
3	Clicking on **Step Out** while in SR1 executes SR1 in its entirety and moves execution of the program on to call SR2.

Step	Action
4 or 4’	If you click on: • **Step Into** (4) execution of the program moves onto `%MW2:=0` Clicking on **Step Into** moves execution of the program onto `%MW3:=5`, and so on if you want to use step by step in SR2. • **Step Over** (4) SR2 is executed and execution of the program moves onto the next instruction.
5	Clicking on **Step Out** while in SR2 executes SR2 in its entirety and moves execution of the program onto the next instruction.

NOTE: When program execution is over an instruction, it will not yet have been executed. It will be executed after a command is pressed.

Specific Case of Step Out

If you want to perform step by step **section by section**, execute the **Step Out** command from the first element of each section.

NOTE: If the next section is an **SFC section or a section implementing an action or a transition** (mandatory in the Mast task), this is skipped unless a language element has a breakpoint.

Rule

In step by step mode the task manager will not detect a **watchdog overflow** for the task currently being debugged. It will, however, detect infinite loops.

Insertion of a Watchpoint in Textual Languages (Structured Text or Instruction List)

At a Glance

When there is no watchpoint, the values of animated variables are displayed at the end of MAST task processing.

The limitation of this mode of operation is that it does not permit the value of a variable at a given point in the program to be known if this value is used in different sections.

The watchpoint is used to **synchronize display of animated variables** with **execution of a program element** (instruction) in order to know their exact value at this specific point in the program.

Variables which are synchronous with the watchpoint must belong to the section in which the watchpoint is set; display of other variables is synchronous with the end of the MAST task.

These variables are displayed:

- in the section language editor,
- in the display window,
- in a single animation table at a given moment when requested by the user.

NOTE: It is not possible to set a watchpoint in an event task.

NOTE: Using a watchpoint degrades real-time performances due to increased application overhead. It is not recommended to use a watchpoint in a code loop, because the watchpoint would be called n times in a cycle and cause major overhead, possibly triggering the task watchdog.

Watchpoint Properties

These properties are:

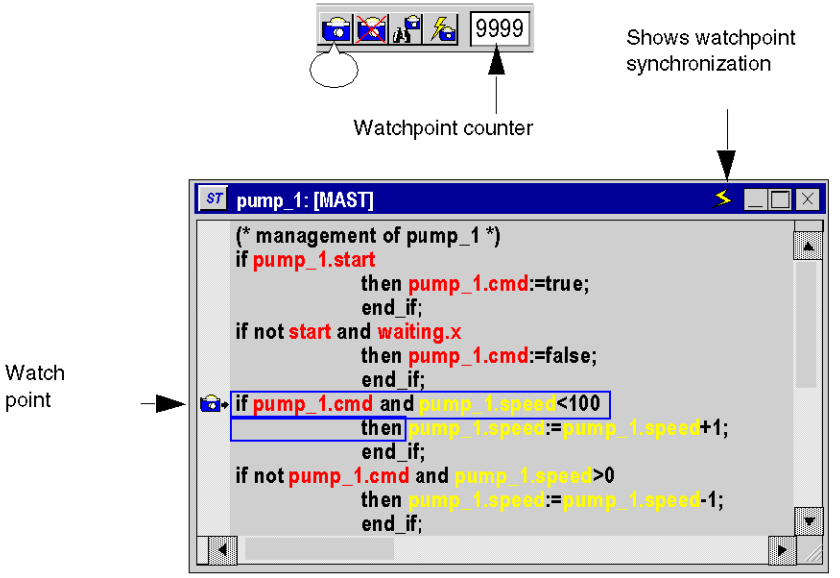
- the watchpoint can be set only in online mode; if the connection is broken, the watchpoint is lost as well,
- a single watchpoint is allowed at a given moment; the watchpoint is exclusive with the breakpoint,
- display of animated variables is effective before execution of the instruction on which the watchpoint has been set,
- a counter is incremented each time the instruction with the watchpoint is executed, up to a maximum value of 9999, then reset to 0,
- modification of a section is not authorized if a watchpoint is set.

How to Insert a Watchpoint

Carry out the following actions:

Step	Action
1	In the section, select the desired instruction.
2	Set the watchpoint in one of the following ways: • by selecting the following command from the menu: **Debug->Set Watchpoint**, • by selecting the **Set Watchpoint** command from the contextual menu, • or select the  button in the toolbar.

Example:



NOTE: Inserting a new watchpoint automatically clears the old one.

How to Locate an Existing Watchpoint

Carry out the following actions:

Step	Action
1	Either: • by selecting the following command from the menu: **Debug->Show Watchpoint**. • or select the  button in the toolbar.
2	The part of the language editor containing the watchpoint is displayed.

How to Synchronize with an Animation Table

Carry out the following actions:

Step	Action
1	Select the animation table desired.
2	either you can: • select the following command from the menu: **Debug->Synchronize Animation Table**. • or select the  button in the toolbar.

How to Delete a Watchpoint

Either you can:

- select the following command from the menu:
  Debug->Clear Watchpoint,
- or select the  button in the toolbar.

Debugging in Function Block Diagram (FBD) Language

Subject of this Section

This section introduces the different tools available and explains how to implement them to debug the program. These tools are:

- the breakpoint,
- step by step execution,
- the watchpoint.

Inserting a breakpoint in Function Block Diagram language

At a Glance

A breakpoint can be used to **stop the execution** of the task at the point at which it was set.

During debugging it can be used:

- to examine the behavior of the code,
- to view the value of variables.

There is a single breakpoint at a given point in the project. This is not saved and is lost on disconnection from the PLC.

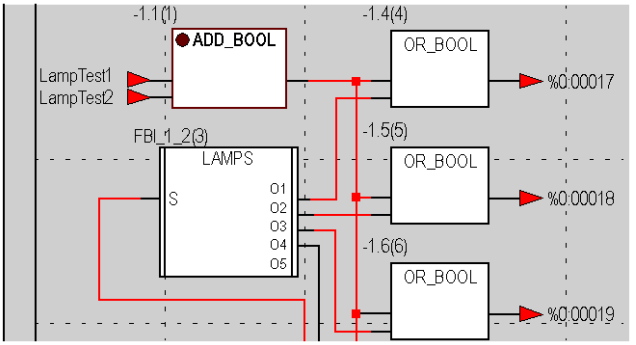
It is implemented in online mode regardless of whether the PLC is in **Run** or in **Stop**.

NOTE: It is not possible to set a breakpoint in an event task.

Element Used for Setting a Breakpoint

The program element used to set a breakpoint is the **function block**.

In other words, the following program section:



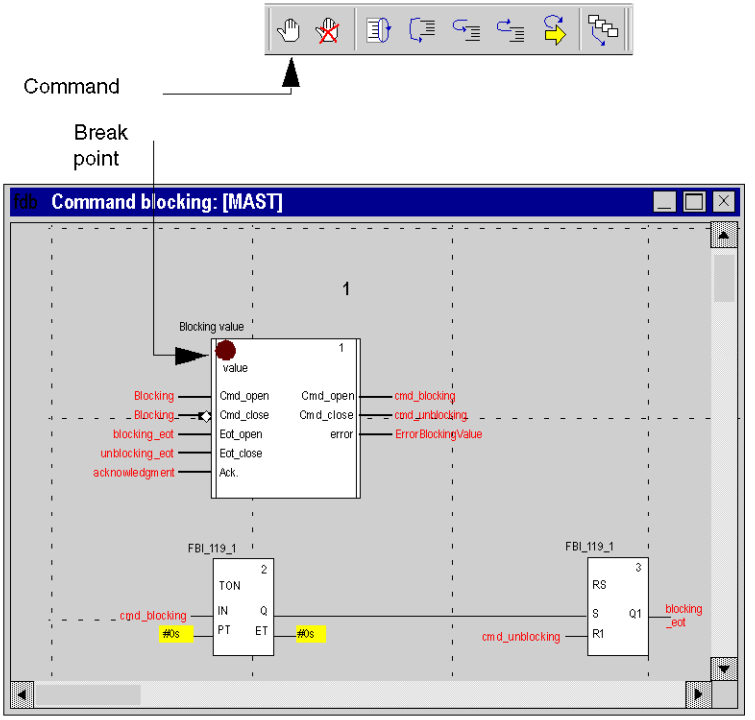
The breakpoint can only be set on 1 block at a time. The number in brackets over the block corresponds to the order of execution.

How to Insert a Breakpoint

Carry out the following actions:

Step	Action
1	Select the desired program element.
2	Set the breakpoint: • by selecting the following command from the menu: **Debug->Set Breakpoint**, • by selecting the following command from the contextual menu: **Set Breakpoint**, • or by selecting the button in the debug toolbar.

Example:



NOTE: Inserting a new breakpoint automatically clears the old one.

From the menu, select the **Debug->Show Breakpoint** command to locate the breakpoint by displaying the section where it has been set in the language editor.

How to Delete a Breakpoint

Either you can:

- select the following command from the menu:

  Debug->Clear Breakpoint,

- or select the  button in the debug toolbar.

Erasing a breakpoint does not restart the task. To do this, you have to press  .

Step by step execution mode in Function Block Diagram language (FBD)

At a Glance

Step by step mode consists in executing the application program **function block by function block**.

This mode is launched by a **breakpoint** which will have been set in advance. It is used to examine the behavior of the code, and the value of the variables.

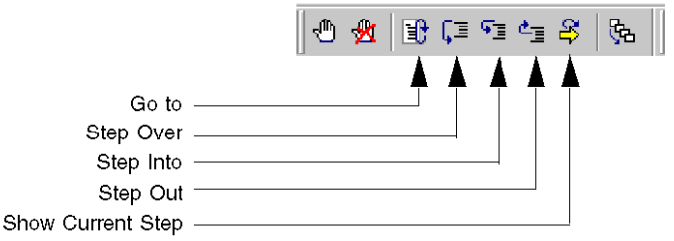
It is implemented in online mode. The section executed in step by step mode stops the corresponding task.

Three commands can be used for step by step mode. These are: :

- the **Step Into** command,
  - if the current element is an SR (sub-routine) or DFB (user function block) instance call, this command can be used to step into the code and go to the first element of the SR or DFB,
  - if the current element is a function block, this command executes it and moves onto the next function block.
- the **Step Over** command,
  - if the current element is an SR or DFB instance call, this command executes it in its entirety as if it were a straightforward element and moves onto the next function block,
  - if the current element is a function block, this command executes it and moves onto the next function block.

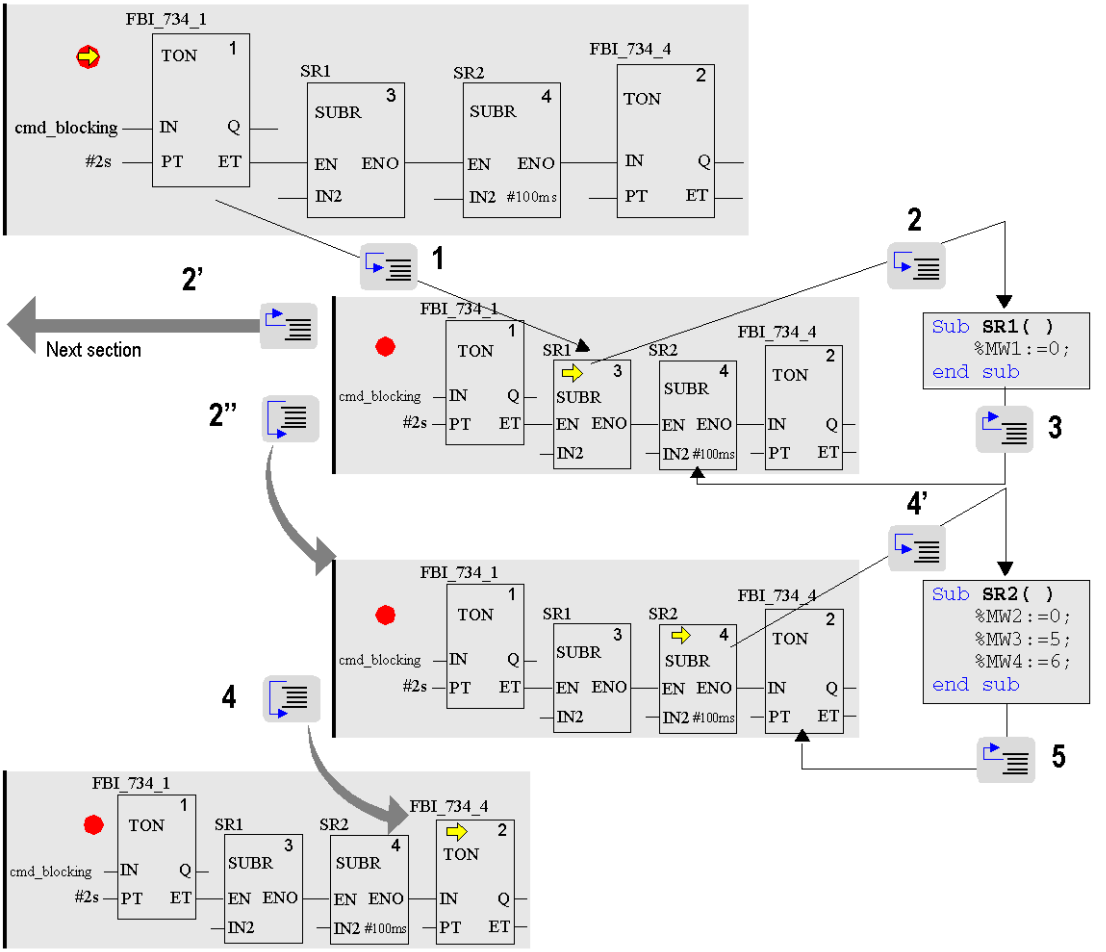
- the **Step Out** command.
  - if the current element is part of the code of an SR or DFB, this command can be used to execute all the elements of the SR or the DFB and to move onto the next element of the SR or DFB.
  - if the current element is a function block, this command executes the current section in its entirety and moves onto the start of the next section.

From step by step mode, if you want to relaunch the execution of the task in order to return to the **breakpoint** you set earlier, select the **Debug->Go** command from the menu, or click on **Go** in the toolbar:



Step Into, Step Out and Step Over

Example:



Description of step by step from the breakpoint reached  :

Step	Action
1	Clicking on the **Step Into** button moves execution of the program to the SR1 call.
2 or 2' or 2''	If you click on: • **Step Into** (2) execution of the program moves onto `%MW1:=0` • **Step Out** (2') the section is executed and execution of the program moves onto the first element of the next section.

Step	Action
	• **Step Over** (2'') SR1 is executed and execution of the program moves on to call SR2.
3	Clicking on **Step Out** while in SR1 executes SR1 in its entirety and moves execution of the program on to call SR2.
4 or 4'	If you click on: • **Step Into** (4) execution of the program moves onto `%MW2:=0` Clicking on **Step Into** moves execution of the program onto `%MW3:=5`, and so on if you want to use step by step in SR2. • **Step Over** (4) SR2 is executed and execution of the program moves onto the next function block.
5	Clicking on **Step Out** while in SR2 executes SR2 in its entirety and moves execution of the program onto the next function block.

NOTE: When program execution is over a function block, it will not yet have been executed. It will be executed after a command is pressed.

Specific Case of Step Out

If you want to perform step by step **section by section**, execute the **Step Out** command from the first element of each section.

NOTE: If the next section is an **SFC section or a section implementing an action or a transition** (mandatory in the Mast task), this is skipped unless a language element has a breakpoint.

Rule

In step by step mode the task manager will not detect a **watchdog overflow** for the task currently being debugged. It will, however, detect infinite loops.

Inserting a watchpoint in Function Block Diagram language (FBD)

At a Glance

When there is no watchpoint, the values of animated variables are displayed at the end of MAST task processing.

The limitation of this mode of operation is that it does not permit the value of a variable at a given point in the program to be known if this value is used in different sections.

The watchpoint is used to **synchronize display of animated variables** with **execution of a program element** (function block) in order to know their exact value at this specific point in the program.

Variables which are synchronous with the watchpoint **must belong to the section** in which the watchpoint is set; display of other variables is synchronous with the end of the MAST task.

These variables are displayed:

- in the section language editor,
- in the display window,
- in a single animation table at a given moment when requested by the user.

NOTE: It is not possible to set a watchpoint in an event task.

Watchpoint Properties

These properties are:

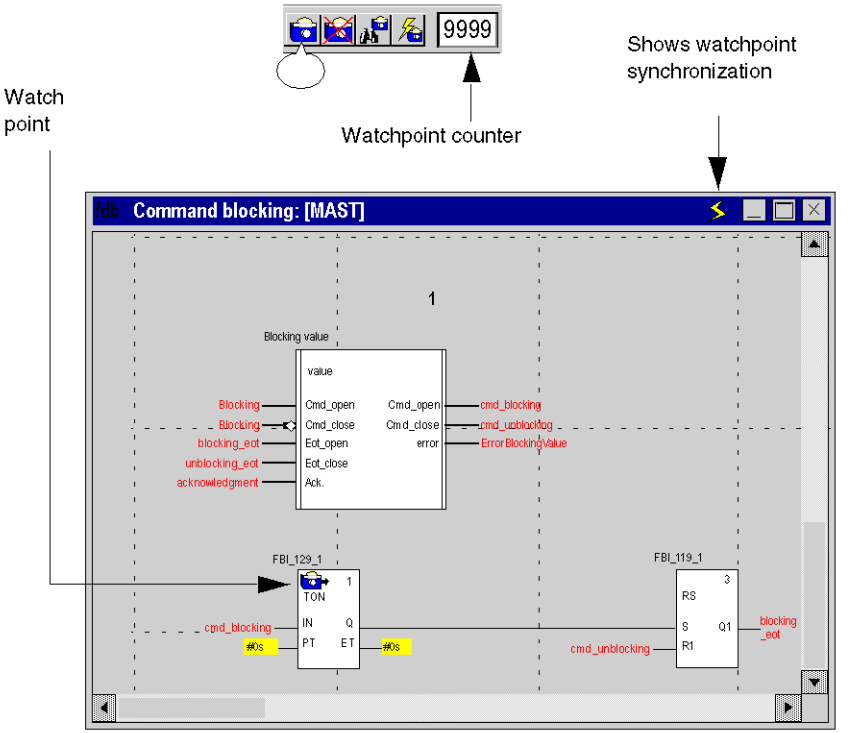
- the watchpoint can be set only in online mode; if the connection is broken, the watchpoint is lost as well,
- a single watchpoint is allowed at a given moment; the watchpoint is exclusive with the breakpoint,
- display of animated variables is effective before execution of the block on which the watchpoint has been set,
- a counter is incremented each time the function block with the watchpoint is executed, up to a maximum value of 9999, then reset to 0,
- modification of a section is not authorized if a watchpoint is set.

How to Insert a Watchpoint

Carry out the following actions:

Step	Action
1	In the section, select the block desired.
2	Set the watchpoint in one of the following ways: • by selecting the following command from the menu: **Debug->Set Watchpoint**, • by selecting the following command from the contextual menu: **Set Watchpoint**, • or select the  button in the toolbar.

Example:



NOTE: Inserting a new watchpoint automatically clears the old one.

How to Locate an Existing Watchpoint

Carry out the following actions:

Step	Action
1	Either: • by selecting the following command from the menu: **Debug->Show Watchpoint**. • or select the  button in the toolbar.
2	The part of the language editor containing the watchpoint is displayed.

How to Synchronize with an Animation Table

Carry out the following actions:

Step	Action
1	Select the animation table desired.
2	either you can: • select the following command from the menu: **Debug->Synchronize Animation Table**. • or select the  button in the toolbar.

How to Delete a Watchpoint

Either you can:

- select the **Debug->Clear Watchpoint** command from the menu,
- or select the  button in the toolbar.

Debugging User Function Block (DFB) Instances

Subject of this Section

This section introduces the different tools available and explains how to implement them to debug the DFB instances of the program. These tools are:

- the breakpoint,

- step by step execution,
- the watchpoint.

Inserting a breakpoint in a DFB instance (user function block)

At a Glance

A breakpoint can be used to **stop the execution** of the task which contains the DFB instance in which it has been set.

During debugging it can be used:

- to examine the behavior of the code,
- to view the value of the variables and the input/output parameters of the instance.

There is a single breakpoint at a given point in the project. This is not saved and is lost on disconnection from the PLC.

It is implemented in online mode regardless of whether the PLC is in **Run** or in **Stop**.

Elements Used for Setting a Breakpoint

The program element used to set a breakpoint in a DFB section depends on the language in which the section has been written:

- the **rung and the nested DFB call** for Ladder Diagram language, page 1413,
- **the instruction** for textual languages, page 1422,
- the **Function Block** for Function Block Diagram, page 1431 language.

How to Insert a Breakpoint

Access the DFB instance section, page 1410 from which you wish to insert a breakpoint.

Depending on the language used in the DFB section, proceed in the following manner:

- for Ladder Diagram language, page 1414,
- for textual languages, page 1423,
- for Function Block Diagram language, page 1423.

If **several instances of a DFB** are used in the application, the breakpoint is valid for all calls as the DFB code is shared by all the instances.

The task corresponding to the first call (order of execution) is stopped when the breakpoint in reached.

By relaunching execution using the **Go** button  the task corresponding to the next call is stopped and so on.

NOTE: Inserting a new breakpoint automatically clears the old one.

How to Locate an Existing Breakpoint

Carry out the following actions:

Step	Action
1	Select the following command from the menu: **Debug->Show Breakpoint**.
2	The part of the language editor containing the breakpoint is displayed.

How to Delete a Breakpoint

Either you can:

- select the following command from the menu:
  Debug->Clear Breakpoint,
- or select the  button in the debug toolbar.

Erasing a breakpoint does not restart the task. To do this, you have to press  .

Step by step execution mode in a DFB instance (user function block)

At a Glance

For a DFB section programmed in:

- Ladder Diagram language, page 1416,
- textual languages, page 1424,
- Function Block Diagram language, page 1433.

Step Into, Step Out and Step Over

Step by step functions are used in the same way for a section belonging to a DFB as for a conventional section.

NOTE: a DFB section may contain a sub-routine (SR).

For a DFB section programmed in:

- Ladder Diagram language, page 1417,
- textual languages, page 1426,
- Function Block Diagram language, page 1435.

DFB with Several Sections

A DFB may be made up of several sections.

The **Step Out** command can be used to move onto the instance following the DFB, and when applied to the last section allows you to exit the instance.

When you want to quit a DFB instance perform as many **Step Out** commands as there are sections. Program execution will go to the first element after the DFB instance call.

Rule

In step by step mode the task manager will not detect a **watchdog overflow** for the task containing the DFB instance. It will, however, detect infinite loops.

Inserting a watchpoint in a user function block (DFB)

At a Glance

Access the DFB instance section , page 1410from which you wish to insert a watchpoint.

For a DFB section programmed in:

- ladder language, page 1419,
- textual languages, page 1427,
- Function Block Diagram language, page 1436.

Debugging in Sequential Function Chart Language (SFC)

Subject of this Section

This section introduces the different tools available and explains how to implement them to debug the program. These tools are:

- the breakpoint,
- step by step execution,
- the watchpoint.

Insert a Breakpoint in Sequential Function Chart Language (SFC)

At a Glance

In contrast to other languages, a breakpoint in SFC does not stop execution of the task corresponding to the point at which it has been set, but freezes the chart and automatically activates the mode "disable transitions".

For this SFC section (section of the breakpoint), the controller remains in "disable transitions" mode until you quit this mode using the **Animation panel** tool.

In "disable transitions" mode, the background color of the editor is blue.

During debugging it can be used:

- to examine the behavior of the code,
- to view the value of variables.

In contrast to other languages, several breakpoints can be set at a given point in an SFC section. They are not saved and are discarded on disconnection from the controller.

They are implemented in online mode, regardless of whether the controller is in **Run** or in **Stop**.

NOTE: It is not possible to set a breakpoint in an event task.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

When using breakpoint in SFC, power off the operating part of the system to ensure that the task and its associated inputs/outputs do not present hazards to people and the equipment.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Element Used for Setting a Breakpoint

From an **SFC section** the program element used to set a breakpoint is the **step**.

From a **section which contains the processing of a step or a transition** the program element used to set a breakpoint corresponds to the language used, i.e.:

- Ladder Diagram language, page 1413,
- Structured Text or Instruction List language, page 1422,
- Function Block Diagram language, page 1431.

Insert a Breakpoint

From an **SFC section**:

Step	Action
1	Select the desired step(s).
2	Set the breakpoint(s) by right-clicking the desired element and selecting the following command: **Animation > Insert/Remove Breakpoint**.

From a section corresponding to an action or a section corresponding to a transition the procedures correspond to the language used, i.e.:

- Ladder Diagram language, page 1414,
- Structured Text or Instruction List language, page 1423,
- Function Block Diagram language, page 1431.

Delete a Breakpoint from an SFC Section

Select the breakpoint to be deleted, then select the **Debug ->Clear Breakpoint** command from the menu.

NOTE: When you initiate a **StepOver** or **StepOut** in a chart, invisible breakpoints are created by SFC editor in the same time than the breakpoint created. In order to remove all these breakpoints use the function **ClearAllBreakpoints**, otherwise the chart execution will stop via **DisableTransition** check button.

Step by Step Execution Mode in Sequential Function Chart Language (SFC)

At a Glance

In SFC language you have two debugging levels:

- from an **SFC section (chart)**,
- from **a section implementing an action or a transition**, where this section is written in one of the following languages (Ladder, Structured Text, Function Block Diagram, Instruction List).

Sections Implementing Actions or Transitions

In this type of section, debugging behaves in practically the same way as debugging for a conventional section using one of the following languages:

- Ladder Diagram language, page 1416,
- textual languages, page 1424,
- Function Block Diagram language, page 1433.

The specific characteristics of this type of debug are:

- it is impossible to enter a section of an action or a transition in step by step mode when coming from another section, unless this section contains a breakpoint,
- when you are at the end of a section of an action or a transition it is impossible to exit it in step by step mode by using the commands **Step Into, Step Over, Step Out**, **Go** is the only command available.

SFC Sections (Chart)

Step by step mode consists in executing the application program **step by step**.

NOTE: When program execution is over a step, it will not yet have been executed. It will be executed after a command is pressed.

This mode is launched by a **breakpoint** which will have been set in advance (on a step). It is used to examine the behavior of the code, and the value of the variables.

It is implemented in online mode. The section executed in step by step mode in contrast to sections in languages other than SFC does not stop the task, but **freezes the chart**.

The debugging of an SFC section (chart) is checked using the Animation Panel (**Services->Animation Panel**). Each SFC section (chart) has its own tool.

Animation Panel Tool

To launch the tool, carry out the following actions:

Step	Action
1	Open the editor for the SFC section (chart) that you wish to debug.
2	Select the following command from the menu: **Services->Animation Panel->Show Animation Panel** **Result**: The tool is displayed.

Animation Panel tool:



Description of the different commands:

SFC section name	
Parameters zone	
Initialize Chart	For an SFC section which is currently being executed, do the following in order: 1) Select this check box to deactivate the active step(s), 2) Clear this check box to activate the initial steps to start the section.
Disable Time Check	Select this check box to disable checking of step execution times.
Disable Transitions	Select this check box to make it possible to freeze the chart in any transition state. This check box must be selected to use the following buttons: **Step Trans Dependent**, **Step Over** and **Step Out**
Disable Actions	Select this check box to stop processing of steps.
Operation zone	
Clear Chart	Deactivates the active step(s).
Reset Time Errors	Resets information on step activity time overruns.
Step Unconditional	Move on to the next step without acknowledgment of the transition condition. This cannot be done if you have a Delay Time (SFCSTEP_TIME.delay) set in the properties of the STEP.
Step Trans Dependent	Moves to the successor step if the transition condition and the step delay time (SFCSTEP_TIME.delay) are fulfilled. If the transition condition is not right, the chart is unfreezed and waits for the right transition's condition. With the activation of the successor step the chart returns to the freeze state.
Step Over	This command moves to the successor step if the transition condition and the step delay time (SFCSTEP_TIME.delay) are fulfilled. If the transition condition is not right, the chart is unfreezed and waits for the right transition's condition in order to move onto the next step or steps (in the event of divergence). With the activation of the successor step the chart returns to the freeze state.
Step Out	Only available if the current element is a step belonging to a macro-step section. This command unfreezes the chart (master and macro belong logically to the same chart) in order to work normally over the end of the macro. The process waits for the valid transition condition(s) of the chart in order to move onto the successor step or steps (in the event of divergence). With the activation of the (first) successor step the chart returns to the freeze state.
Set Break On Selection	Sets a breakpoint for all selected steps.
Set Pre Pos On Selection	Sets the pre-positioning markers on the steps and macro-steps of the SFC section.
Set Pre-positioned	Activates all the steps and macro-steps pre-positioned with the **Set Pre Pos On Selection** command.
Select active steps	Selects the active steps.
Set Selected Steps	Activates the selected steps and macro-steps.

SFC section name	
Reset Selected Steps	Deactivates the selected steps and macro-steps. For Multi-Token operating mode only (**Project Settings > Allow multiple token**).
Clear All Breakpoints	Deactivates all the breakpoints.
Clear All Pre-Positions	Deactivates all the steps and macro-steps pre-positioned.

NOTE: The **Step Into** command is not available when debugging an SFC section (chart).

SFC Save and Restore

The `SFC_RESTORE` function block can be used to restart all SFC charts of an application with a given set of active steps at a state, saved before a controller error.

In a recovery situation it is possible to set a group of steps that represent a given state in the process and to continue at that point.

(See SFC_RESTORE: SFC Save and Restore (see EcoStruxure™ Control Expert, System, Block Library) in *System Library*).

Inserting a watchpoint in Sequential Function Chart language

At a Glance

SFC sections do not support watchpoints.

However, a **watchpoint** can be set on an element in the **section that contains processing for a step or transition**; in this case, the procedures correspond to those described for the language used:

- ladder language, page 1419,
- textual languages, page 1427,
- Function Block Diagram language, page 1436.

Information on Debugging

Subject of this Section

This section describes how to access additional information during debugging.

Tracing task execution

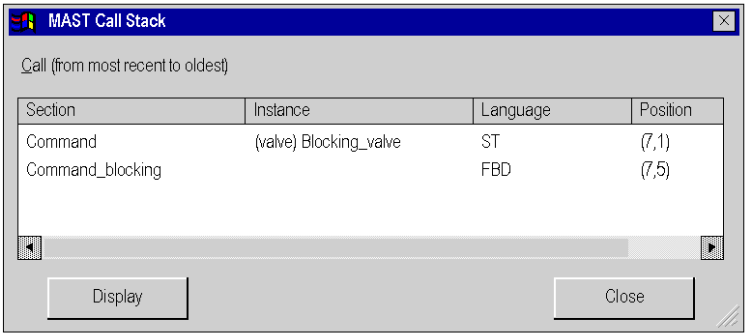
At a Glance

Tracing task execution consists of knowing at a given moment (breakpoint reached, step by step mode in progress) what the task execution path is, that is to say, what **subroutines (SR)**, **user function blocks (DFB)** have been called and what the nesting level is.

For this purpose, a tool is available for your use: the **LIFO stack** (Last In First Out) which memorizes all nestings so that you can monitor task execution.

How to Access a Program Element from the Stack

Carry out the following actions:

Step	Action
1	From a breakpoint or a step-by-step mode in progress either by: • selecting the following command from the menu: **Debug->Call Stack**, • or by pressing the  button in the debug toolbar. Example: **Command_blocking** section having called a **Blocking_valve (valve type)** DFB instance in the **Command** section:  The line number is used for Structured Text and Instruction List editors; for Ladder and FBD editors, this is the number of the rung or the block that is displayed.
2	If you select: • the **Command** section and you click on **Show**, the editor positions itself on the step in progress in the DFB instance; if this is a breakpoint, it is indicated in the left margin by  , • the **Command_blocking** section and you click on **Show**, the editor is positioned on the DFB instance call that is not the step in progress: it is indicated by  .

Clicking the **Close** button closes the window representing the stack: the green triangle disappears following:

- the resumption of step by step,
- a new call to display the contents of the stack,
- a restart of the task  .

Task states during debugging

At a Glance

When debugging a project, it can be useful to determine the current state of tasks it contains. Apart from the master task (MAST), these tasks can include:

- the FAST task,
- the auxiliary tasks (AUX0, AUX1, AUX2, AUX3).

The task state can be:

- **HALT** task stopped following the PLC's switch to "Halt",
- **RUN** task currently being executed,
- **BKPT** task in step by step mode,
- **STOP** task stopped.

How to Determine the Project's Task State

To determine the current task state, select the **Debug->Task State** command from the menu.

Example with a single task (MAST) in step by step mode:

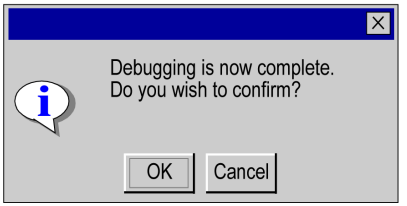


Disconnection During the Debug Phase

Caused by the User

For a variety of reasons, you can decide to switch from **online mode** to **offline mode** even if a breakpoint has been set in a section or if you are in step by step mode.

In this case, you must confirm your choice:



Following confirmation, all application tasks are stopped.

Caused by a Detected Error

When a breakpoint is set in a task section, or when you are in step by step mode, if a hardware or communication problem is detected, the controller switches automatically from **online mode** to **offline mode** and the breakpoint is lost without a request for confirmation.

Operation of the Alarm Relay in Debug Mode

Description

The alarm relay changes status on every step or breakpoint it meets.

At the end of each cycle, the outputs are refreshed.

NOTE: Set the bit %S9 to 1 to force outputs to fallback mode to stop alarm relay changes on every step or breakpoint.

PLC debug screen

Aim of this sub-section

This sub-section shows the debug screen associated with the PLC processor.

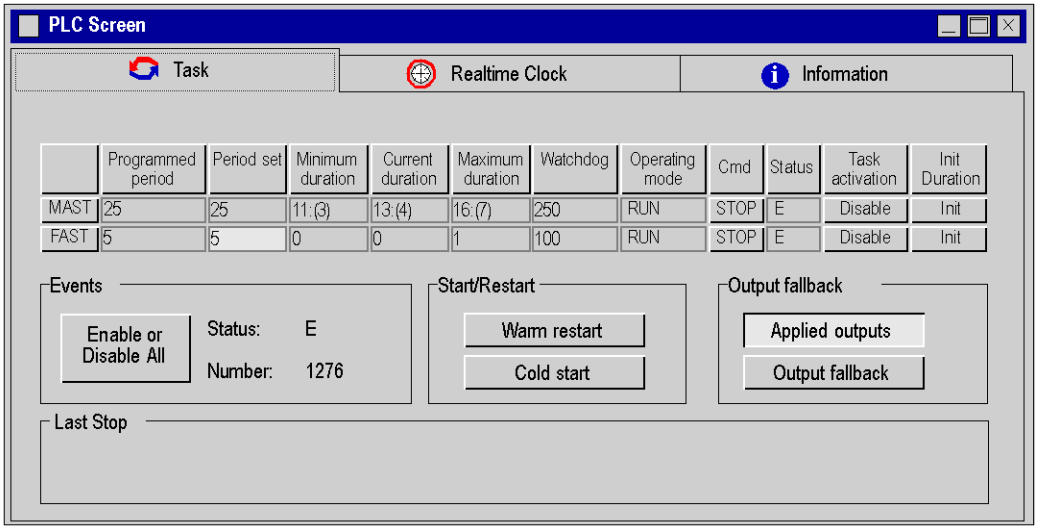
Overview of the CPU debugging screen

Role

This screen is composed of 3 tabs:

- **Task**, statuses and execution commands of the program tasks
- **Real-time Clock**, adjusting the real-time clock
- **Information**, information on the project and the PLC processor

PLC debug screens



NOTE: In the Minimum duration time , Current duration time, and Maximum duration time fields, the values between parenthesis corespond to the Minimum, Current, and Maximum overhead time (see Modicon M340, Processors, Setup Manual).

How to access the PLC debug screen

The CPU debug screen is only accessible in online mode using the **Tools > PLC Screen** command or by double clicking on the PLC processor in the configuration editor and choosing the **Animation** tab.

If the connection is interrupted, the screen remains displayed but is inactive.

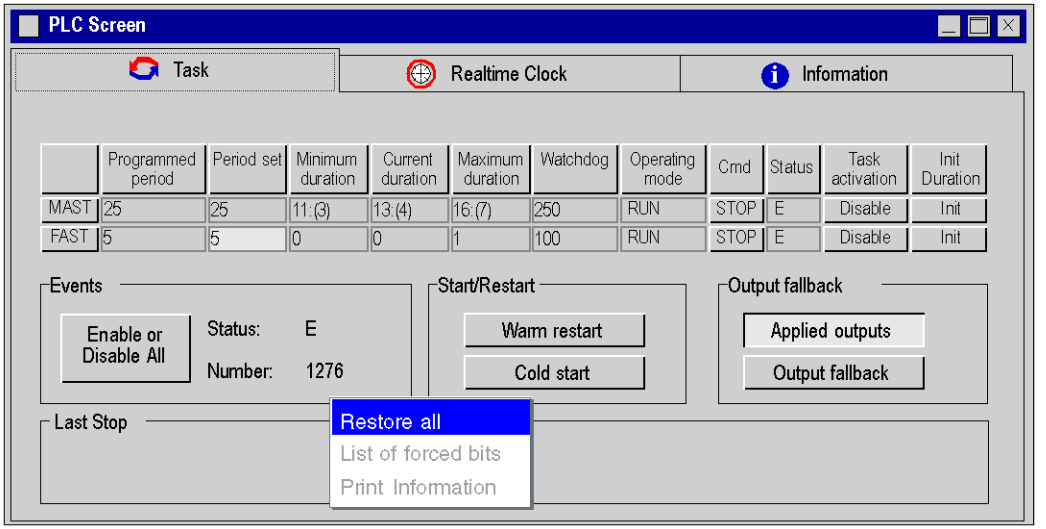
To activate the commands or carry out the adjustments proposed in these tabs, you must have appropriate user rights.

Contextual menu in PLC screen

The contextual menu in the PLC screen displays three options:

- Restore all, page 1456, restore task periods configured for all periodic tasks
- List of forced bits, display list of all forced bits
- Print Information, print data from the information tab

PLC debug screens and its associated contextual menu



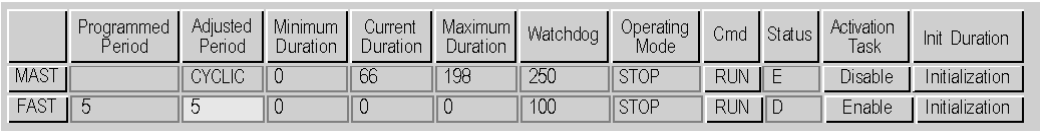
NOTE: The **Restore all** option is only available when Task tab is active and **Print Information** when the Information Tab is active. All the options available in the contextual menu are also available in **Services > Restore** or **Services > Restore all** commands.

Description of the Task tab: Task checking field

At a Glance

This field gives you access to the display and settings of the task periods and the various task execution commands.

Task checking field:



	Programmed Period	Adjusted Period	Minimum Duration	Current Duration	Maximum Duration	Watchdog	Operating Mode	Cmd	Status	Activation Task	Init Duration
MAST		CYCLIC	0	66	198	250	STOP	RUN	E	Disable	Initialization
FAST	5	5	0	0	0	100	STOP	RUN	D	Enable	Initialization

This field may be extended according to the tasks configured in the project: one line per auxiliary task may be displayed in addition to those in the above screen.

Description

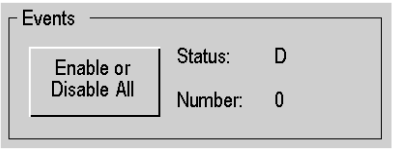
The following table describes the role of each of the screen columns.

Column	Role
Programmed Period	Value of the configured task period (for a periodic task).
Adjusted Period	In periodic operation, enables the task period to be adjusted. This setting is lost on a cold start. The value entered in the configuration is then taken into account. Red background color in a field of this column indicates an erroneous returned by the PLC. It is possible to restore task periods configured for the selected task or for all the periodic tasks with the **Services > Restore** or **Services > Restore all** commands.
Minimum Duration	Minimum duration of execution of the task, measured from the last cold restart or reinitialization using the **Init Duration** command.
Current Duration	Duration of the last task cycle.
Maximum Duration	Maximum duration of execution of the task, measured from the last cold restart or reinitialization using the **Init Duration** command.
Watchdog	Value of the task watchdog. This value may not be modified.
Operating Mode	Operating mode of the task: RUN, STOP, HALT or BKPT (Breakpoint).
Cmd	The **RUN/STOP** buttons associated with the tasks are used to switch to RUN or STOP.
Status	• E (enabled): for active task (the system bits %S30 and %S31 are set at 1), • D (disabled): for inactive task (the system bits %S30 and %S31 are set at 0).
Activation Task	The **Enable/Disable** buttons associated with the tasks are used to enable or disable a task. When a task is **enabled**, the code is scanned and executed. The inputs/outputs are updated. When a task is **disabled**, the code is neither scanned nor executed. The inputs/outputs are still updated.

Column	Role
Init Duration	This command is used to reinitialize the values of the maximum and minimum durations with the current value in order to carry out new measurements.
CPU Load	Only for M580 with SV >= 2.80: percentage of the CPU load used by the task. In addition to the these values the system and total CPU load are given below the tasks checking field. **NOTE:** To perform application transfer in good conditions from primary to standby CPU, it is recommended to not exceed 80% for the total CPU load.

Event Processing

The "Events" field is used to check the execution of event processing



The following table describes the role of each of the columns of the screen above.

Column	Role
Enable or disable all	This button is used to enable or disable all the event processing operations. If the events appear when the processing operations are disabled, they are lost. The master task must be in RUN for this command to be authorized.
Status	indicates the status of the event processing operations: **E**nabled or **D**isabled.
Number	Number of event processing operations executed

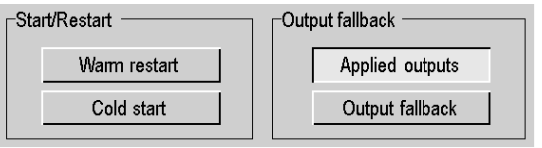
Description of the Task tab: Operating mode field

At a Glance

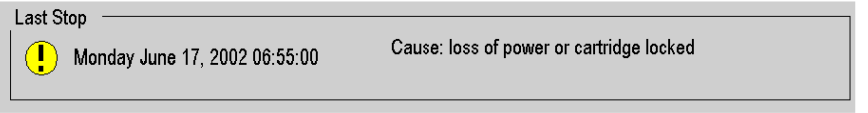
This field gives access to various simulation buttons:

- **Warm restart**,
- **Cold start**,
- **Output fallback**.

Operating Mode field:



This field also indicates the date, time and cause of the last PLC stop.



Description

Buttons and functions:

Button	Function
Warm restart	Click on this button to perform a warm restart of the PLC. This sets the %S1 bit to 1, which, for example, is used to command a partial initialization program. This does not set %S21 and %S13.
Cold start	Click on this button to perform a cold start of the PLC. This initializes the data, the system, and sets the %S0 bit to 1 which (in this case) is used for example to command a specific initialization program. This does not sets %S21 and %S13.
Output fallback	It is used to switch the outputs into fallback mode. The values supplied to the outputs are then the values defined in the configuration (fallback value on PLC STOP). When the outputs are in fallback, the **Applied outputs** button becomes active and allows you to exit fallback mode.
Applied outputs	Is used to stop the fallback mode and to reapply the values supplied by the program to the outputs.

Specific features

Management of output fallbacks for Premium Atrium, M580 and M340 PLCs: when a task is set to STOP, the associated outputs automatically switch to the configured mode

(fallback or maintain). The outputs associated with the other tasks continue to be updated as normal.

For Premium PLCs, during a warm start, an output in fallback or maintain mode goes to the security mode (state 0) while the CPU is not started. When the CPU is started, if the output falback function is still active, the output will return to the fallback mode or stay at 0 in maintain mode.

For M580, refer to chapter *BMEP58xxxx CPU Operating Modes* (see Modicon M580, Hardware, Reference Manual).

For M340 PLCs

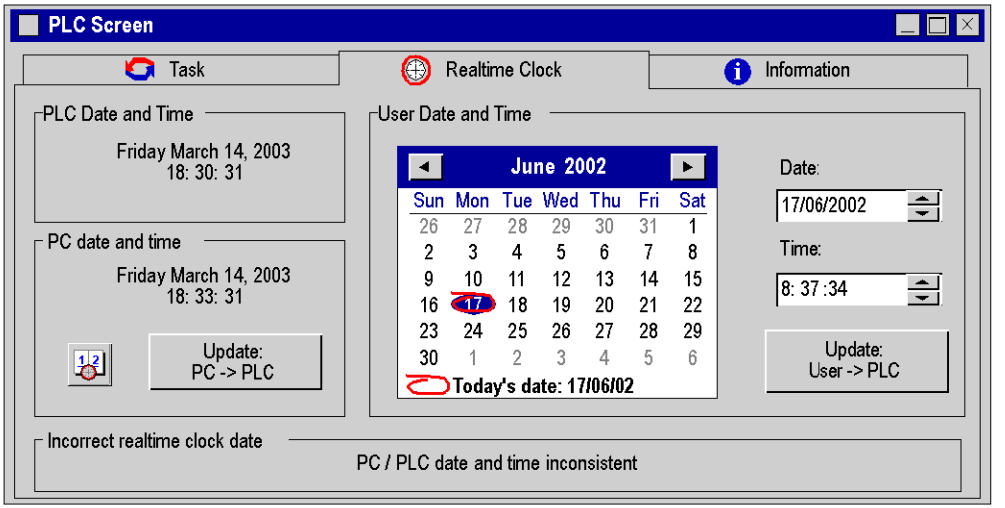
- Whatever the operating mode, an output in fallback or maintain mode do not return to the previous state but stays in security mode (state 0) while the fallback condition is applied.
- The fallback values are only applied if the **Topological** option of **Memory Management** is checked, see in the PLC configuration screen, page 1349.

  NOTE: The buttons **Output fallback** and **Applied Outputs** are not available for Quantum PLCs. On Quantum PLCs, the output of the stopped task will continue to be updated (with the last value written), the others are handled normally. Inputs of the stopped task will continue to be scanned, but the data input will be discarded.

Description of tab: Realtime Clock

At a Glance

This field gives access to the real-time clock settings.



Description

Fields and functions:

Field	Function
PLC Date and Time	Displays the date and time of the PLC processor.
PC date and time	Displays the date and time of the PC. The icon:  is used to access the PC Date and Time setting screen. The **Update PC->PLC** button updates the real-time clock of the PLC in line with the date and time of the PC.

Field	Function
User Date and Time	Is used to adjust the date and time using the calendar and the entry fields: **Date** and **Time**. The **Update User->PLC** button updates the real-time clock of the PLC in line with the date and time set by the user.
Incorrect realtime clock date	This field displays the cause of the error when a date or time is not accepted.

Description of the Tab: Information

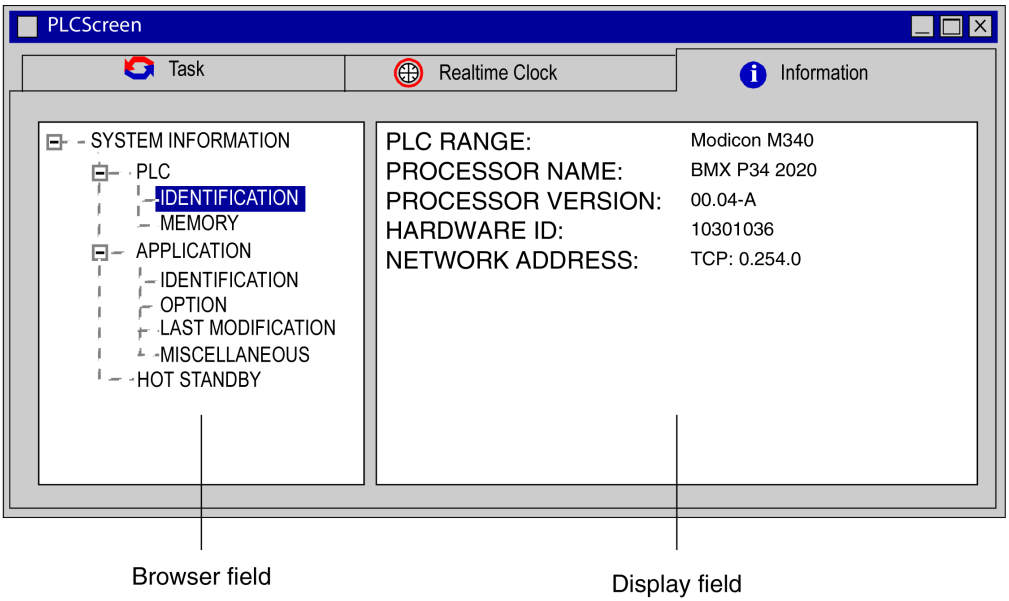
At a Glance

The **Information** tab is composed of 2 areas:

- the browser field, which is used to choose the subject
- the display field displayed the information concerning the chosen subject is displayed.

The **Services > Print Information** command is used to print this information.

Information tab:



NOTE: To view the content of the **Information** tab perform the following actions:

- build the project (**Build Changes** or **Rebuild All Project**)
- connect Control Expert to the PLC (**PLC > Connect**)
- download (**PLC > Transfer Project to PLC**)

PLC IDENTIFICATION Information

This information is used to identify the PLC processor:

Element	Function
PLC RANGE	Type of PLC connected
PROCESSOR NAME	Reference of the connected processor
PROCESSOR VERSION	Software version of the processor
HARDWARE ID	Hardware identifier
NETWORK ADDRESS	Connection address

PLC MEMORY Information

This information is used to identify the types of memory and their size.

Element	Function
RAM CPU	Size of the internal RAM memory
BMX RMSxxx	Part Number of the SD-Card present in the PLC.
SD-Card	A memory card is inside the PLC, but Control Expert cannot read the part number.
RAM CARTRIDGE	Type of the RAM cartridge
FLASH CARTRIDGE	Type and size of the additional RAM cartridge

APPLICATION IDENTIFICATION Information

This information is used to identify the project:

Element	Function
NAME	Project name
CREATION PRODUCT	Software version from which the project has been created
DATE	Project creation date
MODIFICATION PRODUCT	Software version with which the project has been modified
DATE	Project modification date
VERSION	Project version
SIGNATURE	Software signature associated with the project
LIBSET VERSION	**Libset** version used for the project
SOURCE SAFE SIGNATURE	A signature of the several safe parts of the safety PAC application, produced using the SHA256 secure hash algorithm.

APPLICATION OPTION Information

This information describes the content of the project loaded into the PLC.

Element	Function
NEW TERMINAL SUPPORT	Presence or absence of support information for new console (includes "Upload information", "Comments" and "Animation tables").
UPLOAD INFORMATION	Presence or absence of information needed to upload the project to the terminal or to the PLC memory.
COMMENTS	Presence or absence of comments on variables in the PLC memory
ANIMATION TABLE	Presence or absence of animation tables in the PLC memory.
PROTECTION OF SECTIONS	Protection of the project sections
DIAGNOSTICS	Depends on PLC diagnostic project setting: • YES if application diagnostic or system diagnostic is checked • NO if application diagnostic and system diagnostic are unchecked

APPLICATION LAST MODIFICATION Information

This information describes the general information of the project loaded into the PLC.

Element	Function
MODIFICATION PRODUCT	Software version with which the project has been modified.
DATE	Project modification date.
PC NAME	Name of the PC[1]
APPLICATION	Path and name of the project saved in `*.STU`[2]
USER NAME	Display the name of the user profile associated with Control Expert[1]
1 These elements can be hidden if no information are found **2** This element can be hidden if the project has not been previously saved	

APPLICATION MISCELLANEOUS Information

Additional project information

Element	Function
NEW TERMINAL SUPPORT	Update status of support information for new console.
FORCED BITS	Display the number of forced bits in the application. A single click on this line starts the animation table containing all the forced bits. This service is only operational if the content of the PLC is identical to the content of the programming terminal and if there are forced bits. See also Animation Table of Forced Bits, page 1500.

HOT STANDBY Information

Quantum and Premium:	This information displays the Hot Standby status register (`%SW61`). It is used to monitor the current machine status of the Primary CPU and Standby CPU. For details about the bits in this register, refer to *EcoStruxure™ Control Expert, System Bits and Words, Reference Manual*.
Modicon M580:	This information displays the Hot Standby status register part of the Hot standby device DDT T_M_ECPU_HSBY (see Modicon M580 Hot Standby, System Planning Guide for, Frequently Used Architectures).

Viewing and adjusting variables

What's in This Chapter

Introduction to variable animation 1466
Variable Display Window 1467
Variable Display Boxes 1470
Animation Tables 1473
Instruction Editor (LL984) 1501
Modifying values: 1504

Subject of this Chapter

This chapter guides the user through how to implement the tools for viewing and adjusting project variables. These tools are:

- the variable display window,
- the variable display boxes,
- the animation tables.

Introduction to variable animation

Subject of this Section

This section provides an overview of the tools used to view variables.

Displaying variables in debug mode

At a Glance

Variables can be displayed using different language editors (sections), page 1397, and additional tools, which are:

- the variable display window,
- the display boxes,
- the animation tables.

Variables Shown in the Display Window

Certain variables used in a section cannot all be animated directly in the language editor. These are:

- numerical variables in sections that use textual languages (ST, IL),
- numerical variables used in operation and comparison blocks in textual languages (ST, IL),
- structure variables belonging to the DDT family (derived data type),
- parameters and variables belonging to DFB or EFB instances.

These variables are shown in the **display window**, called from the language editors.

The window automatically displays the language editor variables visible onscreen; when the editor runs through the section, the window is updated automatically with the new variables.

Variables Shown in the Display Boxes

Each box displays the name and value of a single variable.

The boxes can be placed anywhere in the section and can be configured.

Variables Displayed in the Animation Tables

Certain variables must be continually monitored, regardless of whether they are visible or not in the language editor at a given time.

These variables are displayed in the **animation tables**, called from the language editors.

Variable Display Window

Subject of this Section

This section guides the user through the implementation of the variable display window.

Introduction to the Variable Output Window

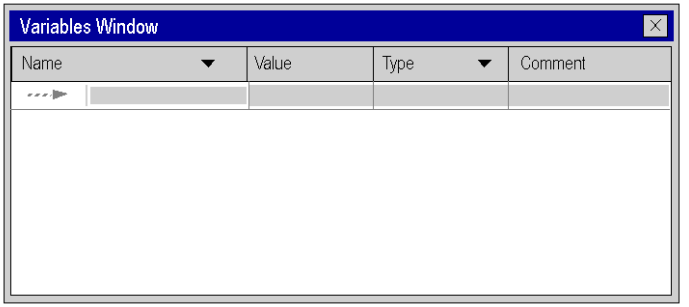
Opening the Tool

The Output Window is launched from the menu with the

"Tools->Variables Window" command.

In online mode, if the language editor is open, the tool automatically contains the variables displayed by the language editor.

Opening in offline mode:



The Output Window includes:

- a line for each variable,
- for each one of the variables, columns describing:
  - their name (symbol or address),
  - their value,
  - their type,
  - a comment.

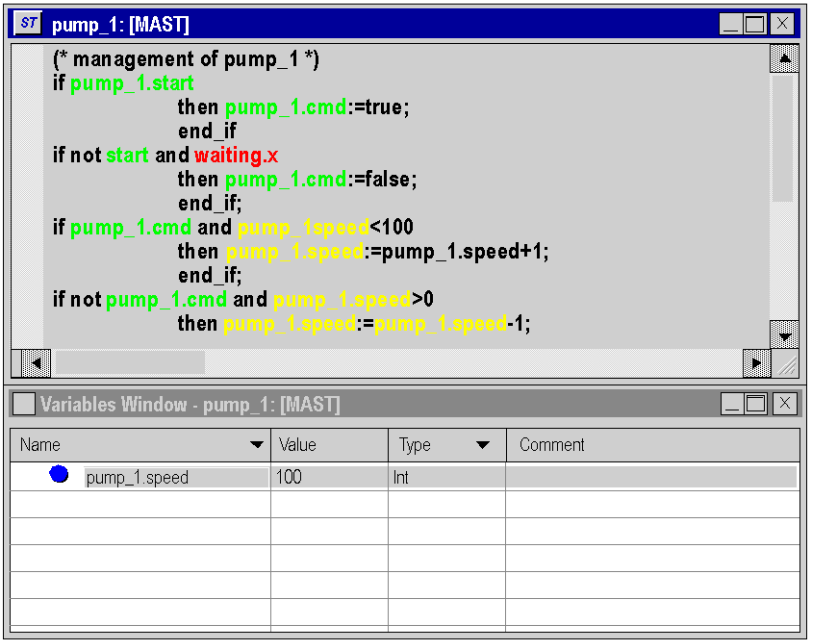
The Output Window and the Textual Language Editors

A textual language editor (Structured Text, Instruction List) animates Boolean data.

Animation of other types of data takes place in the **Variables Window**; this data includes:

- elementary numerical types,
- derived type (DDT) instance variables, except Boolean types,
- function block (EFB,DFB) instance variables, except Boolean types.

Example with Structured Text editor:



NOTE: to display variables of the derived data (DDT) or function block data (EFB\DFB) types, you must expand (+) the name of the corresponding instance.

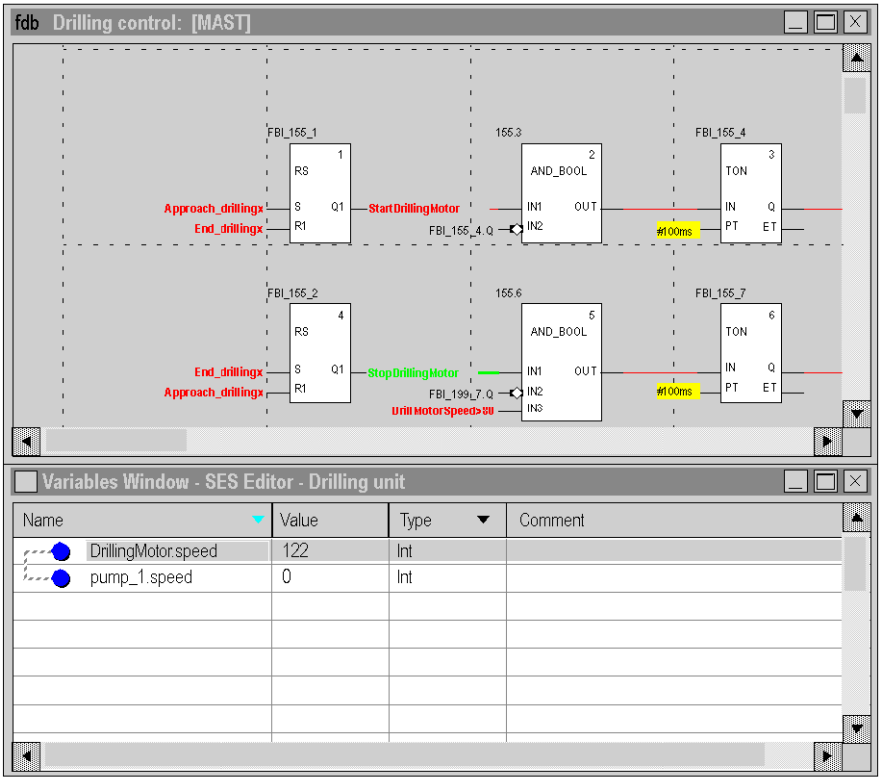
The Output Window and the Graphical Language Editors

A graphical language editor (LD,FBD) animates Boolean and elementary numerical types of data.

Animation of other types of data can be carried out in the **Output Window**; this data includes:

- derived type (DDT) instance variables, except Boolean types,
- function block (EFB,DFB) instance variables, except Boolean types,
- numerical variables used in the operation or comparison blocks in the Ladder Diagram language editor,
- variables for numerical expressions connected to EFB or DFB in the Ladder or Function Block Diagram language editor.

Example with Function Block Diagram editor:



NOTE: to display variables of the derived data (DDT) or function block data (EFB\DFB) types, you must expand (+) the name of the corresponding instance.

Variable Display Boxes

Subject of this Section

This section guides the user through the implementation of the variable display boxes.

Introduction to variable display boxes

At a Glance

Each variable display box displays the name and value of a variable used in the section.

Main characteristics:

- they are called only via sections written in textual languages (Structured Text, Instruction List),
- they contain the variable name and value,
- the boxes can be placed anywhere in the section,
- they can be configured.

The display boxes are implemented in offline or online mode.

Creating a Variable Display Box

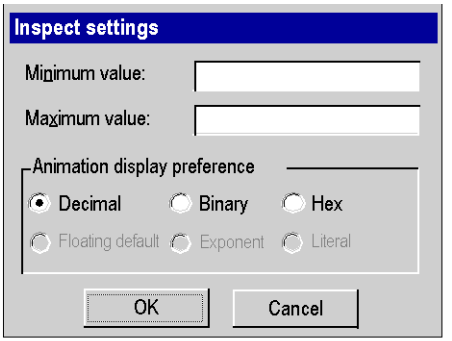
Carry out the following actions:

Step	Action
1	Select the variable from a section (Structured Text or Instruction List).
2	From the contextual menu, select the **"New Inspect Window"** command; the box is displayed in the section.
3	Use the mouse to position the box.

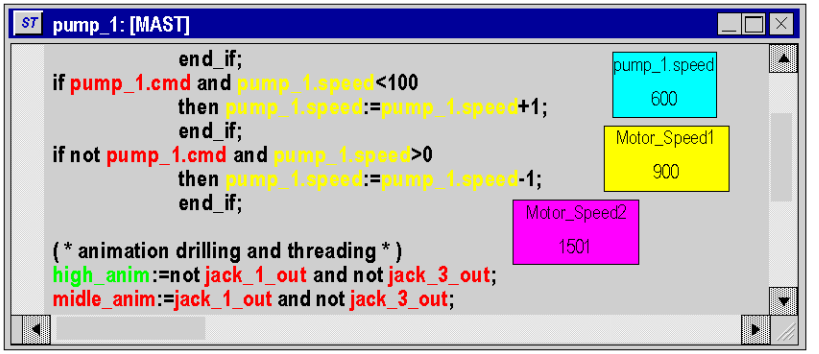
Settings

Carry out the following actions:

Step	Action
1	Select the display box.
2	From the contextual menu, launch the **"Settings"** command.

Step	Action
3	The following dialog box appears:  Enter the minimum and maximum values for the variable: • if the variable value is less than the value entered in **Minimum value**, the display box is shown in **yellow**, • if the variable value falls between the 2 values entered, the display box is shown in **blue**, • if the variable value is greater than the value entered in **Maximum value**, the display box is shown in **pink**. If no value is entered, the box is shown in **blue**.
4	Select the type of variable display to be used in the display box (Decimal, Binary, Hexadecimal, floating default, exponent, literal) by checking the corresponding box.

Example:



Clearing a Variable Display Box

Carry out the following actions:

Step	Action
1	Select the display box.
2	From the contextual menu, launch the **"Delete Inspect"** command.

Animation Tables

Subject of this Section

This section guides the user through the implementation of animation tables.

Permanent and Temporary Animation Tables

Overview

There are 2 kinds of animation tables:

- permanent animation tables
- temporary animation tables

Permanent Animation Table

A permanent animation table is an integrated part of a project.

It is saved with the project and can be exported.

Temporary Animation Table

A temporary animation table

- is not saved with a project but deleted when closing a project

  Therefore no temporary animation table is available in the project browser, when opening a project.
- does not modify a project
- does not modify the build state of a project
- does not create upload information (and therefore can not be uploaded)

- can not be exported
- is not included in the printout of a project

Permanent/Temporary

- Creating an animation table via **Project Browser > Animation Table** you can select if a permanent or a temporary animation table should be created. To create a temporary animation table set the **Temporary Table** checkbox in the **Properties** dialog.
- Creating an animation table outside the project browser (e.g. via **Initialize Animation Table** in a language editor or creating an animation table of forced bits) you will always create a temporary animation table.

  See also Animation Table of Forced Bits, page 1500.

Switching Between Permanent and Temporary

Every temporary animation table (except the animation table of forced bits) can be changed to a permanent animation table and vice versa.

There are 2 ways to do this:

- Use the checkbox available in the **Properties** dialog of an animation table (see Animation Table Properties, page 1480).
- Use the  button top right at the animation table. Click the button to change from a permanent animation table to a temporary animation table and vice versa.

Context Menu

The context menu that can be opened via **Project Browser > Animation Tables** provides 2 additional entries:

- **Make All Tables Permanent**

  Use this context menu entry to change all temporary animation tables to permanent animation tables.

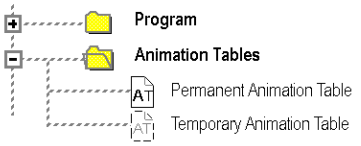
  As mentioned above, an animation table of forced bits is always a temporary one and can not be changed to a permanent.

- **Purge Temporary Tables**

  Use this context menu entry to delete all temporary animation tables.

Icons in the Project Browser

In the **Project Browser** the 2 kinds of animation tables are represented by different icons.



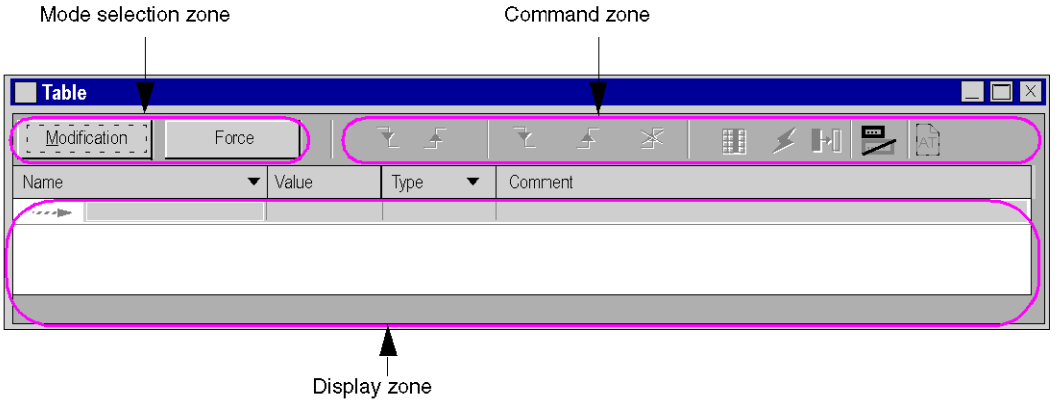
Introduction to Animation Tables

Overview

An animation table is divided into 3 areas that include:

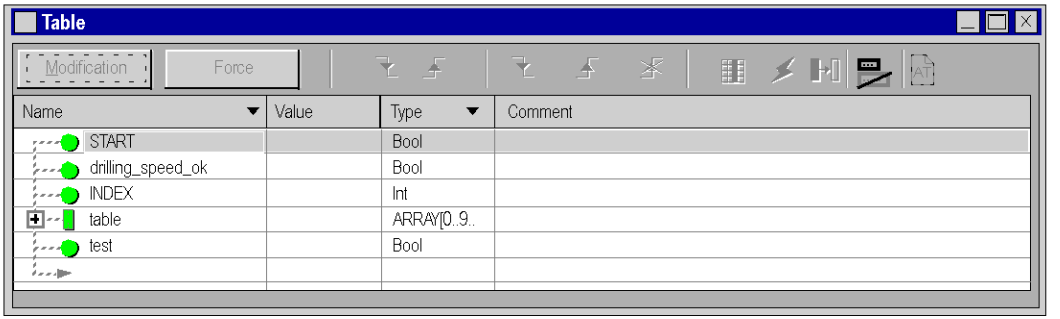
- the **Mode** area,
- the **Command** area,
- the **Display** area,

Animation table:

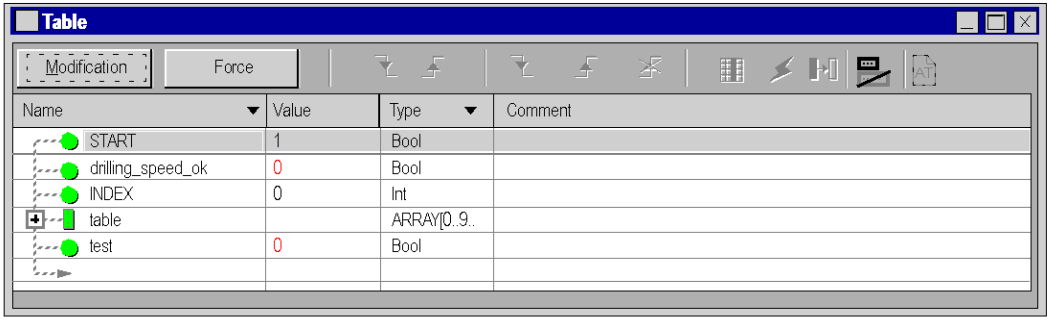


Online and Offline Mode

In offline mode, an animation table does not display values associated with the variables:



In online mode, an animation table displays the values of the variables.



Managing Animation Tables

Creating a New Animation Table

Carry out the following actions:

Step	Action
1	In the project browser, go to the **Animation Tables** directory.
2	From the contextual menu; select the **New Animation Table** command; the table is created with a default name.
3	If you wish, you can rename the table by clicking on its name in the project browser.

Opening an Existing Animation Table

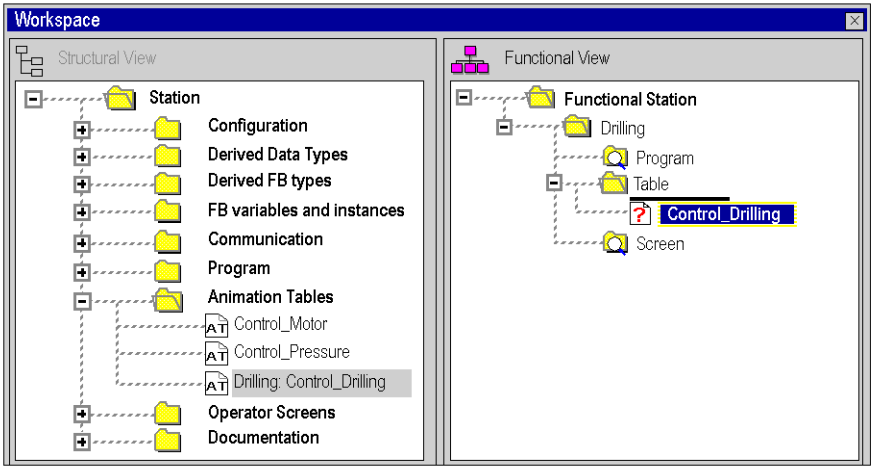
Carry out the following actions:

Step	Action
1	Select the desired table in the **Animation Tables** directory.
2	Either: • Select the **Open** command from the contextual menu, • or double-click on the name of the table.

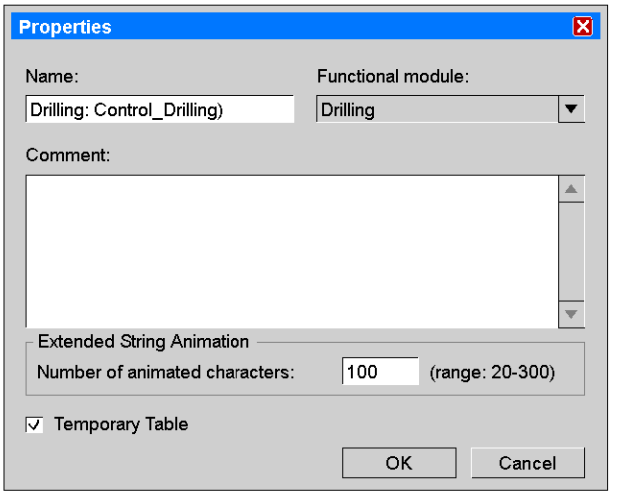
Variables that were displayed previously in the table are re-loaded into the table.

Attaching an Animation Table to a Functional Module

Carry out the following actions:

Step	Action
1	Display the **Structural View** and the **Functional View**.
2	Select the desired table in the **Animation Tables** directory in Structural View.
3	From the Structural View, use the mouse to **drag\drop** the animation table into the desired functional module in the Functional View. The authorized insertion point of the animation table in the functional module, is represented by a line. Example: 

Another way to attach an animation table to a functional module is described in the following table:

Step	Action
1	Display the **Structural View** .
2	Select the desired table in the **Animation Tables** directory in Structural View.
3	Select the **Properties** command from the contextual menu.
4	Select the desired functional module from the **Functional Module** data selection box. Example: 
5	Confirm with **OK**.

Backing Up an Animation Table

The backup is performed automatically.

Deleting an Animation Table

Carry out the following actions:

Step	Action
1	Select the desired table in the **Animation Tables** directory.
2	Select the **Delete** command from the contextual menu.

Properties of the Animation Table

To access the properties of the animation table, carry out the following actions:

Step	Action
1	Select the desired table in the **Animation Tables** directory or in the editor window.
2	Either: • Select the **Properties** command from the contextual menu, • or click **Edit > Properties** in the menu bar. Refer also to Animation Table Properties, page 1480.

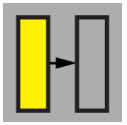
Multiple Modifications of Animation Table Variables

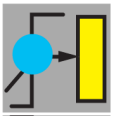
Entering Multiple Mode

To enter this mode follow this procedure:

Step	Action	Icon
1	In the animation table dialog click the **Modification** button:	
2	Click the **Enter Multiple Mode** button:	

Current and Set Values

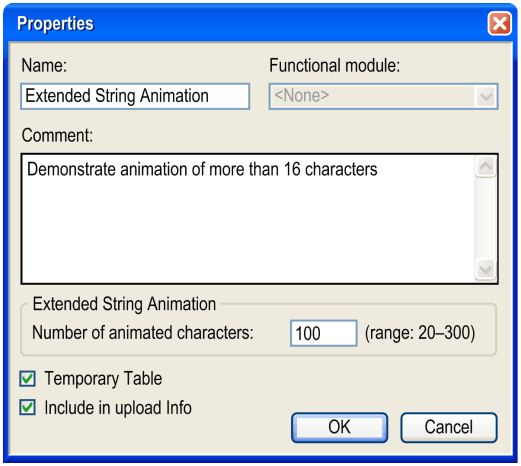
Update **Current Values** in the PLC with **Set Values** by clicking:	
Reset all **Set Values** by clicking:	
Save, import and export animation tables with all their **Set Values**.	

Populate the **Set Values** with the **Current Values** by only one click on:	

NOTE: This feature allows users to manage animation tables as "recipes".

Animation Table Properties

Presentation



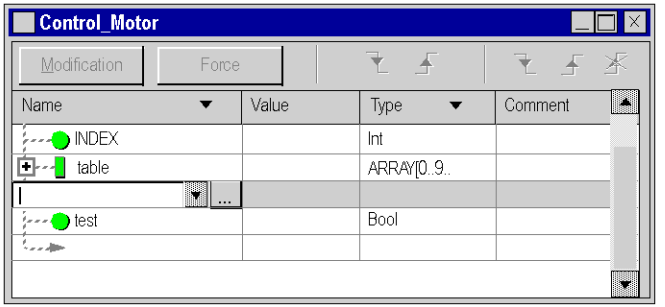
Animation Table Elements

Element	Description
Name	Animation table name
Functional module	Attached functional module
Comment	Animation table comment
Number of animated characters	Number of characters that can be animated in the **Extended Strings** mode (range: 20 ... 300) Refer also to Animation of Strings in the Animation Table, page 1485.
Temporary Table	Type of animation table (temporary/permanent) See also Permanent and Temporary Animation Tables, page 1473.
Include in upload info	The animation table (including comments) is stored in the *.STU or *.STA application file. This box is checked by default and is: • enabled when **Animation Tables** in the **Project Settings** dialog is **also** checked • **not** enabled when **Animation Tables** in the **Project Settings** dialog is **not** checked

Basic functions

Adding a Data Item to the Animation Table

Carry out the following actions:

Step	Action
1	Move to an empty line
2	Either you can: • double-click on the empty line • move to a variable and from the contextual menu, by selecting the **"Insert"** command • type the variable address Example: 
3	Enter the variable name or select it in the **"Instance selection"** window launched with the  button.

NOTE: Variables previously created in the Data Editor, page 348 can be added to the Animation table by entering the address in the **Name** element of the Animation Table.

Adding a Data Item From Another Animation Table

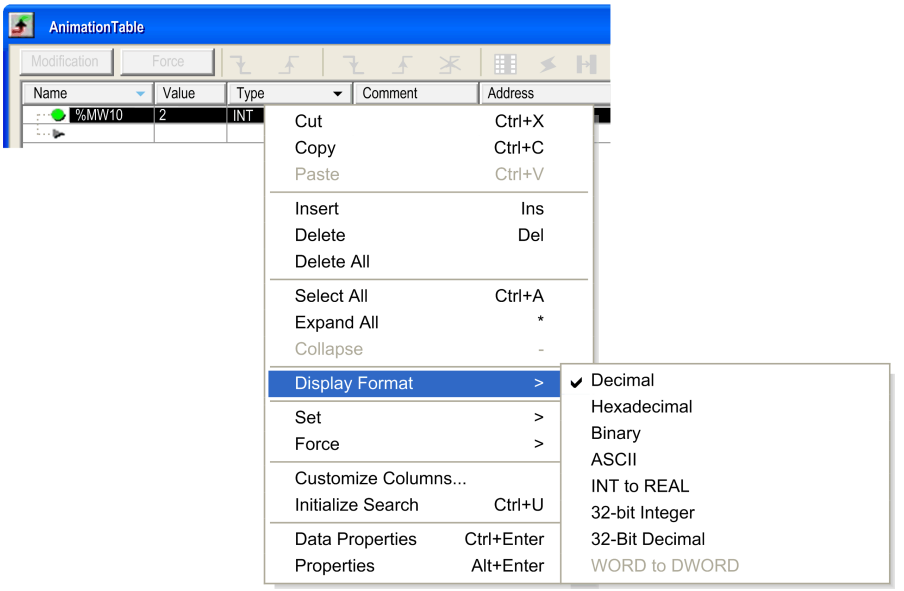
Carry out the following actions:

Step	Action
1	In the **source animation table**, select the desired data item.
2	Either: • select the **"Copy"** command from the contextual menu, and in the **destination animation table**, in the desired line, select the **"Paste"** command. • or using the mouse, **drag** the data item and **drop** it in the desired line in the **destination animation table**. **NOTE:** The drag/drop operation removes the data from the source table and places it in the destination table, whereas Copy/Paste copies the data from the destination table without removing it from the source table.

NOTE: You can also carry out the same manipulation for multiple selections.

Modifying the Display Format

Carry out the following actions:

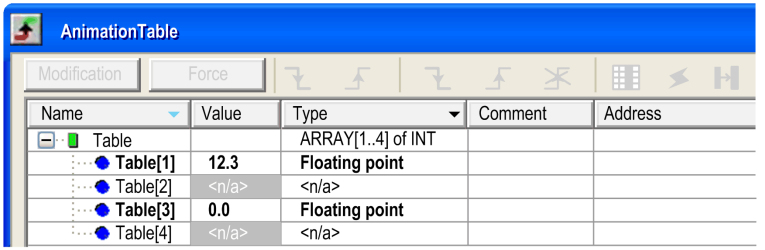
Step	Action
1	In the animation table, select the desired data item.
2	In the contextual menu, select the **"Display Format"**' command and choose the format. Example: 

NOTE: Depending on the display format selected, (Binary, Hexadecimal, ASCII, etc.) the value of the data item is preceded by a **prefix** and followed by a **suffix** (except for Decimal format).

16-bit Data Type arrays (of 2,4,… even) can be displayed as 32-bit formats:

- Array of UINT or INT can be displayed as REAL, 32-bit Integer or 32-Bit Decimal.
- Array of WORD can be displayed as DWORD array.

When a double format is selected, the 32-bit value is written in the field **Value** of the first 16-bit variable and <n/a> is written in the field **Value** of the second 16-bits variable as illustrated below:



NOTE: Only the 32-bit value can be modified directly in the animation table.

%MW<i> variable can be displayed as 32-bit format. Right click on the %MW<i> variable then select the desired double format and the value displayed for %MW<i> is the 32-bit data calculated form %MW<i> to %MW<i+1>.

Deleting a Variable

Carry out the following actions:

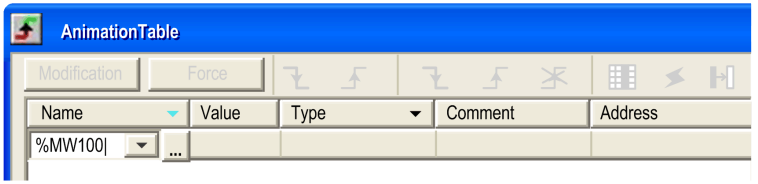
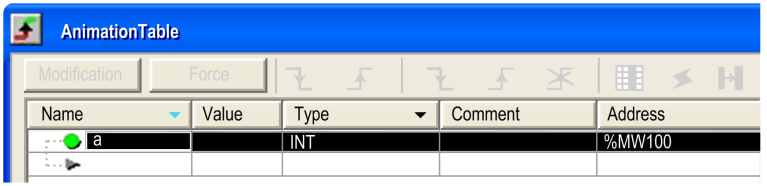
Step	Action
1	In the animation table, select the variable(s) desired.
2	Select the **"Delete"** command from the contextual menu.

NOTE: Once the deletion has been made, the table is reorganized to fill the empty lines.

Displaying a Variable Name

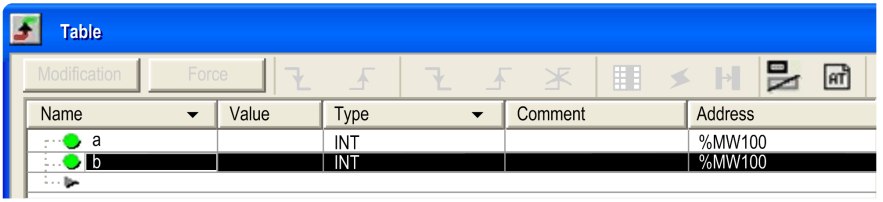
Entering the variable address in the **Name** element displays automatically the variable name.

Carry out the following actions:

Step	Action
1	Move to an empty line
2	Either you can: • double-click on the empty line • type the variable address Example:  **Result**: Animation table automatically displays a in the **Name** element: 

NOTE: It is possible to have more than one variables with the same address entered in the **Name** element. The Animation Table will create an new entry for each variables.

The graphic below shows two variables at the `%MW100` address:



Animation of Strings in the Animation Table

Overview

The animation table offers two different modes (**Modification**/**Force**) to animate strings.

Normally only the first 16 characters of a string are animated (modified).

The reason for this restriction is that for animation handling a PLC reserves a buffer.

The size of this buffer is limited and the animation of complete strings can exceed the size of the buffer easily.

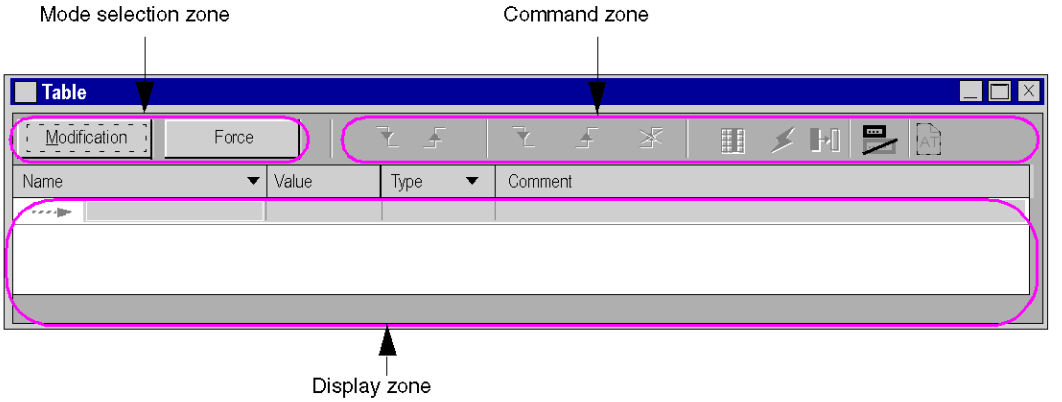
Extended Strings

The animation table can animate more than 16 characters of a string.

At the right side of the command zone area there is the checkbox  (Extended Strings).

It can be used to activate the mode **extended string animation**.

Presentation



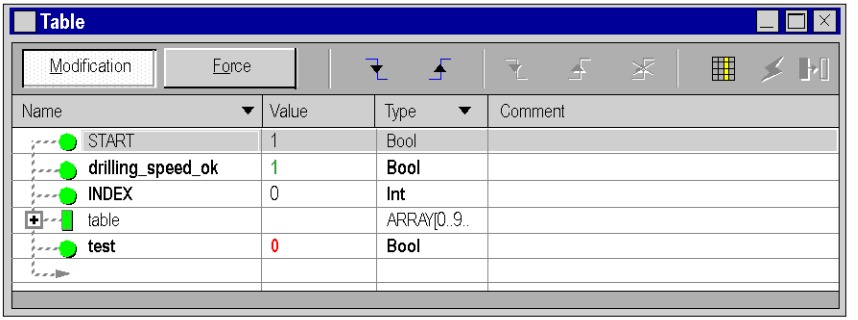
The number of characters to be animated in the mode **extended string animation** can be set in the properties of the animation table (see Animation Table Properties, page 1480).

NOTE: Using this feature you should keep in mind that the animation of big strings can reduce the number of sections and animation tables which can be animated at the same time.

Modification Mode

Implementing Modification Mode

Press the **Modification** button in the Mode area:

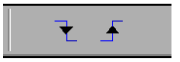


Modifying an Elementary Variable (EDT)

To modify an **immediate value**, carry out the following actions:

Step	Action
1	Double-click in the **"Value"** column on the line corresponding to the variable that you wish to modify.
2	Use the keyboard to enter the desired value.
3	Confirm your choice by pressing the **"Enter" key.**

To modify a **Boolean value**, carry out the following actions:

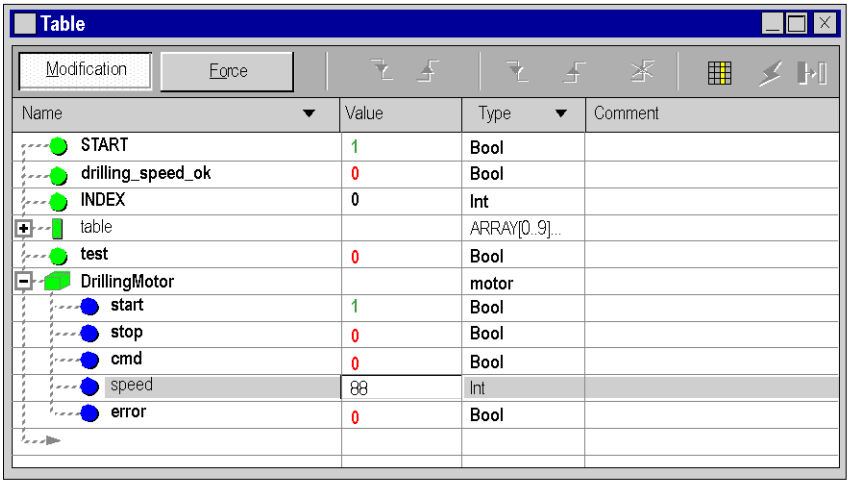
Step	Action
1	Use the mouse to select the Boolean value to be modified.
2	Press one of the buttons  corresponding to the desired value, or from the contextual menu, execute the **"Set to 0" or "Set to 1""** commands. To modify several variables simultaneously, see Modifying Several Boolean Variables, page 1497

Modifying a Derived Variable (DDT)

Derived variables correspond to structures and to tables.

For a structure, if the elements that compose it are elementary variables, modifications are made element by element see Modifying an Elementary Variable (EDT), page 1487.

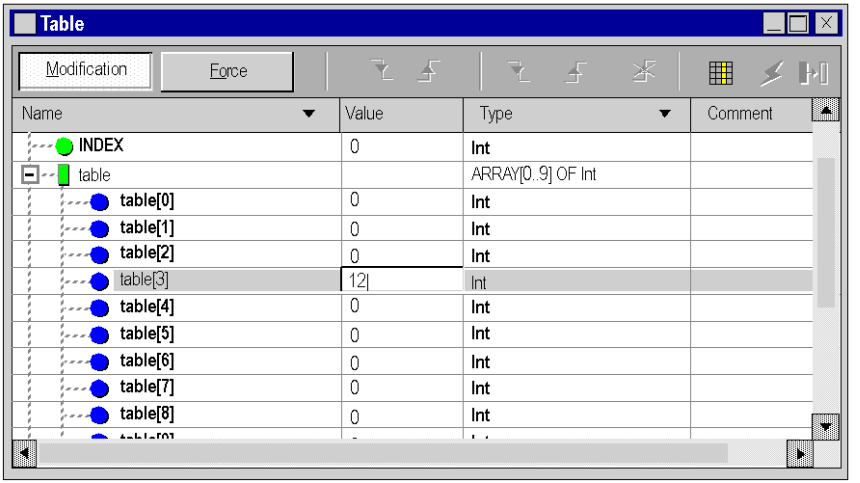
Example:



NOTE: if the structures are nested, you must expand them (+) until you reach the elementary variables.

For a table, if the elements that compose it are elementary variables, modifications are made element by element see Modifying an Elementary Variable (EDT), page 1487.

Example:



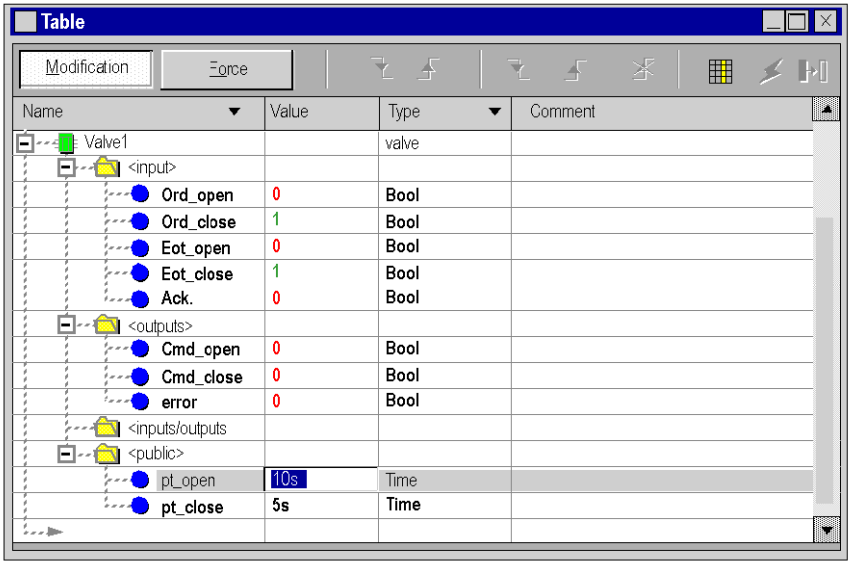
NOTE: If the table contains structure-type elements, follow the procedure described above.

Modifying a Function Block (FB) Data Item

Function block data corresponds to DFBs and EFBs.

If the elements that compose the function block (input/output parameters, public variables) are elementary variables; the modifications are carried out element per element see Modifying an Elementary Variable (EDT), page 1487.

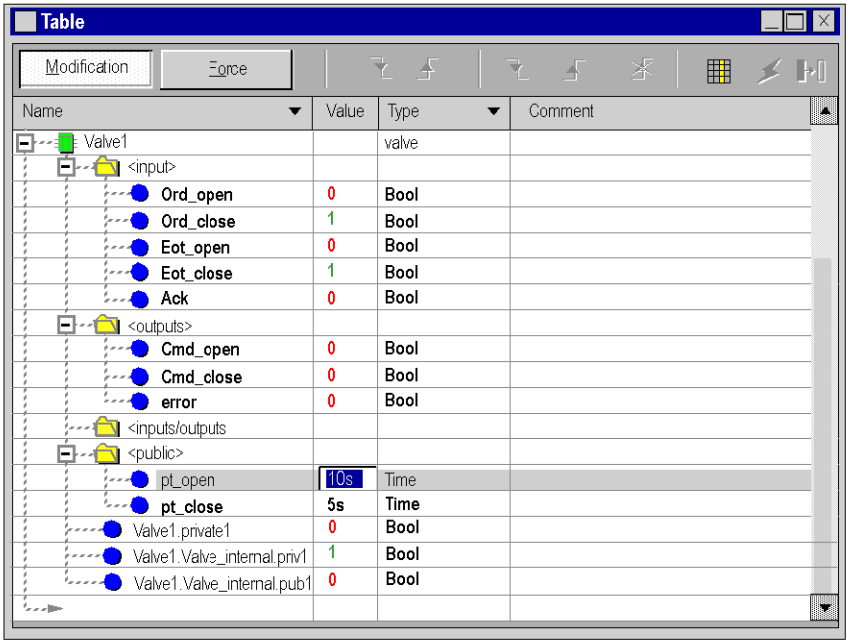
Example of a DFB instance:



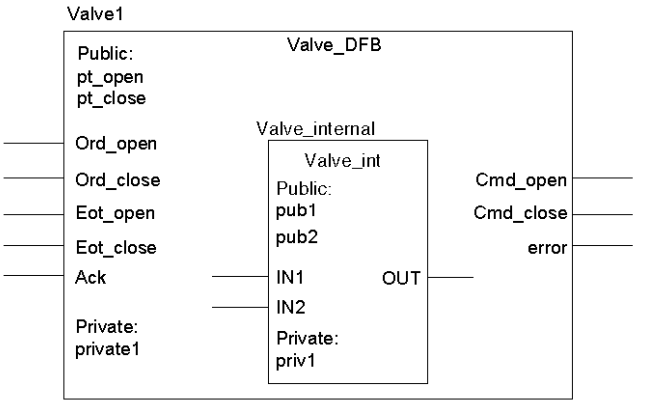
NOTE: If the variables are derived (structures, tables), you must expand them (+) until you reach the elementary variables.

Private variables of derived function blocks cannot be accessed by the pre-defined elements. To display or modify private variables you have to enter them variable by variable at the end of the function block tree in the animation table. With the same syntax is is also possible to enter variables that are used in nested derived function blocks.

Example of a DFB instance with private variables:



Example of nested DFB with public and private variables:



Force Mode

Precautions

Forcing input and output values in a running controller can have serious consequences to the operation of a machine or process. Only those who understand the implications in the controlling logic, and who understand the consequences of forced I/O on the machine or process, should attempt to use this function.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

Ascertain the effect of forcing I/O or forcing application variable values prior to executing the forcing.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

NOTE: Before forcing outputs, understand that forced outputs (%M) exhibit different behavior with Modsoft/NxT/Concept than Control Expert.

- With Modsoft/NxT/Concept you **cannot** force outputs when the Memory Protect switch of the Quantum controller is set to the "On" position.

  With Control Expert you **can** force outputs even when the Memory Protect switch of the Quantum controller is set to the "On" position.
- With Modsoft/NxT/Concept forced outputs **maintain** their values following a cold start.

  With Control Expert forced outputs **lose** their disabled state following a cold start.

Preset Force Values in Offline Mode

The **Modification Values** and **Force Values** for variables can be preset in offline mode via animation tables. This information can be transferred to controller later.

Presetting values can also be used for multiple modification and forcing of Boolean variables, page 1497.

Preset Force Values in Project Information

The preset force values of variables are included in the

- download (**Transfer Project to PLC**) information of a project
- upload (**Transfer Project from PLC**) information of a project.

The **Modification Value**/**Force Value** attribute of a variable is part of the animation table information.

Include in Upload Information

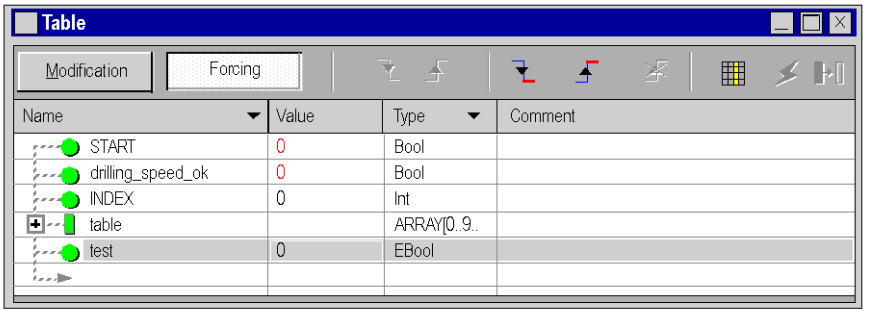
In the properties dialog of an animation table you can activate/deactivate an additional property (**Include in Upload Information**).

As a default this property is activated.

This check box in the property dialog of an animation table is enabled only, if **Animation tables** is activated via **Project Settings > General > PLC embedded data > Upload information**.

Implementing Forcing Mode

Press the **Force** button:



This forcing mode is only available for variables responding to these conditions:

- The variable is an EBool type.
- The forcing attribute is validated in the variable editor.
- The variable is a located variable.
- The variable is an unlocated variable with the following restrictions:
  - For Modicon M580 with firmware version 2.10 or any subsequent supporting version (s), the unlocated variable can be in an EDT or in a Device DDT.
  - For Modicon Quantum 140CPU65150, 140CPU65160, 140CPU65260, 140CPU65860, 140CPU67060, 140CPU67160, 140CPU67260, 140CPU67261, or 140CPU67861 with firmware version 3.0 or any subsequent supporting version(s), the unlocated variable can only be in a Device DDT (X80 modules in EIO drops).
  - For Modicon M340 and Momentum, the unlocated variable cannot be forced.

    NOTE: Forcing or unforcing unlocated variables does not update %SW108. (See the EcoStruxure™ Control Expert, System Bits and Words, Reference Manual).

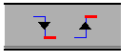
Analog I/O variables forcing is managed in a different way:

- On analog Device DDT (see Modicon X80, Analog Input/Output Modules, User Manual), a set of sub fields (for example EIO2_d1_r0_s1_AMI0800.ANA_CH_IN[0].ANA.FORCE_CMD) allows to force analog I/Os with a predefined value.
- For Modicon M580 and M340 controllers, the analog I/Os variables can be forced in the program.

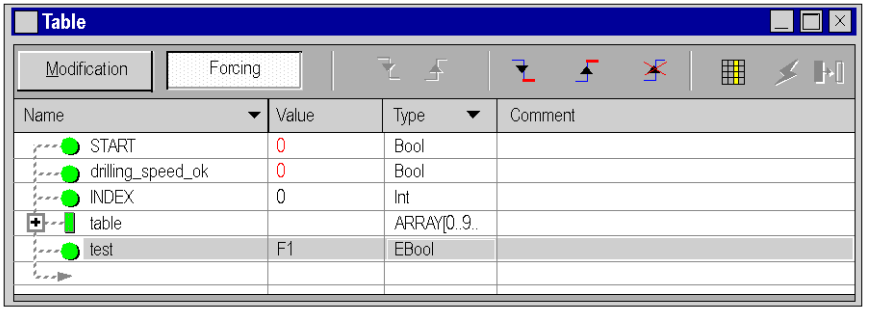
  NOTE: With the animation table, it is not possible to set or reset the bit associated to step S.x.

Forcing a Variable

To force a located Boolean variable, carry out the following actions:

Step	Action
1	Use the mouse to select the Boolean variable.
2	Press one of the buttons  corresponding to the desired value, or from the contextual menu, execute the **Force to 0** or **Force to 1** commands. To force several variables simultaneously, see Forcing several located variables, page 1499.

When a variable is forced, the letter **F** is displayed in front of the value:



Unforcing a Variable

To unforce a located Boolean variable, carry out the following actions:

Step	Action
1	Use the mouse to select the Boolean variable.
2	Press the  . button or execute the **Unforce** command from the contextual menu. To unforce several variables simultaneously, see Unforcing several located variables, page 1499

When a variable is unforced, the letter **F** no longer appears in front of the value.

Creating an Offline Forcing Animation Table

To create a new animation table of forced bits from scratch, carry out the following actions:

Step	Action
1	Create a new (permanent) animation table via the **Project Browser**.
2	Enter the memory bits you want to force either as single bits (like %M1) or as a bit array (like M1:100).
3	Press the **Force** button.
4	Press the **Swaps to multiple mode**  button.
5	Enter forcing values **0** or **1** in the **Force value** column.

To create a new animation table of existing forced bits in the controller, carry out the following actions:

Step	Action
1	There are 3 ways to open the **Forced bits** animation table: • In the status bar click the red **F**. • In the **Project Browser** right-click **Animations Tables** and select **Open Forced Bits Table** from the context menu. • Via **Tools > PLC Screen** open the **PLCScreen** dialog. In the **Information** tab under **APPLICATION > MISCELLANEOUS** click the **FORCED BITS** line.
2	Since this animation table is temporary only, copy and paste the animation table via context menu of the **Project Browser**.
3	Make the copied animation table permanent (using the **Temporary table** button ).
4	In the **copied Forced bits** animation table press the **Force** button.
5	Press the **Swaps to multiple mode**  button.
6	Press the **Populates multiple values using existing forced bits**  button. **Result**: The **Force value** column is populated.

Build/Build changes and **Save** your application.

By building, the values entered in the **Force value** column will be included in the upload information which is downloaded into the controller.

Forcing by Use of an Offline Forcing Animation Table

To force values before starting the controller, carry out the following actions:

Step	Action
1	Open an existing application (**.STU*, **.STA* or an uploaded application).
2	Connect to the controller.
3	Open the manually created/copied **Forced bits** animation table (see Creating an Offline Forcing Animation Table, page 1495).
4	Press the **Force** button to enter forcing mode.
5	Press the **Swaps to multiple mode**  button.
6	Press the **Execute multiple**  button to force the bits from the preset **Force value** column.
7	Now you can start your application.

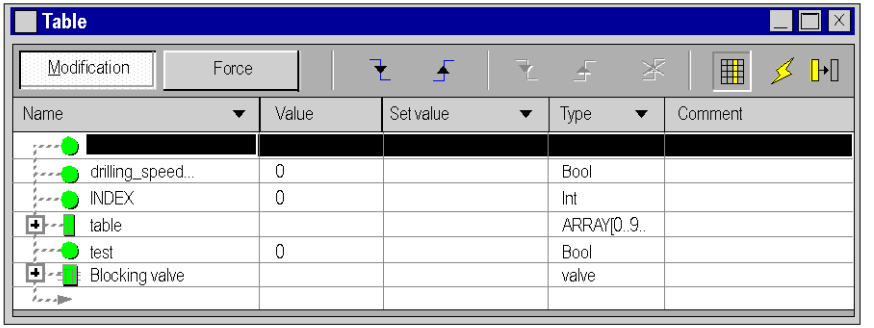
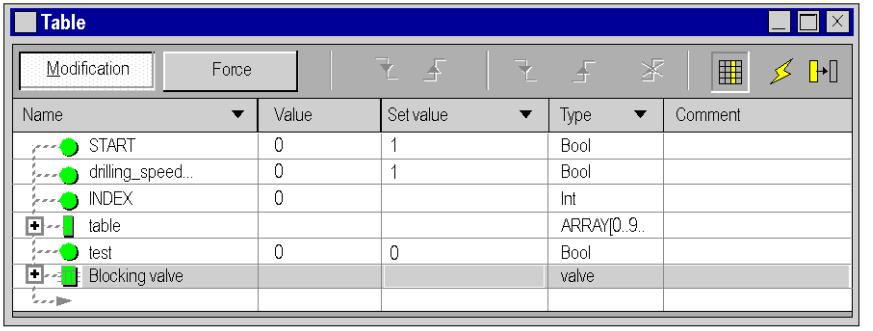
Multiple modification and forcing of Boolean variables

At a Glance

The reason for modifying or forcing a group of several Boolean variables (located) is to enable the PLC to acknowledge these changes in **the same cycle**.

Modifying Several Boolean Variables

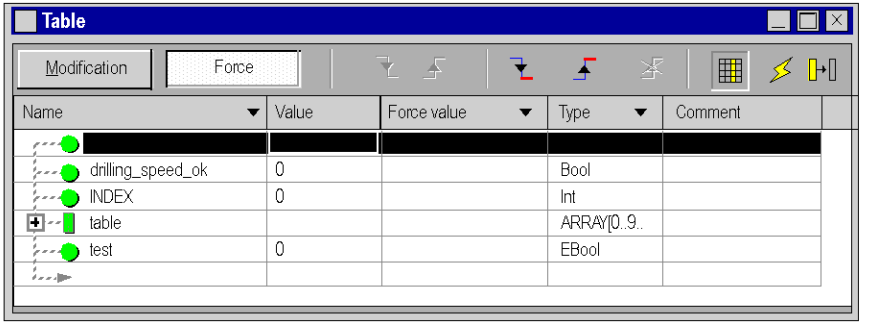
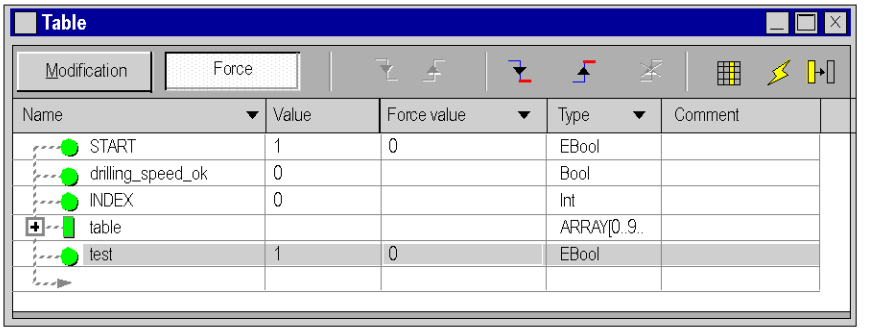
Carry out the following actions:

Step	Action
1	Click on the  button. The **Set value** column is added to the table: 
2	In the **Set value** column, modify the value of the Boolean variables you wish. The contents of the corresponding lines is now shown in bold font: 
3	Press the  button so that the PLC acknowledges the modifications.

The  button clears the **Set value** column.

Forcing Several Boolean (EBool) Variables

Carry out the following actions:

Step	Action
1	Click on the  button. The **Force value** column is added to the table: 
2	In the **Force value** column, modify the value of the Boolean variables you wish. The contents of the corresponding lines is now shown in bold font: 
3	Press the  button so that the PLC acknowledges the modifications.

The  button clears the **Force value** column.

Animation Table of Forced Bits

Overview

You can create an animation table that includes all forced bits of a project.

Table[Forced Bits]

An animation table of forced bits

- can not be modified by you
- is updated automatically if a variable or address becomes forced/unforced or if the value of a forced variable/address changes
- is always created as a temporary animation table and can not be changed into a permanent animation table (see also Permanent and Temporary Animation Tables, page 1473)
- has a fixed name (**Table[Forced Bits]**) which can not be modified
- displays the symbolic names of the forced addresses (if mapping is available

Creating the Animation Table

There are 3 ways to create the animation table of forced bits:

- Via **Tools > PLC Screen** open the **PLCScreen** dialog.

  In the **Information** tab under **APPLICATION > MISCELLANEOUS** click the **FORCED BITS** line.
- In the **Project Browser** right-click **Animations Tables** and select **Open Forced Bits Table** from the context menu.
- In the status bar click the red **F**.

Instruction Editor (LL984)

Instruction Editor

Introduction

For every Control Expert FFB you can use the Control Expert animation tables during runtime. (Select a function block and use **Initialize Animation Table** from the context menu. Please refer to *Animation Tables*, page 644.

But since special FFBs of the UnityLL984 Block Library have parameters of array type there will be no detailed information displayed for these parameters.

So for most of the FFBs of the UnityLL984 Block Library the LL984 editor provides another means of animation called **Instruction Editor**.

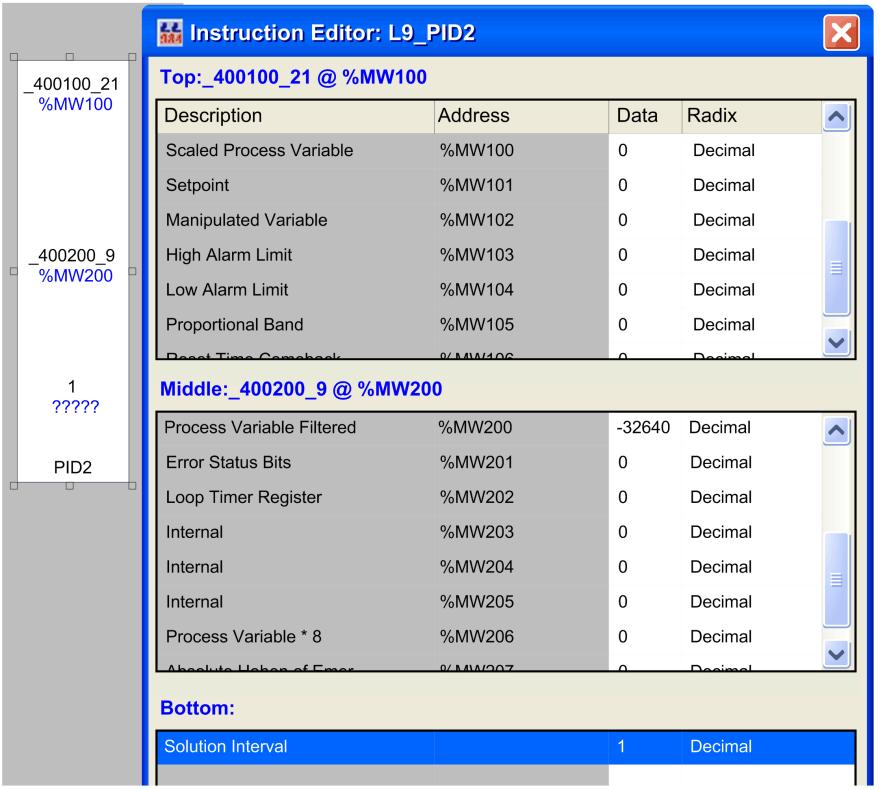
NOTE: For the FFBs of the UnityLL984 Block Library that are not supported by the **Instruction Editor** see table below.

Using the Instruction Editor

To open the **Instruction Editor** for an FFB during runtime, select the FFB and use **Show instruction** from the context menu.

Representation

Representation of the **Instruction Editor**



Items of the **Instruction Editor**

Item	Description
Table Header	displays <Pin name>:<Variable name>@<Address> • pin name of the respective function block node • name of the connected variable
Description	describes the elements of the array
Address	displays the address of the array element in case of located variables In case of unlocated variables this field is empty. In either case the corresponding data column field can be animated/modified.

Item	Description
Data	displays the actual value of the array element The value can be edited by clicking this field.
Radix	allows you to select the displayed data type format Available formats: • Binary • Decimal • Floating Point • HexaDecimal • BCD • ASCII • Signed Integer • 32 Bit Integer

Changing the Data Type

If for example an element of the table is defined as of type floating point or 32 bit integer, the next element is shown as disabled.

For some elements you can change the predefined data type.

For example showing a decimal also as binary.

Changing from floating point to decimal will enable the next decimal row.

Changing back to floating point will disable the next decimal row.

Bit Information

For certain memory words, bit information is also available.

In this case positioning the mouse over the **Data** column of such a memory word will show information for the bits, which may be specially useful when **Binary** is selected as data type format (**Radix**).

FFBs Not Supported by Instruction Editor

The following FFBs are not supported by the **Instruction Editor**.

Family	Not Supported FFBs
CONTL_COMM	L9_ID L9_IE
MISC	L9_LAB L9_RET

Modifying values:

Object of this part

This section describes online modification of values.

Modifying values:

Introduction

You have the several options for changing values of variables online:

- In the shortcut menu for the variables
  - With the **Change Value**
  - With the command **Set Value**
  - With the command **Force Value**
- In the menu **Services**
  - With the command **Change Value**

This applies to all elementary variables and elementary components of derived variables (DDTs and Arrays) with the following applications:

- In LD (also see the Edge recognition (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) chapter in the user guide).
  - Contact
  - Coil
  - Operation block
  - Comparison block

- In FBD
  - FFB Pin
- In SFC
  - Transition
- In IL and ST
  - without limitation

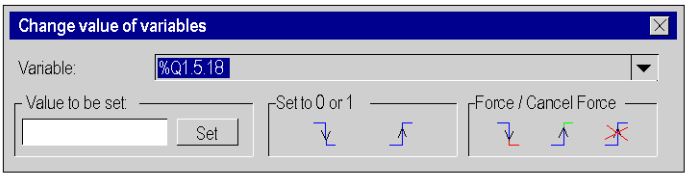
NOTE: The above-named commands are only available in the Animation Mode, page 1397.

Change value of variables

The **Change value of variables** dialog is modeless. This means that the dialog is not closed after modifying a variable. So you can modify several variables in a section.

One dialog is shown per section. Deactivating a section will hide the opened dialog. Closing a section or stopping the animation of a section will close the dialog.

Appearance:



Depending on the selected variables some functions of the dialog may be grayed out.

Elements:

Element	Description
Variable	The selected variable is displayed in this text box. Every variable in a section can be selected either by means of this text box or by clicking on the variable in the section itself.
Value to be set	Enter the value to be set for the valuable in this text box. When making the entry, ensure the correct syntax of the entered literal, e.g. `30s` for a variable of data type `TIME`. Use **Set** to assign the entered value to the selected variable and close the dialog.

Element	Description
Set to 0 or 1	This function is only available for Boolean variables. Use  to set the selected variable to 0 and close the dialog. The same function is available with the command **Set Value > Set to 0** in the context menu of the variable. Use  to set the selected variable to 1 and close the dialog. The same function is available with the command **Set Value > Set to 1** in the context menu of the variable.
Force / Cancel Force	This function is only available for located variables of data type `EBOOL`. Also see the Edge recognition (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) chapter in the user guide. Use  to force the selected variable to 0 and close the dialog. (The forced variable will be marked with a frame around the variable name.) The same function is available with the command **Force Value > Set to 0** in the context menu of the variable. Use  to force the selected variable to 1 and close the dialog. (The forced variable will be marked with a frame around the variable name.) The same function is available with the command **Set Value > Force to 1** in the context menu of the variable. Use  to cancel the Force command for the selected variable and close the dialog. The same function is available with the command **Set Value > Cancel Force** in the context menu of the variable.

Machine-Process Simulator

What's in This Chapter

Introduction 1508

Subject of this Chapter

The SIMAC 100 simulator allows you to simulate a machine or a process piloted by the Premium PLCs.

Introduction

Overview

The SIM function allows you to create a virtual machine which simulates the behaviour of all electrical, mechanical, hydraulic components of your process, in the aim to debug the Control Expert application before starting up on site.

Installation

The SIMAC 100 software must be installed separately of Control Expert software. The software can be downloaded from the PROSYST web site (www.prosyst.fr).

Please, refer to the PROSYST vendor for further installation information.

Diagnostics

What's in This Part

Introduction to diagnostics tools 1511
Diagnostics screens 1515
Diagnostics Viewer 1522
System diagnostics 1543
Project diagnostics 1548

Subject of this Part

This part describes the various diagnostics tools available in Control Expert for the programming, maintenance, and use of PLCs.

Introduction to diagnostics tools

What's in This Chapter

Diagnostics with EcoStruxure Control Expert 1511

Subject of this Chapter

This chapter provides an introduction to the various diagnostics solutions provided by Control Expert software and available on Modicon M340, Premium, Atrium and Quantum PLCs.

Diagnostics with EcoStruxure Control Expert

At a Glance

EcoStruxure Control Expert diagnostics consist of a set of tools and functionalities, which allow you to intervene at all stages in the development and use of an automation application.

The different resources available are as follows:

- Language Objects (IODDTs (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual))
- Diagnostics screens
- The **Diagnostics viewer**
- System diagnostics
- Project diagnostics (diagnostic EFBs and DFBs)

Using the Resources

The following table shows the resources available.

Resource	Use
Language objects (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual)	Language objects are diagnostics tools which can be used in a project to ascertain the status of a channel or module (for example the error bit for channel 1 of the module located in slot 2 of rack 3: %I3.2.1.ERR). These objects are described at the end of the documentation for each module. In general they are used in the form of IODDT (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual) variables.
Diagnostics screens, page 1515	The diagnostics screens can be accessed from the configuration editor for the controller and the application-specific modules. To access the desired diagnostics screen, double-click the component and then, select the **Fault** tab. The detected error information messages are generally explicit (for example: Range over/undershoot on an analog channel).
Diagnostics viewer, page 1522 for EcoStruxure Control Expert	**Diagnostics viewer** is a tool which can be used to view the alarms generated by the system diagnostics and the diagnostic DFBs. There is a **Diagnostics viewer** built into EcoStruxure Control Expert but **Diagnostics viewer** is also available for Magelis HMI terminals or via Factory Cast applications.
System diagnostics, page 1543	System diagnostics is a service which combines the use of system bits and words with**Diagnostics viewer**. When a project has been **built with the System Diagnostics option**, page 551, the system bits and words automatically generate diagnostics messages in **Diagnostics viewer**, page 1522.
Project diagnostics, page 1548 (diagnostics EFBs and DFBs and SFC diagnostics)	Project diagnostics consist of specific EFBs and DFBs which are embedded in the code of an automation project to provide the user or maintenance personnel with clear information on the operating conditions of the monitored process. These program elements will trigger alarms which can be viewed in **Diagnostics viewer**. SFC diagnostics go hand in hand with the system diagnostics, which are an intrinsic part of SFC and enable step activity times to be monitored.

Advantages

The main advantages of these services are as follows:

- Diagnostics are an integral part of the project. It can therefore be thought out at the development stage in order to be able to best meet the requirements of the user and maintenance personnel.
- Errors are logged and dated at source (in the controller), meaning that the information provided is an exact representation of the state of the process.
- You can connect several Viewers (EcoStruxure Control Expert, Magelis, Factory Cast) which will describe the exact state of the process to the user. Each Viewer works independently and any action taken with one (an acknowledgment for example) is automatically visible on the others.

Diagnostics screens

What's in This Chapter

Module and processor diagnostics screens..... 1515

Subject of this Chapter

This chapter describes the diagnostics screens enabling the user to perform troubleshooting for his automation system.

The screens dealt with are those for:

- Premium and Quantum processor modules,
- an application-specific counting module.

NOTE: by way of example, only diagnostics for the application-specific counting module are covered here. For the other modules please refer to the manuals for the corresponding specific applications.

Module and processor diagnostics screens

Subject of this Section

This section introduces the user to the counting module diagnostics screens.

NOTE: By way of example, only diagnostics for application-specific counting modules are covered here. The diagnostics screens for the other application-specific modules as well as those for processors are similar.

Viewing Diagnostics

At a Glance

The diagnostic screens on the module or channel are only accessible in connected mode. When an **un-masked** error is detected, this is reported:

- in the configuration screen of the rack, with the presence of a red square in the position of the module,
- in all screens at module level (**Description** and **Fault** tabs),
  - in the module field with the **I/O** LED.
- in all channel level screens (**Configuration**, **Adjustment**, **Debug** and **Fault** tabs),
  - in the module field with the **I/O** LED.
  - in the channel field with the channel error LED.
- in the error screen that is accessed by the **Fault** where the error diagnostics are described.

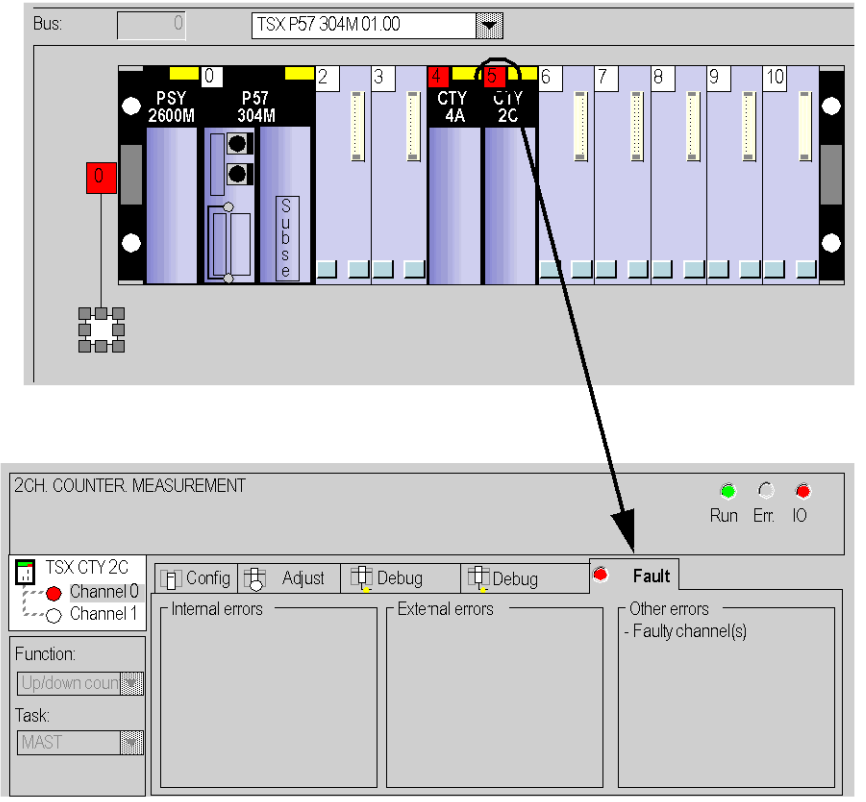
The detected error is also reported:

- on the module, through the centralized viewing,
- with the dedicated language objects: **CH_ERROR** (%Ir.m.c.ERR) and module error **MOD_ERROR** (%Ir.m.MOD.ERR), **%MWr.m.MOD.2**, etc., and the status words (see Premium and Atrium using EcoStruxure™ Control Expert, Counter Modules, User Manual).

NOTE: even if the detected error is masked (TSX CTY 2C), it is reported by the flashing of the LED **I/O** and in the error screen.

Illustration

The diagram below shows the error reporting structure.



Lists of Diagnostics

At a Glance

The messages displayed on the diagnostics screens are used to assist in debugging the project. These messages are concise and sometimes ambiguous (as different detected errors may have the same consequences).

These diagnostics are on two levels: module and channel, the latter being the most explicit. The lists below show the message headings with suggestions for determining the cause of the error.

Module Error Messages

The table below provides a list of the module detected error messages.

Error indicated	Possible interpretation and/or action
Module failure	The module has an error detected. Verify the module mounting. Change the module.
Faulty channel(s)	One or more channels have an error detected. Refer to channel diagnostics.
Self-test	The module is running a self-test. Wait until the self-test is complete. **NOTE:** For Modicon M340, if the Xbus line termination are missing, the module will stay in this state.
Different hardware and software configurations	There is a lack of compatibility between the module configured and the module in the rack. Make the configuration and the software configuration compatible.
Module is missing or off	Install the module. Fasten the mounting screws.

Channel Error Messages

The table below gives the list of detected error messages at channel level.

Error indicated. Other consequences.	Possible interpretation and/or action.
External fault or counting input fault: • encoder or proximity sensor supply error detected, • line break or short circuit of at least one encoder differential signal (1A, 1B or 1Z), • SSI series frame error detected, • specific error detected on absolute encoder. Outputs are set to 0 in automatic mode. **Invalid measurement** message.	Verify the sensor connections. Verify the sensor power supply. Verify the sensor operation. Delete the detected error and acknowledge if the detected error storing is configured (CTY 2C). Counting pulses or incremental encoder: preset or reset to acknowledge the **Invalid measurement** message.
Counting application fault: • measurement overrun • overspeed Outputs are set to 0 in automatic mode. **Invalid measurement** message.	Diagnose the detected error more precisely (external causes). Verify the application again, if necessary. Delete the detected error and acknowledge if the detected error storing is configured (CTY 2C). Counting pulses or incremental encoder: preset or reset to acknowledge the **Invalid measurement** message.
Auxiliary input/output fault: • power supply • short circuit of at least one output Outputs are set to 0 in automatic mode.	Verify the output connections. Verify the input/output power supply (24V). Diagnose the detected error more precisely (external causes). Delete the detected error and acknowledge if the detected error storing is configured (CTY 2C).
Internal fault or channel self-test: • detected error condition, • module missing or off, • module running self-test.	Module detected error at channel level. Refer to module level diagnostics.
Different hardware and software configurations	Module detected error at channel level. Refer to module level diagnostics.
Invalid software configuration: • incorrect constant • bit combination not associated with any configuration	Verify and modify the configuration constants.

Error indicated. Other consequences.	**Possible interpretation and/or action.**
Communication fault	Verify the connections between the racks.
Application fault: Impossible to configure or adjust	Diagnose the detected error more precisely.

Diagnostics Viewer

What's in This Chapter

Reminder of the Diagnostics Associated with the Viewer 1522
Operation of the Diagnostics in EcoStruxure Control Expert 1524
Description of the **Diagnostics viewer** 1525
How to find causes 1528
Configuring the display 1530
Managing Error Messages 1532
Diagnostics Viewer Commands 1534
DiagViewer Properties 1536
Archiving Error Messages 1539

Subject of this Chapter

The following chapter describes the Diagnostics Viewer, used to display error messages.

Reminder of the Diagnostics Associated with the Viewer

Diagnostics in EcoStruxure Control Expert

To verify the correct operation of your process, EcoStruxure Control Expert offers you two types of diagnostics:

- system diagnostics,
- project diagnostics.

The **Diagnostics viewer** is used to display error messages.

System Diagnostics

System diagnostics are performed automatically. When the controller detects a system error (for example, a watchdog being exceeded, an input/output error, division by zero, etc.), information is sent to the **Diagnostics viewer**. The **Diagnostics viewer** displays a system

error message if in the project settings, you selected the **System diagnostics** , page 551 check box.

Project Diagnostics

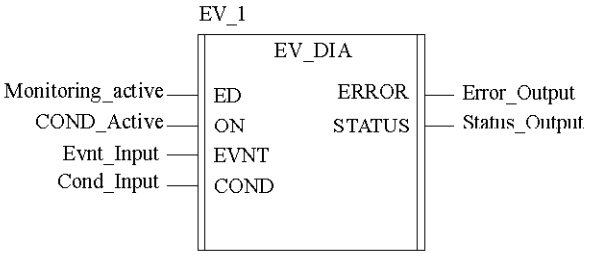
Project diagnostics uses diagnostics EFBs and DFBs. In addition, SFC program diagnostics form an integral part of the system and allow you to test the activation times of the steps.

When the **Project diagnostics** , page 551 check box is selected, information is sent to the viewer when an error is detected, which displays an error message.

Diagnostics EFBs and DFBs

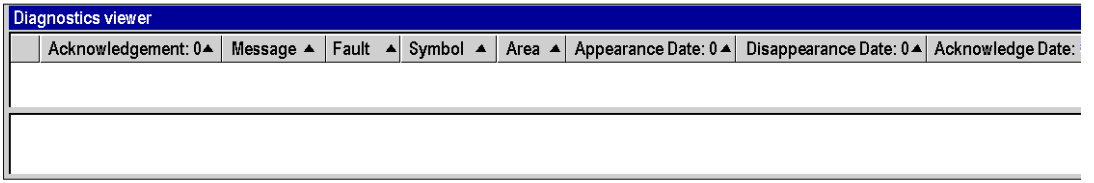
EFBs and DFBs are function blocks that are programmed to carry out diagnostics in your project, page 1548.

The graphic representation of a DFB is as follows:



Viewer

The **Diagnostics viewer** is a tool that is used to display and acknowledge error messages linked to diagnostics.



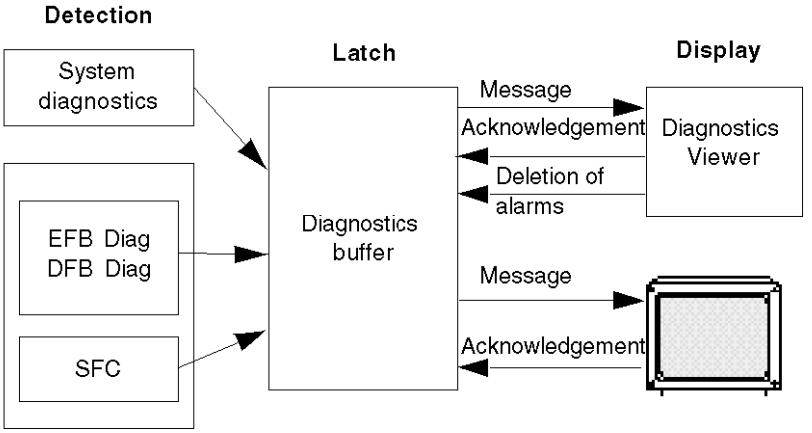
Operation of the Diagnostics in EcoStruxure Control Expert

General

The EcoStruxure Control Expert diagnostics function monitors the elements of the process to detect errors. When an error is detected, it sends an error message to the display systems which display this detected error.

Illustration

The following illustration shows the operation of the EcoStruxure Control Expert diagnostics:



Operation of the Diagnostics

The table below describes the various phases of operation of the EcoStruxure Control Expert diagnostics:

Phase	Description
1	The diagnostics function blocks built into the project program and the diagnostics built into the system detect process errors.
2	The detected errors are stored in the form of date/time stamped messages in the diagnostics buffer memory.
3	The error messages are then displayed via the following display systems: Diagnostics Viewer or Human Machine Interface (HMI), used to: • display messages, • acknowledge messages that need an acknowledgement, • initialize animation tables and cross-references, • launch associated program editors, • delete diagnostics buffer alarms.

Description of the Diagnostics viewer

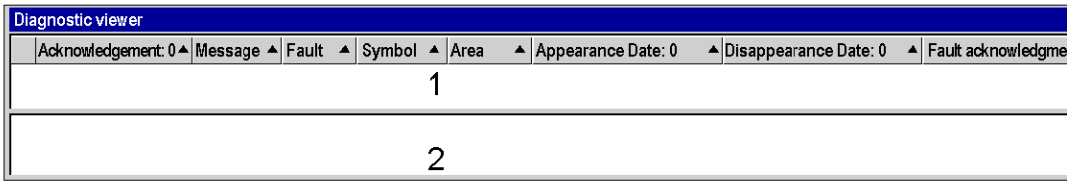
At a Glance

The **Diagnostics viewer** is presented in the form of a display window comprising 2 display fields:

- a field that displays the diagnostics error messages,
- a field that provides additional information on the error message selected from the alarms list.

Diagnostics viewer

The following screen presents the **Diagnostics viewer** and its 2 display fields:



1 List of error messages

2 Additional information on the message selected

Description of Error Messages

For each of the alarms, the error messages list displays the following information:

Field	Description
Acknowledgement	The icon and the text indicate the message status: not acknowledged, acknowledged, deleted or no acknowledgement.
Message	Error message. This message contains the 40 first characters of the text entered in the comment of the function block instance that generated the error.
Fault	Type of diagnostics FB or system bit that detected the error.
Symbol	Symbol associated with the detected error.
Field	Controller zone.
Appearance Date	Date and time of appearance of the detected error.
Disappearance Date	Date and time of disappearance of the detected error.
Acknowledge Date	Date and time of acknowledgement of the detected error.

Definition of Icons

The **Diagnostics viewer** uses the following icons to indicate the message status:

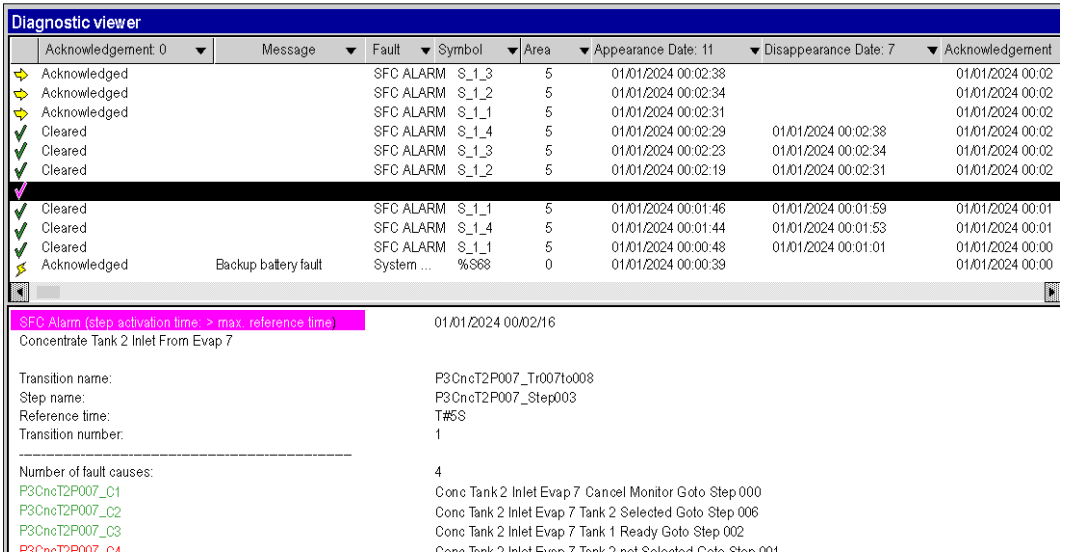
Icon	Description
	The detected error has disappeared and has been acknowledged (if it needed an acknowledgement).
	The detected error needs an acknowledgement.
	The detected error has not disappeared.
	The system detected error has disappeared and has been acknowledged (if it needed an acknowledgement).
	The system detected error has not disappeared.

Description of the Additional Information Field

The additional information field provides the following information for the message selected in the list:

- the alarm type with
  - a comment (defined by the user for the diagnostics FB)
  - the date of appearance,
- specific variables depending on the type of alarm,
- a list of error variables including any associated comments (if these exist),
- interpretation of the status words as a function of the STO file corresponding to this alarm.

The following screen gives an example of additional information on an error message:



If you want to refresh the values of the status words, reselect the message from the list or press the **F5** function key.

NOTE: If you wish, you can find the element that caused the error displayed in the viewer. This is called finding the cause., page 1528

How to find causes

At a Glance

Cause finding involves searching for an element which has triggered an alarm on a diagnostics EFB or DFB input.

Each input with the **Diag** property can be used to search for causes.

How to find causes

The following table describes the procedure for finding causes.

Step	Action
1	Select the alarm in question in the Viewer. **Result**: The alarm line is shown highlighted.
2	Right-click on the input for which you want to search for the cause. **Result**: The contextual menu allows you to initialize an animation table and to search for the DFB input variable so as to ascertain whether it is used in the program. This is called cause finding.
3	Click on the **Initialize Search** to search for cross-references to the selected input. **Result**: The Cross-References screen is displayed.

Lists of diagnostics inputs of diagnostics EFBs or DFBs

The table below specifies the diagnostics inputs (with **Diag** property selected) of diagnostics EFBs or DFBs in the diagnostics library.

Block name	Block type	Name of diagnostics inputs
ALRM_DIA	DFB	COND1 COND0
D_ACT	EFB	UNLOCK REACT
D_DYN	EFB	UNLOCK REACT
D_GRP	EFB	IN
D_LOCK	EFB	UNLOCK
D_PRE	EFB	IN
D_REA	EFB	REACT
EV_DIA	DFB	EVENT COND
MV_DIA	DFB	EVENT COND
NEPO_DIA	DFB	REQ_I REQ_O SENSOR_I

Block name	Block type	Name of diagnostics inputs
		SENSOR_O SECU_I SECU_O SECUPERM
TEPO_DIA	DFB	REQ_I REQ_O SENSOR_I SENSOR_O SECU_I SECU_O SECUPERM
SAFETY_MONITOR	DFB	-
USER_DIAG_ST_MODEL	DFB	COND

Configuring the display

At a Glance

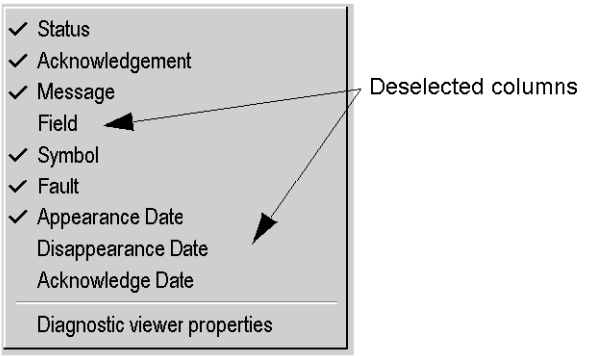
The Diagnostics Viewer allows you to configure the display of the error messages. You can:

- select the columns to be displayed,
- resize the columns,
- choose the color and flashing of the messages.

Selecting the columns to be displayed

By default the Diagnostics Viewer displays all of its columns, but allows you to customize the display by removing columns that are of no use.

To remove a Diagnostics Viewer column, you must perform the following operations:

Step	Action
1	Right-click the title of the column selected. **Result**: A contextual menu containing the list of columns appears on the screen.
2	Click the title of the column you want to deselect. **Result**: The contextual menu disappears and the column no longer appears in the Diagnostics Viewer. **Note**: To reselect a column you have previously deselected, you must proceed in the same way. In this case, the contextual menu disappears and the column appears in the Diagnostics Viewer.
3	If you display the contextual menu again (steps 1 and 2), this column will not be preceded by a pointer (deselected column). 

Resizing of columns

To modify the size of a column, carry out the following operations:

Step	Action
1	Position your cursor in the column titles header, in between the column you want to modify and the next column. **Result**: The column resizing cursor appears on the screen.
2	When the cursor appears, press and hold the right button of the mouse and move the mouse to change the column size.
3	Once the correct column size has been reached release the mouse button. **Result**: The column is resized.

Choosing the color and flashing of messages

The **Properties** dialog box of the Diagnostics Viewer, page 1536 is used to change the color and flashing of error messages.

Managing Error Messages

At a Glance

The **Diagnostics viewer** allows you manage the error messages. You can:

- sort the messages list,
- browse the messages list,
- acknowledge a message in the list,
- delete a message from the list,
- activate another tool,
- delete an alarm from the controller memory.

Number of Messages

The number of messages displayed in the list is defined solely by the size of the available memory.

If the memory becomes insufficient:

- the DIAG information flashes in the status bar,
- the messages associated with the detected errors that have disappeared and have been acknowledged (if they needed acknowledging) are deleted.

Sorting Messages

You can sort the messages by each column contained in the list except for the one containing the icons.

To sort the messages, simply click the header of the column containing the data you want to sort. If you click the same header a second time it will be sorted in reverse order.

By default the messages are inserted in the list in chronological order of appearance of the detected errors.

NOTE: Whatever the sort criterion for the list, a new message always appears at the **top of the list**.

Browsing the List

You can browse the list of messages using:

- the **Up**, **Down**, **PageUp**, **PageDown**, **Home** and **End** keys on the keyboard,
- the mouse. Use the scroll bar if the list contains more messages than can be displayed.

Acknowledging a Message

To acknowledge a message that requires acknowledgement, you must select it and activate one of the following commands:

- the corresponding heading in the contextual menu (accessible by right-clicking with the mouse),
- the **F6** function key,
- the corresponding button in the toolbar,
- double-clicking the icon of the alarm selected.

You can acknowledge several messages at once (multiple selection). When you acknowledge a message, an order is sent to the controller and the associated icon is modified in the list.

A message can be acknowledged from another viewer. In this case, the EcoStruxure Control Expert **Diagnostics viewer** is advised and the message is displayed as if acknowledged locally (the associated icon is modified and the message no longer flashes).

Deleting Messages From the List

You cannot delete a message that needs acknowledgement or where the associated detected errors has not disappeared.

However, you can delete the messages associated with the detected errors that have disappeared and have been acknowledged. To do this activate:

- the corresponding heading in the contextual menu,
- the **Del** key.

Deleting an Alarm From the Controller Memory

This function is used to delete from the controller all the messages selected in the list, even if the detected errors is still present and has not been acknowledged.

Use of this function is only possible if your user profile (see EcoStruxure™ Control Expert, Security Editor, Operation Guide) is different from **ReadOnly** or **Operate**.

Activating Another Tool

If at least one message is selected in the list, you can activate the following EcoStruxure Control Expert tools:

- the animation tables for displaying the diagnostics function block data (FB alarms only) and FFB system alarms,
- the cross-references with FB alarms and FFB system alarms only,
- the language editor where the diagnostics function block instance is referenced.

To launch these tools you must activate one of the following commands:

- the corresponding headings in the contextual menu,
- the **Ctrl + T** (animation tables), **Ctrl + U** (cross references) and **Enter** (language editor) function keys,
- the corresponding buttons in the toolbar.

Diagnostics Viewer Commands

At a Glance

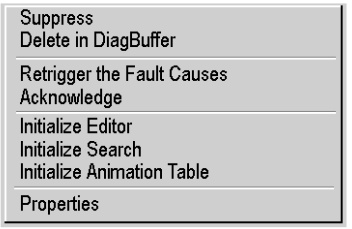
Once you have selected one or more error messages (active errors only), you have a number of options available for accessing the Diagnostics Viewer commands:

- using the contextual menu,
- using the function keys,
- using the toolbar.

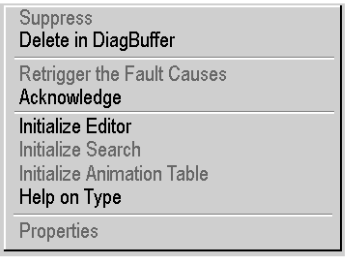
Contextual Menu

The contextual menu can be accessed by right-clicking a message from the list with the mouse:

None FFB error



FFB error



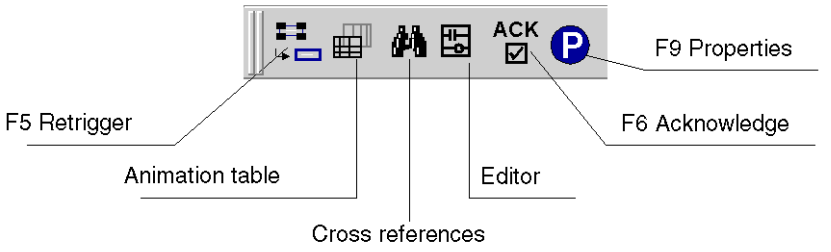
The commands and their function keys from the contextual menu are as follows:

Command	Function key	Description
Suppress	-	deletion of the message(s) selected
Delete in DiagBuffer	-	deletion of alarms in the diagnostics buffer
Retrigger the Fault Causes	F5	reactivation of the diagnostics function block
Acknowledge	F6	acknowledgments of selected message(s)
Initialize Editor	Enter	activation of the language editor
Initialize Search	Ctrl + U	activation of the cross references tool (search for active errors)
Initialize Animation Table	Ctrl + T	activation of the animation table tool (initialize animation table of active errors
Help on Type	-	opening the online help for the concerned FFB
Properties	Alt + Enter	displaying the Diagnostics Viewer properties box

NOTE: The **Initialize Search** and **Initialize Animation Table** commands are also accessible through the contextual menu in the additional message information area.

Toolbar

The icons of the toolbar corresponding to the contextual menu or to the function keys are as follows:



DiagViewer Properties

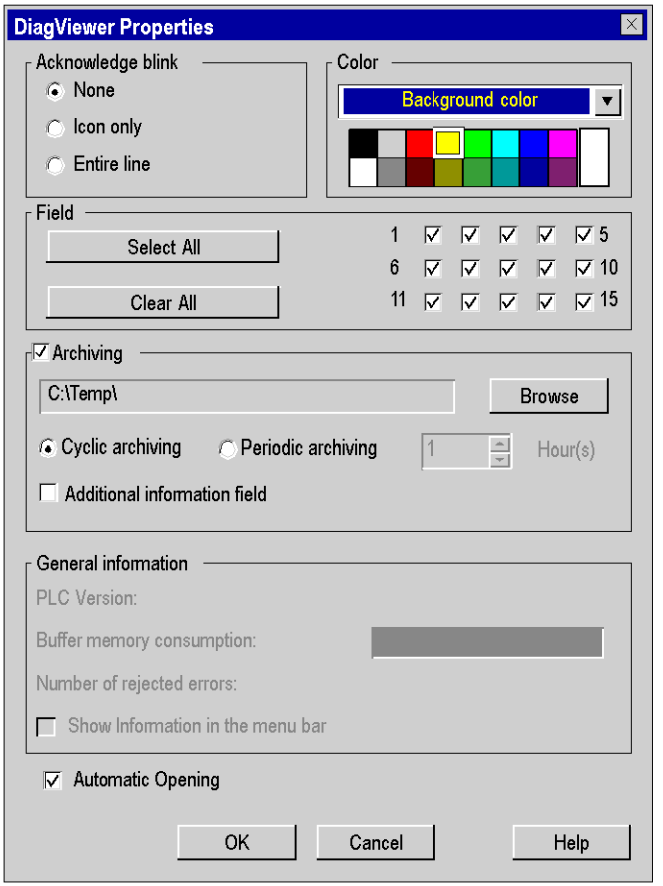
At a Glance

The **DiagViewer Properties** dialog box, accessible via the **DiagViewer Properties** command in the contextual menu, is used to:

- Define the color of the error messages,
- Define the flashing mode of the error messages,
- Choose the fields to be displayed,
- Define the archiving mode,
- Define the mode of opening the **Diagnostics viewer**.

Properties Dialog Box

The screen below presents the **DiagViewer Properties** dialog box:



Description

The following table describes the various fields of the **DiagViewer Properties** dialog box:

Field	Description
Acknowledge blink	Allows you to define the type of flashing when an error message to be acknowledged appears. Once acknowledged messages no longer flash. • **None**: no flashing, • **Icon only**: only the icon flashes, • **Entire line**: the entire line flashes.
Color	Allows you to define the color of the non-acknowledged error messages.
Field	Allows you to choose the fields to be displayed (fields 0 to 15). By default all the fields are displayed. • **Select all**: selects all fields, • **Clear All**: deselects all fields. To select a field, select the corresponding check box.
Archiving	Click this box to generate an archive, page 1539. The entry field and the **Browse** button allow you to define the path of the archive file. • **Cyclic archiving**: a new archive file is generated every 1000 saves. • **Periodic archiving**: a new archive file is generated every x hour. • **Hour** allows you to define the number x (from 1 to 24). • **Additional information field**: allows you to store additional data in the file.
General information	Displaying various information: • Controller version. • Diagnostics buffer memory usage. • Number of alarms rejected if the diagnostics buffer is full. • **Show Information in the menu bar** is used to display this information in the title bar of the **Diagnostics viewer** window.
Automatic Opening	When this check box is selected, the **Diagnostics viewer** is automatically displayed when en error is detected.

Archiving Error Messages

At a Glance

Archiving is used to store in a file all error messages as soon as they appear, disappear or are acknowledged. As soon as a message is read in the diagnostics buffer it is saved in the archive file.

If an error is detected and then disappears, it is represented by a single message in the **Diagnostics viewer** (a single line), but by 2 lines in the archive file.

The **Properties** dialog box allows you to activate archiving of error messages. It also allows you to define the type of archiving and the location of the archive file.

Type of Archiving

You can choose between 2 types of archiving:

- **cyclic archiving**: The history file created is named *ProjectName.his*, were *ProjectName* is the name you entered in the **Project Properties** dialog box is.

  in order to avoid the size of the archive file becoming too large, a new *ProjectName.his* file is created every 1000 saves. The old file then changes name and becomes *ProjectName.bak*.

  Note: If a *ProjectName.bak* file already exists, it will be overwritten by the new file of the same name. To avoid a loss of data, you must therefore manage your archive files.
- **periodic archiving**: the user defines the frequency with which a new archive file is created (between 1 and 24 hours). The file is created under the name *ProjectDraft_ date_time* (for example, for a file created on 31 July 2001 at 12 h 26 min 55 s: *ProjectDraft_073101_122655*. If the chosen frequency is 1 h, the next file will be created under the name *ProjectDraft_073101_132655*, etc.). Creating a new file does not overwrite the previous file.

Archive File Format

The archive file is a delimited ASCII file (each item of information is separated by the **;** character). You can therefore easily import this file into a text editor or a spreadsheet.

Each line of the file takes the following form:

```
SystemAlarm ; 256 ; 1 ; 51 ; 0 ; 28/04/03  23:50:13 ; 28/04/03
23:51:15; 28/04/03 23:57:16 ; 0 ; 2; 10003 ;148 ; Arithmetic error
; %S18 (MAST);
```

where:

- **SystemAlarm**: identifies the detected error.

  The types of detected error can be:

  - FBAlarm: function block detected error
  - SFCAlarm: sequential function chart detected error
  - SystemAlarm: system detected error
  - UndefAlarm: undefined detected error
  - AlarmERROR: errors
- **256**: detected error identifier,
- **1**: name of this detected error,
- **51**: state of the diagnostics bits,
- **0**: number of the declared area,
- **28/04/03 23:50:13**: date and time of detection of the error,
- **28/04/03 23:51:15**: date and time of acknowledgement of the detected error,
- **28/04/03 23:57:16**: date and time of disappearance of the detected error,
- **0**: available number of reasons for the detected error,
- **2**: status length of the detected error,
- **10003**: detected error type,
- **148**: detected error classification,
- Arithmetic error; %S18 (MAST): additional data if you have selected the **Additional information field** check box.

The following table describes the additional information to a message according to the detected error type.

Error type	Information
FBAlarm	Comment Name of the function block instance Name of the function block type Name of the input or output of the erroneous block
SFCAlarm	Comment Name of the erroneous step Name of the erroneous transition Format of the reference time Additional information
SystemAlarm	Detected error name Additional information according to the detected error: • watchdog, • arithmetic, • task, • SFC, • ...

System diagnostics

What's in This Chapter

System Diagnostic 1543

Subject of this Chapter

This chapter describes how to implement system diagnostics and its functionalities.

System Diagnostic

At a Glance

The system diagnostic is performed automatically. When the controller detects a system error (for example, a watchdog being exceeded, an input/output error, division by zero, etc.), information is sent to the **Diagnostic viewer**. The **Diagnostic viewer** will display a system error message if you have selected the System diagnostics, page 551 check box.

NOTE: As with the project diagnostic, the information viewed on the Viewer comes from the controller diag buffer, page 1524, consequently the events are dated at the source and give the precise status of the process monitored.

Implementation

The following table describes the procedure to follow to use the system diagnostic on the Premium, Atrium, and Quantum controllers in EcoStruxure Control Expert.

Step	Action
1	Select **Tools > Project Settings** Result: the project settings configuration window is displayed.
2	In the **Diagnostics** area on the **Build** tab, select the **System diagnostics**, page 551 check box.
3	Confirm with **OK**.
4	Build your project.
5	Transfer your project to the controller.
6	Open the **Diagnostic viewer** by clicking **Tools > Diagnostic viewer**. **Result**: the system diagnostic is operational and each system alarm appears in the Viewer.

List of System Alarms Monitored

The following table summarizes the system information monitored automatically by the system diagnostic service.

System object	Succinct description of the alarm
%S10	Input/output error
%S11	Watchdog overflow
%S15	Character string error
%S18	Overflow or arithmetic error
%S19	Task period overrun
%S20	Index overflow
%S39	Saturation in event processing
%S51	Time loss in real time clock
%S65	Extract card command
%S66	Backup application to the memory card
%S67	State of the PCMCIA application memory card battery
%S68	State of controller battery
%S76	Diagnostic buffer configured
%S77	Diagnostic buffer full
%S96	Previously backup program
%S118	General Fipio I/O error
%S119	General in-rack I/O error
%SW0	Master task scanning period
%SW1	Fast task scanning period
%SW2	Period of auxiliary task scanning 0.
%SW3	Period of auxiliary task scanning 1.
%SW4	Period of auxiliary task scanning 2.
%SW4	Period of auxiliary task scanning 3.
%SW11	Watchdog duration
%SW17	Error status for floating operation
%SW76	Diagnostic function: save

System object	Succinct description of the alarm
%SW77	Diagnostic function: de-registration
%SW78	Diagnostic function: number of errors
%SW96	Save/restore %MW in flash memory.
%SW97	Storage card error code
%SW125	Type of blocking error
%SW146	Fipio bus arbiter function display
%SW153	List of Fipio channel manager detected errors
%SW154	List of Fipio channel manager detected errors

Project diagnostics

What's in This Chapter

How to Create a User Diagnostic DFB 1548
Project diagnostic 1550

Subject of this Chapter

This chapter describes how to implement project diagnostics and its functionalities.

How to Create a User Diagnostic DFB

At a Glance

You can create a diagnostic DFB in one of two ways.

- You can create a standard DFB and then define its values and program it so that it is a diagnostic DFB.
- You can use the USER_DIAG_ST_MODEL (see EcoStruxure™ Control Expert, Diagnostics, Block Library) model that we will provide you with, or use a model you already have.

These two procedures are described below.

NOTE: to understand how user diagnostic DFBs work, refer to the USER_DIAG_ST_MODEL (see EcoStruxure™ Control Expert, Diagnostics, Block Library) model described in the diagnostic library.

Creating a Diagnostic DFB without a Model

The following table describes the procedure to follow to create a diagnostic DFB without a predefined model.

Step	Action
1	In the structure view of the Project Browser, right-click on the directory **Derived FB Types**. **Result**: A shortcut menu is opened.
2	Activate the **Open** command.

Step	Action
	Result: The Data Editor window is opened. It is positioned directly under the **DFB Types** tab and shows the list of the project's DFBs.
3	Select the first empty cell in the **Name** column (indicated by an arrow) and enter the name of the new DFB type, then confirm using **Enter**. **Result**: The empty structure of the DFB type is now created. This new DFB is added to the list of existing DFBs. It also appears in the **Derived FB Types** directory tree.
4	In the Data Editor, right-click on the type of DFB you have just created. **Result**: the contextual menu appears.
5	Choose the **Data Properties** command. **Result**: the Data Properties window appears.
6	Click on the **Diag** button. **Result**: the checkbox is checked in red. This type of DFB is now a diagnostic DFB. If, in the Project Settings, page 551, the application diagnostic has been requested, the project's generation will incorporate the project diagnostic service in the PLC.
7	Create the inputs, outputs, inputs/outputs and the internal variables of the DFB. **Note**: in the input's properties, remember to check the **Diag** checkbox for inputs that will be monitored by the DFB.
8	Write your DFB's code using the REGDFB (see EcoStruxure™ Control Expert, Diagnostics, Block Library) function to save and date the alarm, and the DEREG (see EcoStruxure™ Control Expert, Diagnostics, Block Library) function to de-register the alarm. **Note**: refer to the description of the code of the (see EcoStruxure™ Control Expert, Diagnostics, Block Library)USER_DIAG_ST_MODEL user diagnostic DFB model if you are not familiar with this type of DFB.
9	Run a project analysis to validate your new type of DFB. **Result**: your new type of DFB is created. You can now create and insert instances of this type of DFB in your program. When your PLC is in RUN mode, the Viewer, page 1522 will display the alarms according to the operation you have programmed.

Creating a Diagnostic DFB with the USER_DIAG_ST_MODEL Model

The following table describes the procedure to follow to create a diagnostic DFB using the USER_DIAG_ST_MODEL predefined model.

Step	Action
1	In the structure view of the Project Browser, right-click on the directory **Derived FB Types**. **Result**: A shortcut menu is opened.
2	Choose the **Get from Library** command. **Result**: the **Library Browser** window appears.
3	Select the USER_DIAG_ST_MODEL diagnostic DFB model in the diagnostic library, then click on the **>>** button to transfer it to your project.
4	Confirm the transfer with **OK**. **Result**: the model appears in the Project Browser.
5	Replace the USER_DIAG_ST_MODEL name with the name of the type of user DFB you want to create.
6	Modify and create the inputs, outputs, inputs/outputs and the internal variables of the DFB. **Note**: in the input's properties, remember to check the **Diag** checkbox for inputs that will be monitored by the DFB.
7	Write your DFB's code using the model. **Note**: refer to the description of the code of the user diagnostic DFB model (see EcoStruxure™ Control Expert, Diagnostics, Block Library) if you are not familiar with this type of DFB.
8	Run a project analysis to validate your new type of DFB. **Result**: your new type of DFB is created. You can now create and insert instances of this type of DFB in your program. When your PLC is in RUN mode, the Viewer, page 1522 will display the alarms according to the operation you have programmed. **Note**: the message displayed by the Viewer is the comment associated with the instance of the DFB. You must therefore remember to fill in this comment for each instance created.

Project diagnostic

At a Glance

The project diagnostic uses the diagnostic EFBs and DFBs and the diagnostic integrated in the SFC to generate alarms on the Diagnostic Viewer.

Each diagnostic EFB and each diagnostic DFB has its own specific operation described in the diagnostic library (see EcoStruxure™ Control Expert, Diagnostics, Block Library). If you cannot find the appropriate EFB or DFB among these elements, you can create your own diagnostic DFB (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).

NOTE: It is strongly recommended to only program a diagnostic DFB instance once within the application.

NOTE: As with the system diagnostic, the information viewed on the Viewer comes from the PLC's diag buffer, page 1524, consequently the events are dated at the source and give the precise status of the process monitored.

Implementation of diagnostic EFBs or DFBs

The following table describes the procedure to follow to use the project with a diagnostic EFB or DFB on the Premium, Atrium and Quantum PLCs in Control Expert.

Step	Action
1	Select **Tools** > **Project Settings** Result: the project settings configuration window is displayed.
2	In the **Diagnostics** area on the **Build** tab, check the **Application diagnostics**, page 551 checkbox.
3	Choose the language of the messages in the Viewer.
4	Choose the **Application level**, page 551 (alarm cause search level).
5	Confirm with **OK**.
6	Integrate the diagnostic EFBs or DFBs (see EcoStruxure™ Control Expert, Diagnostics, Block Library) in your application. **Note**: The messages displayed in the Viewer will be the comments you will have associated with the instances of your diagnostic EFBs or DFBs.
7	Build your project.
8	Transfer your project to the PLC.
9	Open the Diagnostic Viewer by selecting: **Tools** > **Diagnostic Viewer**. **Result**: the system diagnostic is operational and each alarm generated by your EFBs or DFBs now appears in the Viewer.

Implementation of the SFC diagnostic

The following table describes the procedure to follow to use the SFC diagnostic on the Premium, Atrium and Quantum PLCs in Control Expert.

Step	Action
1	Select **Tools > Project Settings** Result: the project settings configuration window is displayed.
2	In the **Diagnostics** area on the **Build** tab, check the **Application diagnostics**, page 551 checkbox.
3	Confirm with **OK**.
4	Build your project.
5	Transfer your project to the PLC.
6	Open the Diagnostic Viewer by selecting: **Tools > Diagnostic Viewer**. **Result**: the SFC diagnostic is operational and each alarm linked to the SFC now appears in the Viewer.

Error message display

The number of messages it is possible to display is limited only by the size of the memory buffer.When there is not enough memory, a message warns the user and any messages of errors that have disappeared or have been acknowledged (if necessary) are then deleted.

It is possible to modify the color of the messages and the blinking associated with an acknowledged message.

In the viewer, it is possible to show only those messages which come from one or more specific zones.

The list of messages can be sorted according to each field. To do this, simply click on the column header containing the data on the basis of which the sort is to be carried out.

A second click carries out the sort in opposite order.

By default, the error messages are inserted into the list in the chronological order in which they appear.

Operator screen

What's in This Part

General information 1555
First steps 1558
Editing runtime screens 1568
Runtime screens in online mode 1635
Runtime screens objects library 1646

Subject of this Part

This part describes the operator screens built into Control Expert:

General information

What's in This Chapter

Introduction 1555
Creation of Runtime Screens 1556

Object of this section

This chapter introduces Control Expert built-in operator screens: description, creation and operator screens

Introduction

Introduction

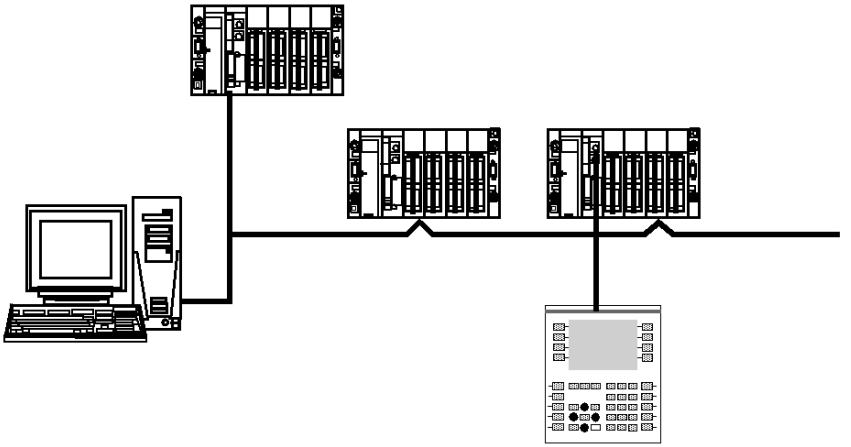
The operator screens aid the operating of an automated process.

They use in the Control Expert software:

- The **project browser** for browsing through the screens and launching different tools (the graphics editor, variables editor, messages editor, ...)
- The **graphics editor** for creating or changing screens. In connected mode, it also allows the viewing of animated screens and process driving
- The **library of objects** which presents design objects and enables their insertion in the screens. It also allows users to create their own objects and insert them in a library family.

Example of use

Example of automation structure that uses operator screens:



In this structure, we find:

- The **PLC** which contains the process management automation project.
- The **terminal** which contains the automation project with the operator screens. It is connected to the PLCs by the console line or by a network.
- The operator screens view the process and can be ordered by the keyboard of the **PC**, the mouse or even by a command **terminal** connected to the PLCs.

Creation of Runtime Screens

Introduction

For a given project, you can use the graphic editor to create runtime screens.

These screens are produced with texts and graphic objects that you can draw yourself (lines, rectangles, curves, etc.) or retrieve from the graphic objects library. They comprise static (background, title, etc.) and dynamic (animated) parts which reflect the status of the process.

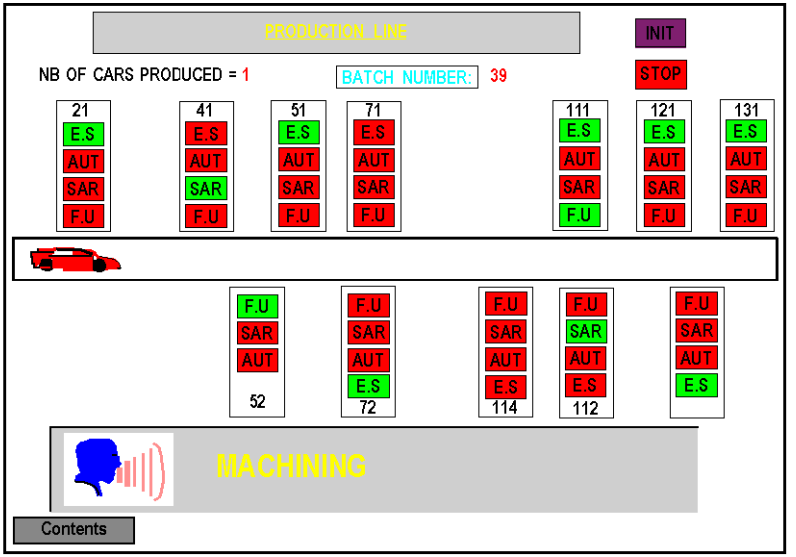
You must assign a variable to them in order to animate the dynamic objects, the variables' value will determine display.

You can also insert driving objects (buttons, input zones, etc.) in your screens to drive the process.

The screens can then be linked together in order to respond to the specific automation requirements.

Example of screen

Example of a screen that shows the state of a process, the quantities produced and allows the safety management of a **production line**.This screen comprises dynamic displays which change to match the process and static displays that remain unchanged.



First steps

What's in This Chapter

Working principles 1558
Menus and tools in the runtime screen editor 1559
Configuring options from the runtime screen editor 1566

Object of this chapter

This chapter gives the elements required for understanding the built-in runtimescreens: notion of runtime screen editor, main menus, etc.

Working principles

Subject of this section

This section recalls the working principles of the runtime screen editor as well as the method of access.

How to access the operator screens

At a Glance

An operator screen is a window just as any other editor (configuration, language, data editor). This window can be opened, moved, and resized.

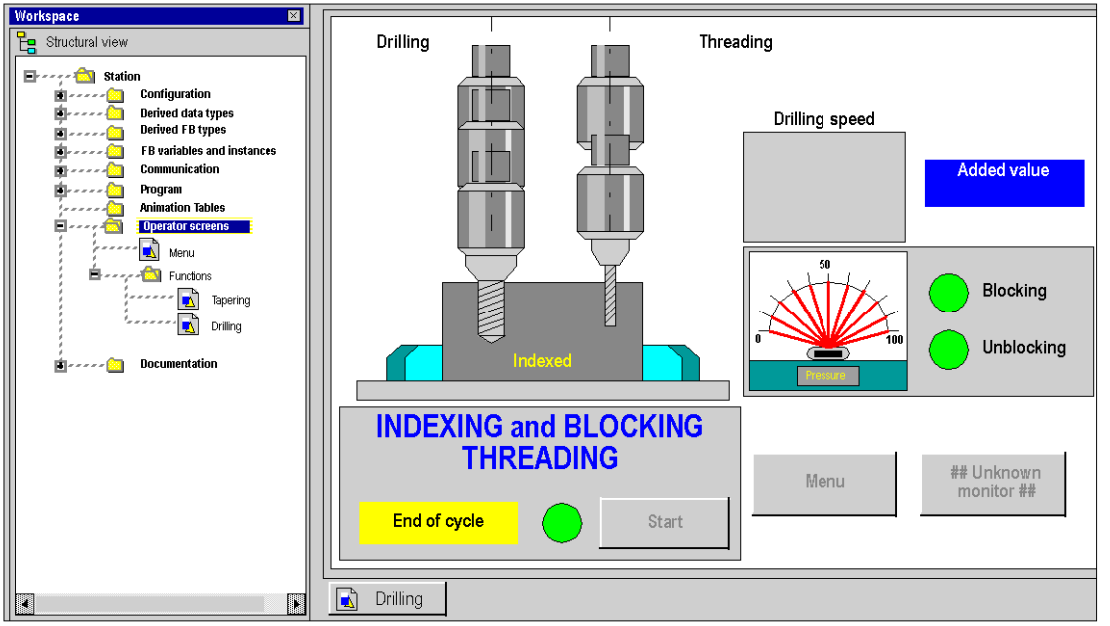
Methodology

The following table lists the operations to perform in order to access operator screens.

Step	Action
1	View the project in structural view (**View > Structural View**).
2	In the project browser, expand the folder **Operator Screens** or the folder of the family for which you want to open a screen.
3	Open the screen or screens to be viewed. Select a screen to open it and activate the command **Open** in the contextual menu.

Example of screen

The following screen shows an example of operator screens.



Menus and tools in the runtime screen editor

Subject of this section

This section introduces the different menus as well as the toolbars offered by the runtime screen editor.

Menus in the operator screen editor

Main menus in the operator screens

The Control Expert menu bar is identical for all tools accessible from the project browser. This is a contextual bar; in other words the commands accessible via the menus depend on the element (window, graphic object, etc.) that is activated when the menu is scrolled down.

The commands of the operator screen editor can be accessed via the menus:

- **Edit** which enable the creation or modification of the operator screens,
- **View** is used to define screen properties (background color, size, etc.),
- **Services** for choosing the editing tool, selecting an animated object, etc.

Edit menu

Commands that can be accessed in the **Edit** menu are.

Field	Description
Undo all	Undoes all changes made in the screen since the last validation.
Enable	Validates all changes made in the screen since the last validation.
Cancel	Cancels the last completed change.
Redo	Re-performs the changes previously cancelled by a command **Cancel**.
Cut	Deletes from the screen the selected item(s) and places it (them) on the clipboard.
Copy	Copies the selected item(s) and places it (them) on the clipboard.
Paste	Pastes the content of the clipboard on the screen.
Delete	Deletes from the screen the selected item(s), without copying them in the clipboard.
Select all	Selects all the items contained in the screen
New	Sub-menu from which you may select: **Line**: Create a line **Rectangle**: Create a rectangle **Ellipse**: Create an ellipse **Curve**: Create a curve **Polygon**: Create a closed polygon **Text**: Create texts **Image**: incorporation of an image

Field	Description
	Screen browsing: Create an inter-screen browser button **Controls**: Creating a driving object (command) of type: • button • checkbox • entry field • counter • cursor • explicit exchanges
Select mode	Select one or several objects.
Align	Sub-menu from which you may select: **Top**: Upper alignment **Bottom**: Lower alignment **Left**: Left alignment **Right**: Right alignment
Flip	Sub-menu from which you may select: **Flip Vertical** **Flip Horizontal**
Group	Grouping of objects.
Ungroup	Ungrouping of objects.
Order	Sub-menu from which you may select: **Foreground**: Placing in the foreground **Background**: Placing in the background
Properties	Is used to define properties of the selected item: color, frame, piloted variable, related text, etc.
Information	Displays information specific to the selected object: type of item, position, size, related variable.

Display menu

Commands accessible from the **View** menu are as follows.

Field	Description
Full screen	Full screen display of the graphics editor window.
Open model used	Enables user to open the model if it is used in the current screen.
Screen properties	Is used to define the properties of the active screen: screen name, background color, family, etc.

Services Menu

The commands accessible from the **Services Menu** areas follows.

Field	Description
Animation	Starts or stops the animation of the operator screen
Enable variable modification	Authorizes the user to modify the variables in the PLC from the command items.
Driving via the PLC	Used to drive the screens with the PLC. The PLCs requirements are taken into account by means of a variable defined in the advanced parameters, page 592. Since the screen is called up by the PLC, editing and modification of the configuration parameters are disabled.
Driving via the operator	Used to drive the screens via the Operator. The requirements of the PLC are disabled, it is the operator who selects the screen that he wants to view.
Previous animated object	Selects the previous animated object.
Next animated object	Selects the next animated object.
Initialize search	Used to create a list of cross references for variables associated with objects selected.
Initialize animation table	Used to create an animation table made up of the variables associated with selected items.

Important

Right clicking can be used to gain direct access to the functions available for the current context. This is the contextual menu.

In online mode, all commands are available if they have not been disabled, page 602.

Multi-instantiation

The operator screen editor is used to open several windows (instances) to open simultaneously. Each window contains a different screen. The keyboard command **Ctrl + Tab** is used to switch between windows. Even if several windows are open, only a single tool bar is displayed and acts on the active window alone.

The full screen display is only used to view one screen.

Edit toolbar

At a Glance

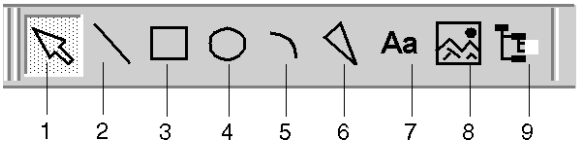
Clicking **Tools > Customize** displays the Edit toolbar described above. Each icon (tool) corresponds to an action concerning one or several objects in the graphical editor.

The figure below presents the tools available via the editor toolbar.



Drawing tools for standard objects

The available tools are as follows:



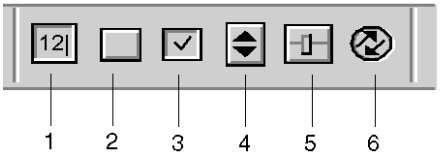
They are used to complete the operations below:

Number	Description
1	Select one or more objects
2	Create a line
3	Create a rectangle
4	Create an ellipse
5	Create a curve

Number	Description
6	Create a closed polygon
7	Create texts
8	Incorporation of an image
9	Create an inter-screen browser button

Design tools for command objects

The available tools are as follows:

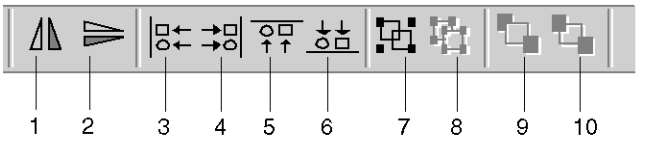


They are used to complete the operations below:

Number	Description
1	Create an input zone
2	Create a button
3	Create a check box
4	Create a counter
5	Create a cursor
6	Create an explicit exchange button

Positioning tools for a group of objects

The available tools are as follows:



They are used to complete the operations below:

Number	Description
1	Flip Vertical
2	Flip Horizontal
3	Left alignment
4	Right alignment
5	Upper alignment
6	Lower alignment
7	Grouping of objects
8	Ungrouping of objects
9	Placing in the foreground
10	Placing in the background

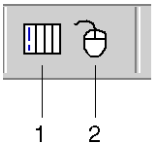
Object properties

The proposed tool is the following. It is used to access the properties of an object.



Screen driving tools

The available tools are as follows:

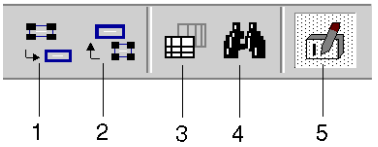


They are used to complete the operations below:

Number	Description
1	Driving via the PLC. This service may also be accessed via a keyboard shortcut **F2**.
2	Driving via the operator. This service may also be accessed via a keyboard shortcut **F3**.

Access tools to the animated objects

The available tools are as follows:



They are used to complete the operations below:

Number	Description
1	Selects the next animated object. This service may also be accessed via a keyboard shortcut **F4**.
2	Selects the previous animated object. This service may also be accessed via a keyboard shortcut **F5**.
3	Initialization of an animation table. This service may also be accessed via a keyboard shortcut **CTRL+T**.
4	Creates a list of cross-references for variables associated with objects selected. This service may also be accessed via a keyboard shortcut **CTRL+U**.
5	Authorizes the user to modify the variables in the PLC from the command items. This service may also be accessed via a keyboard shortcut **F7**.

Configuring options from the runtime screen editor

Subject of this section

This section describes all the configuration options for the runtime screen editor.

General configuration of operator screens

At a Glance

General configuration options and settings for all the operator screens can be used to customize their functions.

The options, page 602 are accessible by clicking **Tools > Options > Operator Screens tab**.

The settings, page 591 are accessible by clicking **Tools > Project Settings > Operator Screens tab**.

These modifications are immediate and are automatically saved.

Editing runtime screens

What's in This Chapter

Editing screens from the browser 1568
Creating an object 1580
Default Properties screen 1589
Modification of an object's attributes 1591
Manipulation of objects in runtime screens 1616
Managing the runtime screens messages editor 1624

Object of this section

This chapter describes the different functions offered to the user, for editing the runtime screens: create, modify, delete a screen or a family of screens, manage built-in runtime screen messages etc.

Editing screens from the browser

Subject of this section

This section describes the different editing actions that users can perform on the screens or family of screens: create, modify, delete, etc.

Editing operator screens

Introduction

You can access operator screens through the project browser in the directory **Operator Screens**. It displays the information linked to each project in a tree directory.

The screens may be directly associated with the directory or grouped in **families** of screens.

The tree directory of operator screens may be expanded and contracted in order to obtain the required level of display.

Screens

The operator screens accept all editing functions:

- Create
- Modify
- Delete
- Move
- Attach/Detach
- Export

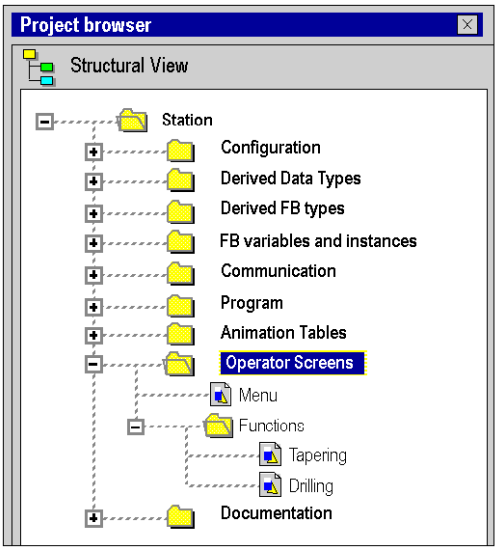
Screen families

The screens are regrouped in families which accept the same editing functions as the screens:

- Create
- Modify
- Delete
- Move
- Attach/Detach
- Export

Introduction to the project browser window

The screen below shows the project browser window with in this example a single screen (Menu) and a screen family (Functions).



Operator screen editor functions

At a Glance

You can access all functions via the **Edit** menu or via the context menu by right clicking in the project browser window.

How to Create a Screen

The Table Below Shows the Procedure for Creating a Screen.

Step	Action
1	Right click the **Operator Screens** directory contained in the project browser tree directory. **Result:** The relevant contextual menu appears
2	Enable the **New screen** command. **Result:** A new screen with a generic name, appears in the tree directory

How to Modify the Name of a Screen

The Table Below Shows the Procedure for Modifying a Screen Name.

Step	Action
1	In the project browser, right click on the screen to be modified. **Result:** The relevant contextual menu appears
2	Enable the **Screen properties** command. **Result:** The screen properties dialog box appears
3	Modify the screen name in the **Name** field and confirm with **OK**. **Result:** The screen appears with a new name, in the tree directory

How to Delete a Screen

The Table Below Shows the Procedure for Deleting a Screen.

Step	Action
1	In the project browser, right click on the screen to be deleted. **Result:** The relevant contextual menu appears
2	Activate the **Delete** command. **Result:** The screen is deleted from the directory tree

How to Move a Screen

The Table Below Shows the Procedure for Moving a Screen in the Tree Directory (For Example, Change Families).

Step	Action
1	Right click on the screen to be moved, from the project browser. **Result**: The relevant contextual menu appears
2	Enable the **Screen properties** command. **Result:** The screen properties dialog box appears
3	Define in the **Location: Family** field, the name of the new family under which you want to place the screen. Click **OK** to confirm. **Result:** The screen is placed in its new position To position it in the root (**Operator Screen** directory) leave the **Location: Family** field empty.

NOTE: The screen can also be moved with the Drag-Drop function

Operation of Screen Movement

When you move a screen from a family to the root, if:

- the family is associated with a functional module, the screen loses this link.

When you move a screen from a family to another **or** from the root to a family, if:

- the destination family screen is associated with a functional module, the screen will be also.
- the destination family screen is not associated with a functional module, the screen will not be associated with any functional module.

How to Attach a Screen to a Functional Module

The Following Table Summarizes the Processes Used to Attach a Screen to a Functional Module.

Step	Action
1	Display the structural view and the functional view.
2	In the project browser of the structural view, select the screen to be associated with a functional module.
3	Hold down the left mouse button to maintain the selection.
4	Drag and Drop the screen in the destination functional module of the functional view. The authorized insertion point of the screen in the destination functional module, is represented by a line. **Result:**The screen is now associated with the chosen functional module

How to Detach a Screen from a Functional Module

The Following Table Summarizes the Processes Used to Detach a Screen from a Functional Module.

Step	Action
1	In the functional view, right click on the screen to be detached from its functional module. **Result:** The relevant contextual menu appears
2	Activate the **Detach** command. **Result:**The screen loses its link with this functional module It can be associated with another functional module.

How to Delete All Screens and All Families

The Table Below Shows the Procedure for Deleting all the Screens and Families of a Project.

Step	Action
1	In the structural view project browser, select the **Operator Screen** directory.
2	Right click on the **Operator Screen** directory. **Result**: The relevant contextual menu appears
3	Activate the **Delete all** command. **Result:**: A dialog box appears
4	Click **OK** to confirm your choice of deleting all screens and all families. **Result:**: All screens and all families are deleted from the project

How to Export Screens

To find out the procedure for exporting operator screens, page 1713.

How to Import Screens

To find out the procedure for importing operator screens, page 1715.

Screen Family Editor Functions

At a Glance

All Functions Can Be Accessed via the **Edit** Menu or by Right Clicking on the Browser Window in the Project Browser.

How to Create a Screen Family

The Following Table Summarizes the Processes Used to Create a Screen Family.

Step	Action
1	Right click the **Operator Screens** folder contained in the project browser tree directory. **Result:** The relevant contextual menu appears
2	Enable the command **New family**. **Result:**: A new family with a generic name appears in the directory tree.

How to Modify the Name of a Screen Family

The Table Below Shows the Procedures for Modifying the Name of a Screen Family.

Step	Action
1	Right click on the family to be modified, from the project browser.
2	Enable the **Family properties** command. **Result:**: The screen family dialog box appears
3	Modify the name of the family in the **Name** field, and click **OK** to confirm. **Result:**: The screen family appears under a new name in the directory tree

How to Delete a Screen Family

The Following Table Summarizes the Processes Used to Delete a Screen Family.

Step	Action
1	Right click on the family to be deleted, from the project browser.
2	Activate the **Delete** command. **Result:**: The screen family is deleted from the directory tree

How to Move a Screen Family

The Following Table Summarizes the Processes Used to Move a Screen Family.

Step	Action
1	Select the screen family to move in the project browser.
2	Hold down the left mouse button to maintain the selection.
3	Move the screen family to the desired position in the project browser.
4	Release the mouse button. **Result:**: The screen family is moved to chosen place

How to Attach a Screen Family to a Functional Module

The Following Table Summarizes the Processes Used to Attach a Screen Family to a Functional Module.

Step	Action
1	Display the structural view and the functional view.
2	In the project browser of the structural view, select the screen family to be associated with a functional module.
3	Hold down the left mouse button to maintain the selection.
4	Drag and Drop the screen family to the destination functional module of the functional view. The authorized insertion point of the screen family in the functional module, is represented by a line. **Result:**: The screen family is now associated with the chosen functional module

How to Detach a Screen Family from a Functional Module

The following Table Summarizes the Processes Used to Detach a Screen Family from a Functional Module.

Step	Action
1	In the functional view, right click on the screen family to be detached from its functional module. **Result:** The relevant contextual menu appears
2	Activate the **Detach** command. **Result:**: The screen family no longer has a link with a functional module It can be associated with another functional module.

How to Export a Screen Family

The export of a screen family takes place in the same way as screen export, page 1713.

Operator screen properties

At a Glance

When a screen is selected, the dialog box that is accessed by clicking **Edit > Screen properties** can be used to:

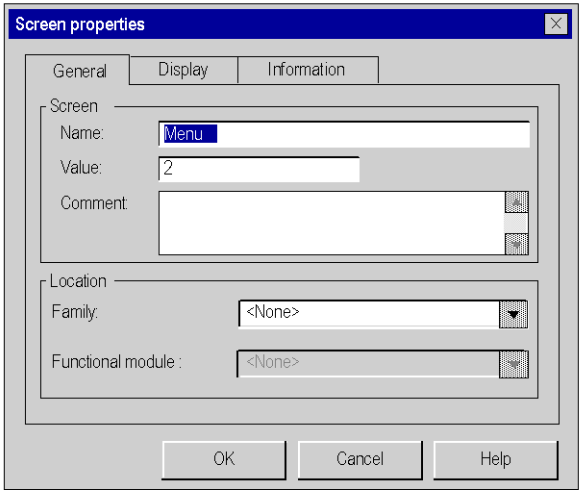
- Modify the screen label
- Modify the value associated with the screen
- Modify the comment associated with a screen
- Modify the screen position
- Modify the screen type
- Modify the color of the screen
- Modify the size of the screen
- Query its creation and modification dates
- Query the number of objects in the screen
- Query the number of variables used in the screen

The dialog box includes three tabs:

- **General**
- **Display**
- **Information**

Screen properties dialog box

The screen properties dialog box is as follows:



Tab: General

The following table describes the different functions offered by the **General** tab of the screen properties dialog box.

Field	Description
Name	Screen name. It can be modified and a maximum of 200 characters can be typed in this field.
Value	Screen identification number You can modify this number which is used when you wish to associate a screen browse, page 1604 button to the screen. This number can be used when the PLC, page 602 controls the screen display in connected mode.
Comment	Screen comment
Family	Name of the family to which the screen belongs. The button to the right of the field is used to access the list of existing families.
Functional Module	Name of the functional module to which the screen is linked. The button to the right of the field is used to access the list of existing the functional modules.

Tab: Display

The following table describes the different functions offered by the **Display** tab of the screen properties dialog box.

Field	Description
Used as Pattern	When this box is checked, the screen may be used as a pattern for any project screen. A pattern screen cannot contain animated objects.
Uses pattern	When this box is checked, the background of this screen uses the background pattern that you have defined. The right button can be used to select the model screen.
Default color	When this box is checked, the screen background is the default color used by Windows software for windows.
Custom color	When this box is checked, the screen background is the color you defined (color of the field located on right side of the check box). The button allows access to the screen color settings.
Size	Screen sizes. It can be changed and its default value is 1280 pixels by 1024 pixels.

Tab: Information

The following table describes the different functions offered by the **Information** tab of the property dialog box of a screen.

Field	Description
Information	This information cannot be modified; it indicates: • The screen creation date • The last screen modification date • The number of objects contained in the screen • The number of variables used in the screen

Properties of Operator Screen Families

At a Glance

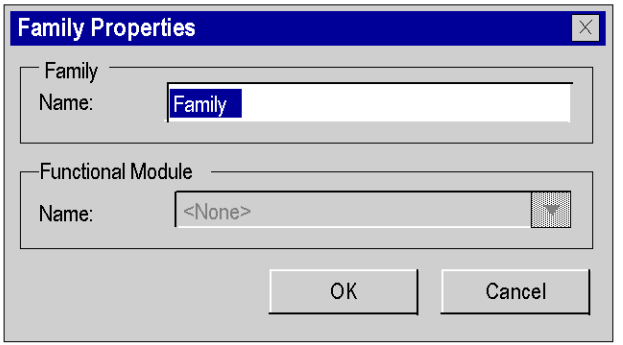
When a screen family is selected, the dialog box accessible via the **Edit > Family Properties** command allows you to:

- modify the name of a screen family

- associate the family with a functional module.

Properties dialog box of a family

The properties dialog box of a family is as follows.



Description

The following table describes the different functions offered by the properties dialog box of a family.

Field	Description
Family	Name of the family to which the screen belongs. It can be modified and can contain up to 200 characters.
Functional Module	Name of the functional module with which the family is associated. The button to the right of the field is used to access the list of existing functional modules.

NOTE: A functional module cannot be created from this dialog box.

Creating an object

Subject of this section

This section describes how to create or insert an object.

Introduction of a graphic screen object

General

4 types of objects can be created in a graphic screen :

- standard objects: line, rectangle, ellipse, curve, polyline, text,
- images: bitmap files with the **BMP** or **JPG**extension,
- the driving (or command) objects: button, check box, input field, counter, cursor, explicit exchange object, screen browser button,
- compound objects: group of objects of previous 3 types, created by the user or from the object library.

All the objects used to create a screen can be static or animated.

How to create standard objects

At a Glance

Standard objects are the basic elements of graphic screens.

They are the simple objects which allow the composition of complex objects.

There are two types of standard objects:

- Static objects
- Dynamic or animated objects

Static objects do not have variables associated with them. Their graphic representation is fixed.

Dynamic objects possess one associated variable, which allows their display to be modified.

How to create a standard object

The create standard object procedure is identical regardless of the object, with the exception of polygons.

Step	Action
1	Select the type of object that you wish to create from the tool palette.
2	Move the mouse cursor to the position where you want to place the top left corner of your object.
3	Press down the left mouse button.
4	Hold the button down and move the mouse cursor to the position where you want to place the bottom right corner of the object.
5	Release the button. **Result**: The object is created when you release the mouse button
6	Right click on the object. **Result**: The contextual menu appears
7	Enable the **Properties** command. **Result**: A contextual dialog box appears This box depends on the type of object and allows the attributes, page 1591 to be fixed.

How to create a polygon object

The procedure for creating a polygon object is as follows.

Step	Action
1	Select the type of polygon object that you require from the tool palette.
2	Move the mouse cursor to the position where you want to place the first point of your object.
3	Press the left mouse button then release. **Result**: The object's start point is created
4	Move the mouse cursor to the position where you want to end the first line.
5	Press the left mouse button then release. **Result**: You have just created the first line of your object
6	Repeat steps 4 and 5 to create new lines to form your polygon object. **Result**: Segments of consecutive straight lines are created
7	Finish by left clicking on the end point of the object.

Step	Action
	Result: The object is created
8	Right click on the object. **Result**: The contextual menu appears
9	Enable the **Properties** command. **Result**: A contextual dialog box appears This box depends on the type of object and allows the attributes, page 1591 to be fixed.

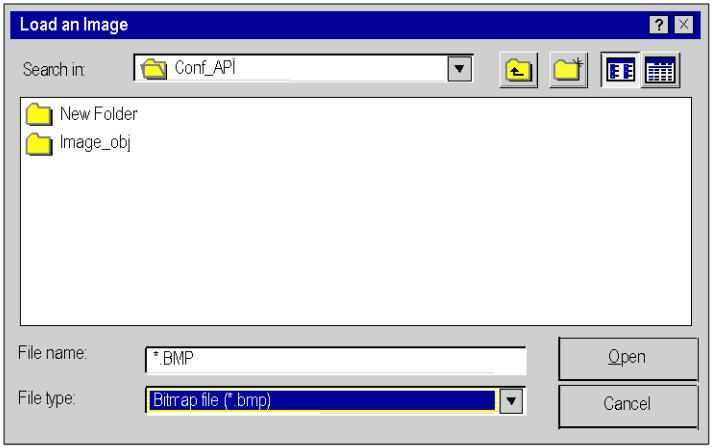
How to insert an image in a screen

At a Glance

An operator screen can contain one or several bmp, jpg or jpeg format images.

How to insert an image object in a screen

Follow the steps below to insert an image in a screen.

Step	Action
1	Select the image icon from the tool palette. **Result:** the selection window of the bitmap file to be inserted appears 
2	Select the type of file to display (bmp, jpg or jpeg).
3	Select the file to be inserted and enable the **Open** command. **Result:** the image is inserted in the screen It appears with selection handles.

Rules and precautions for use

Only images in bmp, jpg or jpeg formats are recognized.

Optimize loading time:

- Do not integrate too many images into one screen
- Limit the size of images
- Avoid modifying the image's proportions and size in the screen

Optimize display quality:

- Avoid modifying the image's proportions and size in the screen
- If a screen contains several different images, use 24 bit screen depth

If the image, page 1608 is not copied in the project, it will not be exported, during a screen export, page 1713.

How to create driving objects

At a Glance

Driving (or command) objets allow the user to perform an action:

- Browse from one screen to another
- Modify the value of a variable
- Send a command to the PLCs task module

There are 7 types of these objects:

- Buttons
- Check boxes
- Input zones
- Counters
- Cursors
- Buttons for browsing between screens
- Objects for explicit exchanges

How to create a driving object

The procedure for creating a driving object is as follows.

Step	Action
1	Select the type of driving object that you wish to create from the tool palette.
2	Move the mouse cursor to the position where you want to place the top left corner of your object.
3	Press down the left mouse button.
4	Hold the button down and move the mouse cursor to the position where you want to place the bottom right corner of the object.
5	Release the button. **Result:** The object is created when you release the mouse button
6	Enable the **Selection** tool from the tool palette.

Step	Action
7	Right click on the object. **Result:** The contextual menu appears
8	Enable the **Properties** command. **Result:** A contextual dialog box appears This box depends on the type of object and allows the attributes, page 1591 to be fixed.

Type of data driven according to the driving object

The table indicates the types of variables that may be driven by the driving objects.

Object	Bit	Byte	Word	Double word	Floating point
Button	X	X*	X*	X*	X*
Check box	X	-	-	-	-
Entry field	X	X	X	X	X
Counter	X	X	X	X	X
Cursor	X	X	X	X	X
Legend:					
* : Only in the case of unsynchronized buttons					

NOTE: Driving objects are always placed in the foreground. To prevent command errors, we strongly advise against placing driving objects between them.

Operation of the driving objects

At a Glance

The piloting objects are enabled by clicking on the mouse (or via the keyboard). Depending on the fixed attribute, these objects act on their associated variables.

NOTE: The state of associated variables can be changed by the PLC application. The action of the driven object doesn't constitute a forcing of the variable.

Furthermore, so long as the input of a value has not been validated or that the driven object maintains control, the displayed value may differ from the real value of the variable contained in the PLC memory.

How the buttons operate

Depending on its attributes, a button works as follows.

When you click on the button, if the associated variable is...	and if	Then
a bit,	driving is unsynchronized,	the associated variable switches to 1 when the left mouse button is held down, and switches back to 0 when released.
digital,	driving is unsynchronized,	the associated variable takes the fixed threshold value in the button attributes.
a bit,	driving is synchronized,	the state of the associated variable changes. It moves to 0 if it was at 1, and vice versa.

How a check box operates

A check box works as follows.

When the box	Then
is checked,	the associated bit is set to 1.
isn’t checked,	the associated bit is set to 0.

How an input zone operates

An input zone works as follows.

When	and if	Then
the entry zone contains a value,	you press **Enter**,	the associated variable takes the value read in the entry zone.
the associated variable is modified by the PLC,	you did not enter a value in the entry zone,	the zone displays the value read in the PLC.

How a counter operates

A counter operates as follows.

When	and if	Then
the counter is horizontal,	you click on the right arrow of the counter,	the associated variable is incremented by one unit.
the counter is horizontal,	you click on the left arrow of the counter,	the associated variable is decreased by one unit.
the counter is vertical,	you click on the top arrow of the counter,	the associated variable is incremented by one unit.
the counter is vertical,	you click on the bottom arrow of the counter,	the associated variable is decreased by one unit.

NOTE: The increment and decrement value is multiplied by 10, 100 then 1000 when the arrow is held down.

How a cursor operates

A cursor operates as follows.

When the cursor	and if	Then
is horizontal,	it is moved to the right,	the associated variable is incremented proportionally to the move made and to the minimum and maximum thresholds defined for the variable.
is horizontal,	it is moved to the left,	the associated variable is decremented proportionally to the move made and to the minimum and maximum thresholds defined for the variable.
is vertical,	it is moved upwards,	the associated variable is incremented proportionally to the move made and to the minimum and maximum thresholds defined for the variable.
is vertical,	it is moved downwards,	the associated variable is decremented proportionally to the move made and to the minimum and maximum thresholds defined for the variable.

How an inter-screen browser button operates

An inter-screen browser button operates as follows.

When the button	Then
is enabled by a click,	the screen with the associated number is displayed.

NOTE: In the case of Multi-Instance, page 591 mode, clicking on the browser button opens a new window with the specified screen. If this is not the case, the new screen opens in the current window.

How an explicit exchange button operates

An explicit exchange button operates as follows.

When the button	Then
is enabled by a click,	the associated explicit exchange is launched. The memory zone of the PLC and that of the module are updated according to the configured, page 1610 parameters.

Default Properties screen

Subject of this section

This section present the Default Properties screen of the operators screen, its elements and its behaviour

Default Properties screen

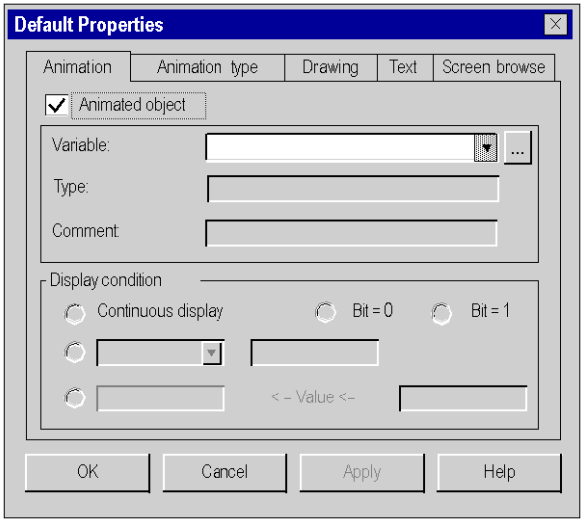
At a Glance

The Default properties screen allows to set common attributes for all objects you will place in the operator screen.

You can access this screen by right-clicking on the operator screen and select properties in the menu. (No object must be selected)

Default properties screen

The following screen shows the different tabs of the default properties screen. (In this example animation tab is visible)



Description

To see the description of all the tabs refer to object properties tab screens.

- Animation tab, page 1596
- Animation type tab, page 1598
- Drawing tab, page 1592
- Text tab, page 1594
- Screen browse tab, page 1604

Behaviour of the Default Properties screen

When you modificate attributes in the Default Properties screen, the new attributes are taken into account only for the objects you will place in the operator screen after the modifications. If there are already objects in the operator screens, the modifications will not impact them. In this case you have to modify the attributes for each object (or compound of objects).

Modification of an object’s attributes

Subject of this section

This section describes the attributes of objects, how to access these attributes and the associated information.

How to access an object’s attributes

At a Glance

Each object created in a screen has attributes which depend on its type. These attributes are accessed via the tabs below.

- Animation, page 1596
- Type of animation, page 1598
- Design, page 1592
- Text, page 1594
- Image, page 1608
- Driving, page 1606
- Screen browsing, page 1604
- Explicit exchanges, page 1610

Access to these attributes is the same for all created objects.

How to access an object’s attributes

The table below indicates how to access the settings of object attributes.

Step	Action
1	Select the object whose attributes you wish to modify.
2	Click on the object with the right mouse button. **Result:** The contextual menu appears
3	Enable the **Properties** command. **Result:** A dialog box displays the attributes linked to the kind of object chosen.

Design tab

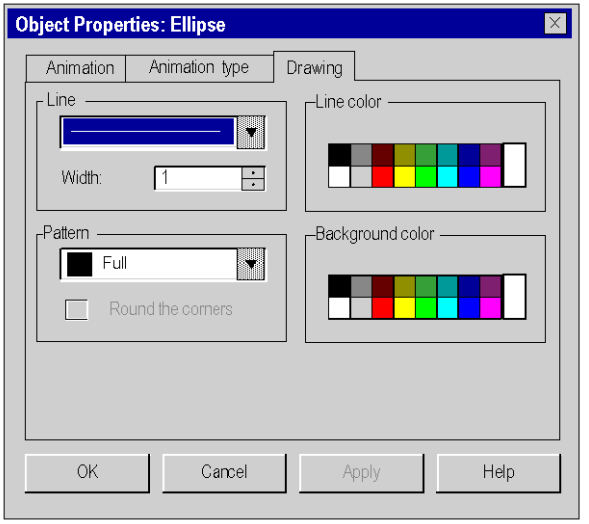
At a Glance

This tab is used to set the design attributes of a graphic object:

- Modification of the object's contour line
- Modification of the object's pattern and color

Design attributes setting screen

The screen below is used to set the design attributes of a rectangle type graphic object.



Description

The table below describes the settings that you may define.

Field	Description
Line	Contour line of the graphic object: • Single line • Single or complex dotted line • No contour line
Thickness	Thickness of the object's contour line. It is defined in pixels and may range from 1 to 10 pixels maximum.
Outline color	Color of contour line. The color palette contains 16 predefined colors and provides access to the extended colors, page 1593 parameter screen.
Pattern	The pattern inside the object. This pattern is defined by a frame and a background color. The available frames are as follows: • None (the background is transparent) • Fill (filled with the selected background color) • Horizontal frame • Vertical frame • Diagonal frame to the top • Diagonal frame to the bottom • Grid • Lattice
Round the corners	When this box is checked, the corners of rectangle type objects are rounded.
Background color	Color of the object's content (background). The colors palette contains 16 predefined colors and provides access to the extended colors, page 1593 parameter screen.

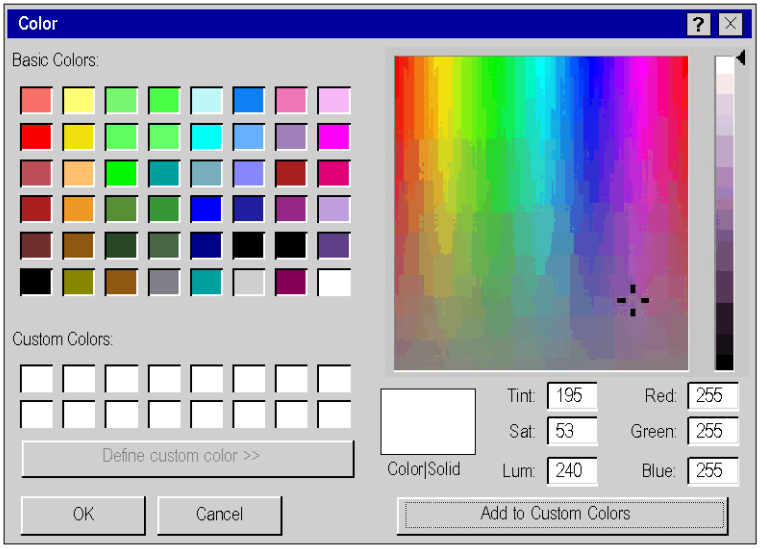
Extended colors

At a Glance

This dialog box proposes 48 basic colors and allows you to create personalized (RGB) colors enhanced with tint, saturation and brightness.

Color parameter screen

The following screen allows you to create or modify the extended colors.



Creating a personalized color

This table describes how to create a new color, that can be used for the graphic objects.

Step	Action
1	Position the cross hair for selecting color, tint and saturation to find the required shade.
2	Adjust (up or down) the light and color selection cursor to refine your choice.
3	Select a box from **Custom Color**.
4	Click on the **Add to custom colors** button to create a new color.

Text tab

At a Glance

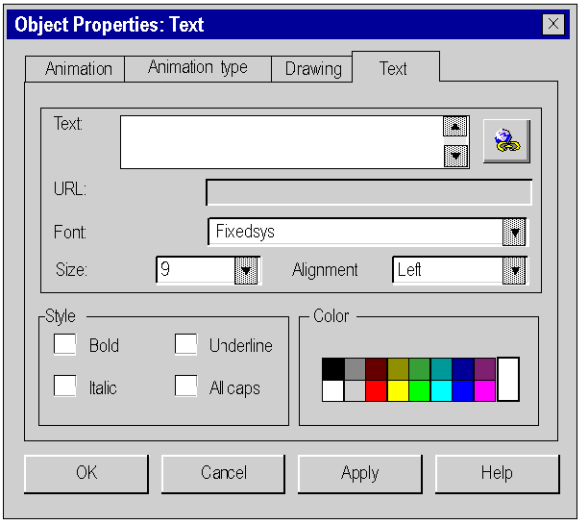
This tab is used to set the text attributes of a graphic object:

- text input,

- associate a hyperlink,
- select the character font,
- select font size,
- set text alignment,
- select text style,
- select text color.

Text attributes setting screen

The screen below is used to set the text attributes of a graphic object.



Description

The table below describes the settings that you may define.

Field	Description
Text	Text of the object. 32635 maximum characters entered on several lines, separated by a carriage return. If the related variable is of the real type, users may define a display format as #XX.XXX# (in this case a 5 figure display with three decimals).
button 	Allows access to the dialog box to define the hyperlink. This hyperlink, page 1774 will be associated with the text attribute of the graphic object.

Field	Description
Url	Static display of the hyperlink defined by the  button.
Font	Character font. The fonts available for selection are those installed in Windows.
Size	Font size.
Alignment	Text alignment: align left, right or centered in relation to the display area.
Style	Text style: bold, underlined, italics or capitals.
Color	Text color. The colors palette contains 16 predefined colors and provides access to the extended colors, page 1593 parameter screen.

Animation Tab

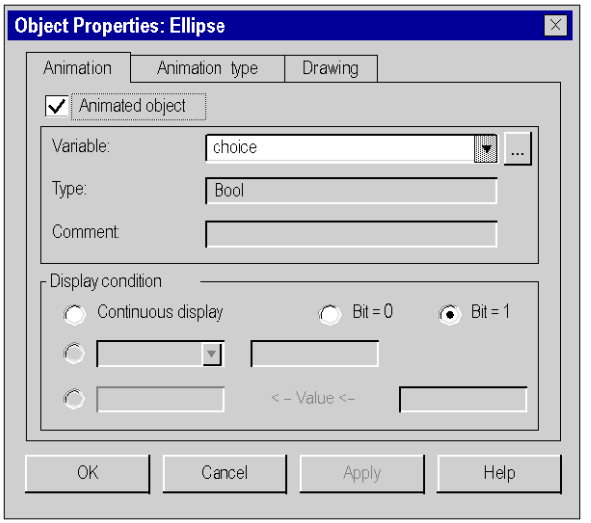
At a Glance

All the graphic objects (simple or compound) that you create in a screen can be animated. This tab allows users to set the animation attributes of a graphic object.

- to decide to animate an object,
- to choose the animation variable,
- to choose the display conditions.

Animation attributes setting screen

The following screen is used to set the animation attributes of a graphic object.



Description

The table below describes the settings that you may define.

Field	Description
Animated object	The object is animated if this box is checked.
button 	Calls the variable selection tool to select the variable to associate with the object.
Variable	The animation variable linked to the object. Possibilities: • simple variables, • the public variables of a function block, • the parameters (input, output, input/out) of a function block.
Type	Type of animation variable linked to the object. The type must be EDT (Bool, Ebool, Int, Dint, Uint, Udint, Real, Time...). It is not possible to use structures or tables. However, it is possible to use items from tables or structures if they are of type EDT.

Field	Description
Comment	Comment of the animation variable linked to the object.
Display condition	The object's display condition: • Continuous display: the graphic object is always displayed on the screen, • Bit = 0: the graphic object is displayed if the associated variable (bit) is in state 0, • Bit = 1: the graphic object is displayed if the associated variable (bit) is in state 1, • <=, >=, =, <>: the graphic object is displayed if the defined condition is true. • <= value <=: the graphic object is displayed if the defined condition is true.

Animation type tab

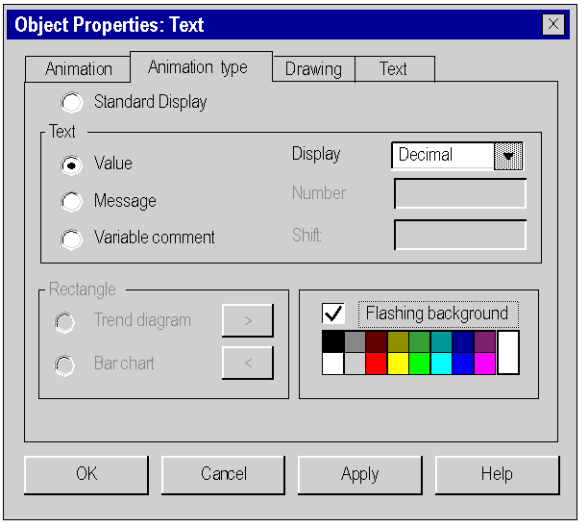
At a Glance

All the graphic objects (simple or compound) that you create in a screen can be animated. This tab is used to set the animation type which depends on the nature of the graphic object to be animated:

- to select the object's display mode,
- to select the display mode of the text linked to the object,
- to select flashing option,
- to select the trend diagram mode,
- to choose bar chart mode.

Animation type attributes setting screen

The following below is used to set the animation type attributes of a graphic object.



Description

The table below describes the settings that you may define.

Field	Description
Standard Display	The object is displayed with its attributes if this button is selected.
Value	The value taken by the variable is displayed according to the display mode chosen. In offline mode, even if this animation style is selected, the text defined in the **Text** field of the object is displayed on the screen with the selected font and size.
Display	Choice of display mode: • Decimal, • Hexadecimal (value preceded by 16#), • Binary (value preceded by 2#), • ASCII. When the length of the object's definition zone is too small to display the value, the field is filled with **#** characters. The font used is the one defined in the Text tab, page 1594. If the animation variable is of the **String** type (string of characters), the **Display** field cannot be accessed because viewing is automatically in ASCII.
Message	You must select this button to display a message configured in the Operator screens messages editor, page 1624. The **Number** field allows you to enter either a message

Field	Description
	number, or the term Val. In the last case, the displayed message will be the one whose value is contained in the variable associated with the object. The **Shift** field allows you to enter a value that will be added to the one contained in the variable.
Variable comment	If this button is selected and if the text animation condition is met, the comment associated with the variable is displayed. **Note**: This function allows, among other things, to avoid a double entry. Enter the comment once in the variables editor. This will be used as a message linked to the variable.
Trend diagram	You may use rectangle type objects such as trend diagrams, page 1601. Selecting this box will give you access to the trend diagram's parametering box.
Bar chart	You may use rectangle type objects such as bar charts, page 1602. Selecting this box will give you access to the bar chart's parametering box.
Flashing background	When this box is selected and if the display condition is met, the background of the graphic object flashes according to the user-specified flashing color.

Properties

The table below shows the types of animation that may be associated with the driving objects.

Object	Standard	Value	Message	Trend diagram	Bar chart	Flashing
Line	X	-	-	-	-	X
Rectangle	X	-	-	X	X	X
Ellipse	X	-	-	-	-	X
Curve	X	-	-	-	-	X
Polygon	X	-	-	-	-	X
Text	X	X	X	-	-	X
Compound object	X	-	-	-	-	-
Image	X	-	-	-	-	-
Control	X	-	-	-	-	-
Legend:						
X: Yes - : No						

Trend diagrams

At a Glance

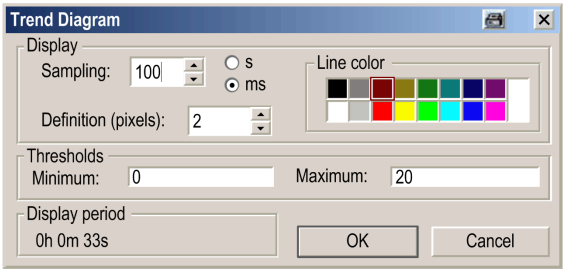
A trend diagram allows the graphic representation of a variable status, with the representative curve moving from right to left.

You are advised to limit the number of trend diagrams on a same page because their display decreases the system's performance.

Only rectangular type objects can be used as trend diagrams.

Trend diagram parameter screen

The screen below allows to parameter the animated rectangles of the trend diagram format.



Description

The table below describes the settings that you may define.

Attribute	Description
Sampling	Period in which a new value is added at the end of the diagram and the diagram is refreshed. Default value: 1 second. When the sampling period is given in seconds, values between 1 and 3600 are allowed. When it is given in milliseconds, the allowed range is 1 to 9999. Sampling periods below 100 ms produce a lot of windows messages and generate a high CPU load. Therefore only few diagrams with such small periods should be animated at the same time and long lasting actions like **Build Changes** should be avoided while diagrams are animated (see also the **NOTE** at the end of the table).
Definition	Length of value display. Each new value is represented by a right segment whose length is fixed by the definition.

Attribute	Description
	The definition is expressed in pixels between 1 and 20. The value by default is 2.
Line color	Color of the timing diagram's right segments The colors palette contains 16 predefined colors and provides access to the extended colors, page 1593 parameter screen.
Threshold: minimum and maximum	Thresholds between which the value representation develops. For binary values, the thresholds must be 0 and 1.
Display period	This value is calculated. It depends on the size of the rectangle, the sampling value and the definition. The procedure is as follows: Period = Rectangle size / Definition x Sampling.

NOTE: Internally, for every diagram, a timer is started with the sampling period defined by the user. When the timer fires, the last value of the corresponding variable transmitted from the PLC is read and added at the end of the diagram. Such timers may not always fire exactly in equidistant periods (e.g. they may be delayed when a lot of tasks run in parallel on the PC). Therefore the displayed values may not represent the exact values in the PLC at the end of each sampling period.

Bar chart

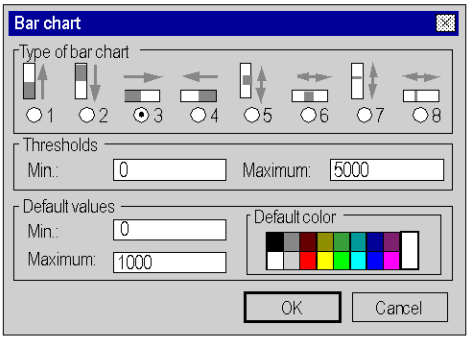
At a Glance

Bar charts allow the graphic representation, in rectangular format, of the status of a variable.

Only rectangular type objects can be used as bar charts.

Bar chart parameter screen

The screen below allows parameterizing of animated rectangles in bar chart format.



Description

The table below describes the settings that you may define.

Attribute	Description
Type of bar chart	1. The bar chart moves from bottom to top. 2. The bar chart moves from top to bottom. 3. The bar chart moves from left to right. 4. The bar chart moves from right to left. 5. The bar chart moves symmetrically in relation to a horizontal axis corresponding to the average of the thresholds. 6. The bar chart moves symmetrically in relation to a vertical axis corresponding to the average of the thresholds. 7. The bar charts grow vertically, by symmetrically filling the gap between the value of the variable and the average of the thresholds. 8. The bar charts grow horizontally, by symmetrically filling the gap between the value of the variable and the average of the thresholds.
Thresholds	Values indicating the range of status displayed. When the associated variable value is equal to the minimum threshold value, the bar chart is empty. When this value is equal to the maximum limit value, the bar chart is full (the rectangle's color). By default these values are set to 0 (minimum) and 1000 (maximum).
Default values	Values that indicate the thresholds from which the bar chart is displayed with the default color. The bar chart uses the default color when the associated variable value is: • higher than or equal to the maximum value • lower than or equal to the minimum value The colors palette contains 16 predefined colors and provides access to the extended colors, page 1593 parameter screen.

Screen browse tab

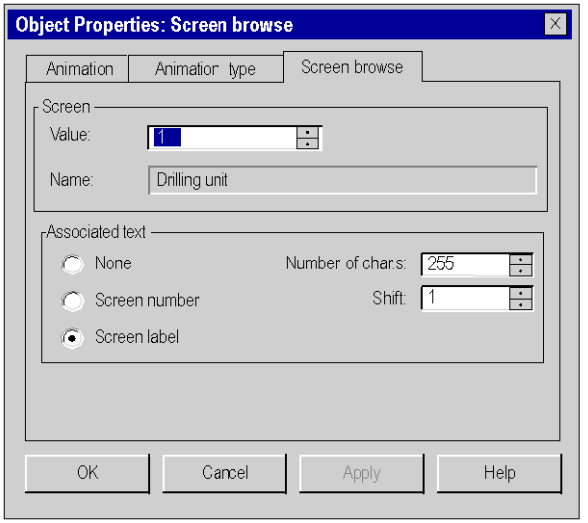
At a Glance

The screen browse buttons allow users to browse between screens (move from one screen to another). The parametering of a button's attributes links the button to the screen. The button's browser attributes allow users to:

- choose the screen to be displayed,
- define the text displayed in the button,

Screen browse attributes setting screen

The screen below allows users to set the animation attributes of a screen browse button.



Description

The table below describes the different parameters that you may define.

Field	Description
Value	This field allows entry of the screen that is displayed when you click on the button. Two cases are possible: • the value entered is a number that indicates the screen number to display, • you have entered the term **VAL**. The screen displayed will be the one whose number is contained in the animation variable linked to the button object (See Animation Tab, page 1596).
Name	This field indicates the label of the screen you have defined by its number in the **Value** field. If you have defined the screen by the term **VAL**, this field displays the characters **## ? ##**.
None	If this box is checked, no text is associated with this button.
Screen number	If this box is checked, the screen number is displayed on the button.
Screen label	If this box is checked, the screen label is displayed on the button.

Field	Description
Number of char.s	If the **Screen label** box is checked, the field allows you to define the number of characters to display in the button.
Shift	If the **Screen label** box is checked, the field indicates the position (shift) of the first character of the label to display in the button.

NOTE: Each browser button has a logical number that links it to a screen. If this number is modified, the button will no longer point on the screen.

NOTE: The software does not take into account the deletion or import of screens.

Control tab

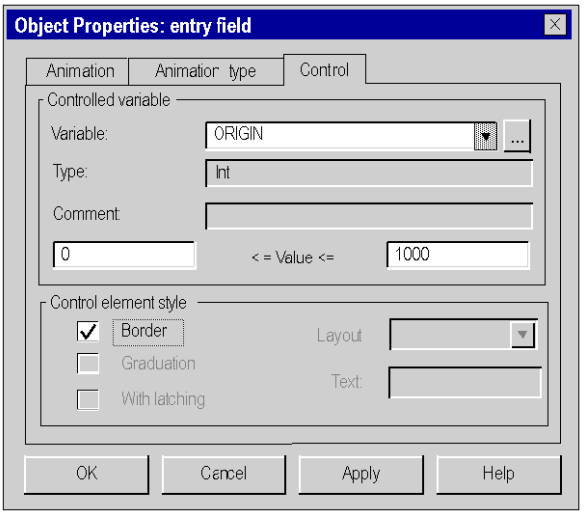
At a Glance

This tab allows you to define the characteristics of a control object:

- to choose a control variable,
- to choose the minimum and maximum values,
- to determine the monitoring style.

Control attributes setting screen

The screen below allows users to set the control attributes of a command graphic object.



Description

The table below describes the settings that you may define.

Field	Description
Variable	Variable controlled by the graphic object.
button ...	Calls the variable selection tool to select the variable to associate with the object.
Type	Type of variable linked to the object.
Comment	Comment of the variable linked to the object.

Field	Description
< = value <=	The control object will not exceed these thresholds. By default, the minimum value is 0 and the maximum value is 100. If the variable to control is a bit, these fields are not accessible.
Control element style	Control style which depends on the type of control object. It allows you to modify the representation of the graphical object: • **Border** displays a boarder around the object, • **Graduation** displays color graduations, • **With latching** denotes a button with latching, • **Orientation** determines the orientation of the object on the screen, • **Text** displays the text with the object.

NOTE: If the graphic object is a button, and if the control variable is a word, a double word or a real word, the maximum threshold becomes a **dynamic threshold**. It is the value of this threshold that is sent when you click on the button

Image Tab

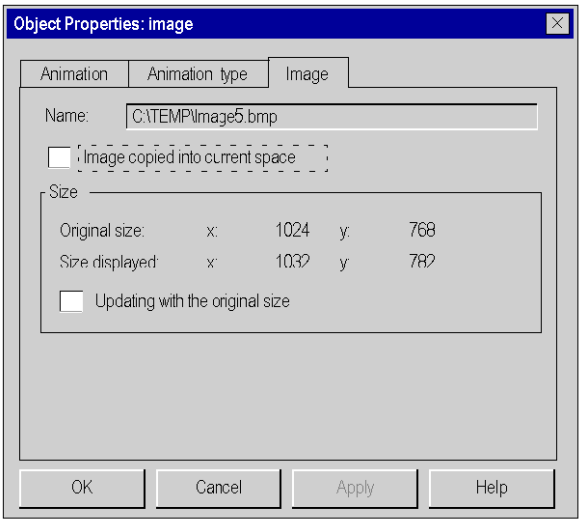
At a Glance

This tab allows you to set the attributes of the .BMP, JPG or JPEG format images built into the operator screens:

- To choose the image to be displayed,
- To integrate the image into the project,
- To query the size of the image (real and visual).

Image Attributes Setting Screen

The screen below allows you to set the attributes of a bitmap image built into an operator screen.



Description

The table below describes the settings that you may define.

Field	Description
Name	This field indicates the name and the access path to the file.
Image copied into current space	When this box is checked, the file is copied and compressed in the project file (.STU or STA for an archive). Loading is faster when using the screens and the image is exported at the same time as the screen during an export, page 1713. If the image is already built into the project, this box does not appear.
Size	This information, accessible in read only gives information on the original size and the displayed size of the image. Its is used to know if the image has been modified.
Updating with the original size	When this box is checked, the image changes to its original size. This box does not appear if the original size and the displayed size of the image are identical.

Explicit Exchanges Tab

At a Glance

Some Bits and Words Are Not Implicitly Exchanged (at Each PLC Cycle) Between the Modules and the PLC Processor. These bits and words are grouped into 3 families: information on status (STATUS), command (CMD) and parametering (PARAM) and are explicitly exchanged.

Explicit exchanges are normally made by program, but the user can also initiate such an exchange by clicking on a button, located in an operator screen.

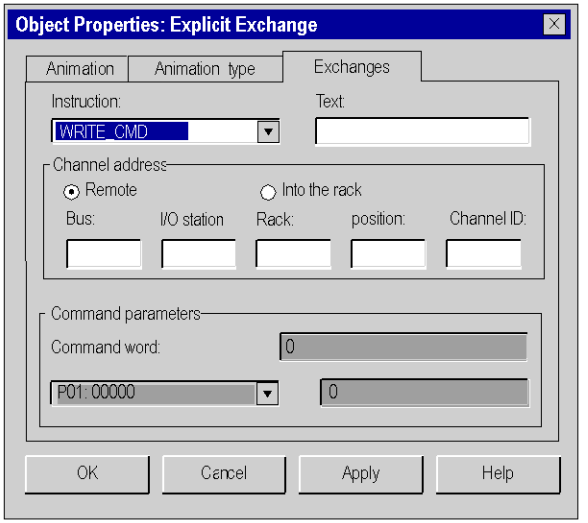
The explicit **exchange** tab allows you to define the exchange characteristics:

- to select an explicit exchange instruction,
- to select the text of the button,
- to select the address of the channel to which the exchange will be applied,
- To display the parameters associated with the instruction.

NOTE: Only the exchanges of objects associated with a **channel** are authorized.

Explicit Exchanges Attributes Setting Screen

The screen below allows you to set the attributes of an explicit exchange object.



Description

The table below describes the settings that you may define.

Field	Description
Instruction	Explicit exchange instruction: • READ_STS: reading of channel status words contained in the module and updating the data in the PLC. • WRITE_PARAM: writing of channel parametering words, contained in the module. • READ_PARAM: reading of channel parametering words, contained in the module. • SAVE_PARAM: backup in a specific memory space of channel parametering words, contained in the module. • RESTORE_PARAM: restoration of values saved by SAVE_PARAM. • WRITE_CMD: implements the execution of a command on a channel.
Text	Text that appears on the button.
Channel address	**Remote**: this box is checked if the module to be interrogated is on a remote station. **In rack**: this box is checked if the module to be interrogated is on the rack of the local station. Address of the module channel. This address has the form: • \bus.station\rack.location.channel: ◦ bus: bus number (shaded if local station), ◦ station: number of the device connection point (shaded if local station), ◦ rack: station rack number, ◦ position: position of the module on the rack, ◦ channel: channel number.

NOTE: For Write_Param and for Write_Cmd we have to modified the associated %MW by animation table or another one (by program, by 'Entry field' through operator screen, ...) before sending the instruction (Write_Param or Write_Cmd) through the Operator screen.

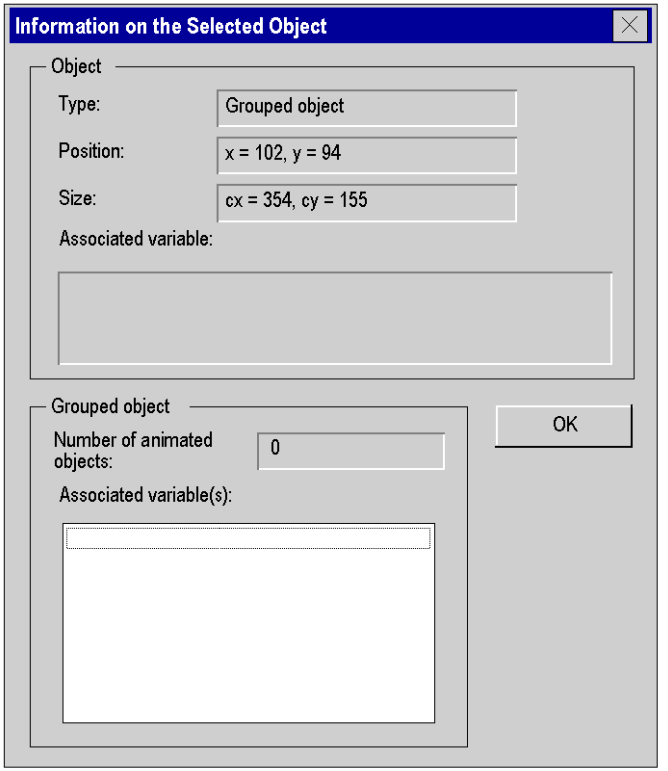
Information on the objects

At a Glance

The information screen on the objects, accessible via the **Edit > Information** command allows the information on each of the objects contained in a screen to be displayed. This screen is particularly useful for compound objects.

Information screen on an object

The screen below provides information on a graphic object.



Description

The following table describes the information offered by this screen.

Information	Description
Type	Type of object. ellipse, button, cursor, object compound, etc.
Position	Coordinates of the object's top left point in pixels.
Size	Size of the object in pixels.

Information	Description
Associated variable	Variable linked to a simple object or controlled by a simple object. In the case of a compound object, this field is left blank. The information is placed in the **Grouped object** zone.
Grouped object	Number of simple animated objects contained in a compound object and the variables associated with (or controlled by) this compound object. In the case of a simple-type object, this field is not displayed.

The data selection tool

At a Glance

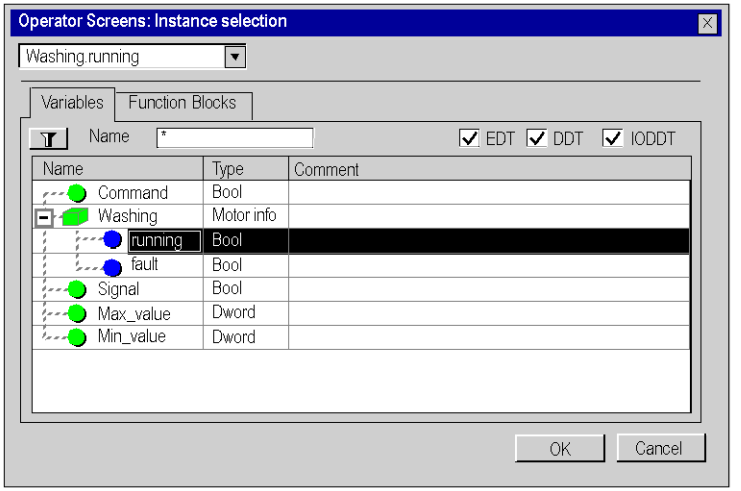
All the graphic objects (simples or compound) that you create in a screen can be animated, page 1596. The variables selection tool allows you to associate a variable to animate the object. Variables that can animate objects include:

- simple variables, (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual)
- the public variables (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual)of a function block,
- the parameters (input, output, input/out) (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual)of a function block.

Procedure

The table below presents the procedure for using a variable to animate an object.

Step	Action
1	Select to object to parameter in the operator screen.
2	Right click in the contextual menu to select the **Properties** command. **Result:** A dialog box displays the attributes linked to the kind of object chosen.
3	Select the animation tab.
4	Check the **Animated Object** box.
5	In the **Variable** field enter the name of the variable or use the  button to open the variables selection tool. **Result:** The variables selection tool appears.

Step	Action
	
6	From the **Variables** tab or the **Function Blocks** tab select the variable to be used to animate the object.
7	Click **OK** to confirm.

Variables tab

The table below describes the elements of the **Variables** tab of the data selection tool.

Field	Description
Button 	Clicking on this button opens a dialog box, page 433 which you can use to refine the filter.
Name	Column in which you enter the name (symbol) of the instance or data type that you want to display. You can use wild cards (***** or **?**).
EDT	Authorizes the display of the Elementary Data Types.
DDT	Authorizes the display of Derived Data Types.
IODDT	Authorizes the display of the derived data concerning inputs and outputs.
Name	Column in which the name (symbol) of the instance or data type is located.

Field	Description
Type	Column in which the data type is located.
Comment	Column in which the data comment is located.

Function Blocks tab

The table below describes the elements of the **Function Blocks** tab of the data selection tool.

Field	Description
Button 	Clicking on this button opens a dialog box, page 433 which you can use to refine the filter.
Name	Column in which you enter the name (symbol) of the instance or data type that you want to display. You can use wild cards (***** or **?**).
Named by default	Authorizes the display of data named by default
EFB	Authorizes the display of Elementary Function Block data types.
DFB	Authorizes the display of Derived FB data types.
Name	Column in which the name (symbol) of the instance or data type is located.
Type	Column in which the data type is located.
Comment	Column in which the data comment is located.

Manipulation of objects in runtime screens

Subject of this section

This section describes how to manipulate objects in the runtime screens. how to select objects, move objects, duplicate or delete objects,...

How to select graphic objects

At a Glance

When objects have been created in a screen, you may have to select some of them in order to position them, regroup them, etc. The selection or de-selection of graphic objects is one of the basic tasks during the construction of an operator screen.

How to select a single object

The following table describes the operations to perform in order to select an object.

If you want to select	Then
an isolated object.	validate selection mode then click on the object with the left mouse button.
an object among several overlaid objects.	validate the selection mode, press **ALT** then right mouse click several times on the area occupied by the objects. **Result**: You select each object one after the other even they are hidden (in the background).
successive objects on a screen.	press the space bar successively. **Result**: Each time you press the spacebar, you select the next object even if it is hidden behind another object.

How to select several objects

The following table describes the operations to perform in order to select several objects at the same time.

If you want to select	Then
several contiguous objects.	frame the objects to be selected by defining a selection zone (this operate mode is identical to that for the creation of objects). **Result**: A rectangle with handles is displayed on the screen. It regroups all the objects located in the selection zone.
several objects scattered in the operator screen.	press **Shift** then click on the objects to be selected while keeping the **Shift** key held down. **Result**: A rectangle with handles is displayed on the screen, it groups all the selected objects.

How to deselect a selected object or a group of objects

The following table describes the operations to perform in order to deselect an object.

Step	Action
1	Press **Shift** and hold down this key.
2	Left click on the selected object that you want to deselect. **Result**: The selection handles associated with the object disappear, indicating that the object is no longer part of the selection.

NOTE: Press **ESC** to delete the entire active selection. Clicking on the screen background gives the same results.

How to move and resize objects

Introduction

You may have to move or resize objects created in a screen, if any .

How to use the mouse to move objects

The following table describes the operations to perform in order to move one or several objects with the mouse.

Step	Action
1	Select the object or objects to be moved.
2	Click with the left mouse button on a selected objects or in the rectangle representing the selection.
3	Keep the button held down.
4	Drag the selected object or the rectangle to the desired position. **Result**: the selection is represented by a dotted zone as you move it.
5	Release the mouse button. **Result**: the selection is placed in its new location.

How to use the keyboard to move objects

The following table describes the operations to perform in order to move one or several objects with the keyboard.

If you want to move	Then
precisely one or several objects by using the keyboard	after having selected one or several objects to move, press as many times as necessary on the keyboard direction (arrow) keys. **Result**: the selection represented by a dotted zone can be moved pixel by pixel or according to the steps of the grid if it is active.
rapidly one or several objects by using the keyboard	after having selected one or several objects to move, press on the **Shift** key, then while holding down this key press as many times as necessary on the keyboard direction (arrow) keys. **Result**: the selection represented by a dotted zone can be moved in 10 pixel-steps.

How to resize objects

The following table describes the operations to perform to resize one or several objects.

Step	Action
1	Select the object or objects to be modified
2	Drag the handles in the desired directions in order to resize the frame. **Result**: the selection represented by a dotted zone is resized as you move the mouse.

How to delete and copy objects

At a Glance

You may need to delete a graphic object or duplicate the same object in a screen.

How to delete objects

The following table describes the operations to perform in order to delete one or several objects.

Step	Action
1	Select the object or objects to be deleted. **Result**: The selection is represented by a frame with handles.
2	Press **Delete**. **Result**: The selected objects are deleted. They are not memorized in the clipboard.

How to use the mouse to copy objects

The following table describes the operations to perform in order to copy one or several objects with the mouse.

Step	Action
1	Select the object or objects to be copied.
2	Click with the left mouse button on one of the selected objects or in the rectangle representing the selection.
3	Hold down the button and press **CTRL**.
4	Drag the selected object or the rectangle to the desired position. **Result**: The selection is represented by a dotted zone as you move it.
5	Release the mouse button. **Result**: The copy of the selection is placed in its chosen location.

How to cut, copy or paste objects

The table below indicates the procedures for cutting, copying or pasting one or several objects.

Step	Action
1	Select the object or objects to manipulate. **Result**: The selection is represented by a frame with handles.
2	Select the command to execute: • **Edit > Cut** • **Edit > Copy** • **Edit > Paste** **Result**: **Cut** deletes the selected item(s) and memorizes it (them) on the clipboard. **Copy** duplicates the selected items in the clipboard and **Paste** duplicates the clipboard contents on the screen.

How to position objects

At a Glance

You may have to position an object in relation to another in a screen:

- to align the objects,
- to make objects symmetrical,
- to position the objects in the foreground and the background in relation to the others.

How to align a group of objects

The table below describes the procedure for aligning a group of objects.

Step	Action
1	Select the objects to align. **Result**: The selection is represented by a frame with handles.
2	Select the command to execute in the toolbar: • Left alignment, • Right alignment, • Upper alignment, • Lower alignment. **Result**: The selected objects are aligned according to what has been chosen.

How to make an object or a group of objects symmetrical

The table below describes the procedure for flipping an object or a group of objects vertically or horizontally.

Step	Action
1	Select the object or objects to be flipped. **Result**: The selection is represented by a frame with handles.
2	Select the flip command in the toolbar: • vertical flip, • horizontal flip. **Result**: The selected object or group of objects is positioned according to the chosen symmetry.

How to put an object in the foreground

The table below describes the procedure for placing an object in the foreground.

Step	Action
1	Select the object to place in the foreground. **Result**: The selected object is equipped with handles.
2	Select the **Foreground** command in the toolbar. **Result**: The selected object is positioned in the foreground.

How to put an object in the background

The table below describes the procedure for placing an object in the background.

Step	Action
1	Select the object to place in the background. **Result**: The selected object is equipped with handles.
2	Select the **Background** command in the toolbar. **Result**: The selected object is positioned in the background.

How to create compound objects

At a Glance

The objects of a screen are sometimes associated to become a single compound object (that can be animated). Similarly, you may have to unbind a compound object into several simple objects, in order to manipulate them individually.

How to Bind a group of objects

The table below describes the procedures for binding a group of objects.

Step	Action
1	Select the objects to bind. **Result**: The selection is represented by a frame with handles.
2	Activate the **Group** command. **Result**: The objects are grouped in a single **compound object**. the handles belonging to the selected objects disappear and a rectangle containing all of the selected objects and represented by handles is displayed.

How to Unbind a group of objects

The table below describes the operations to perform to unbind a group of objects.

Step	Action
1	Select the compound object to unbind. **Result**: The selected object is represented by handles.
2	Activate the **Ungroup** command. **Result**: All the objects contained in the compound object are viewed with their handles. If the object is animated, the animated characteristics are lost.

How to select an element belonging to a compound object

The table below describes the operations to perform to modify the characteristics of an item belonging to a compound object, without having to ungroup it.

Step	Action
1	Select the compound object. **Result**: The selected object is represented by handles.
2	Press **Alt** then right click on the compound object, while holding down the **Shift** key. **Result**: The elements comprising the compound object are selected successively. The selected object can not be manipulated as a simple object.

Managing the runtime screens messages editor

Subject of this section

This section describes how to manage the runtime screens messages editor.

Operator screens messages editor

At a Glance

Messages comprise a text and a label. They are memorized in the project and are linked to objects via their number. In this way, it is not necessary to create a message per object and duplicate identical messages.

When a project is created, the message list is empty, and the user needs to fill it in: by creating your own messages or by importing messages from another file or another tool.

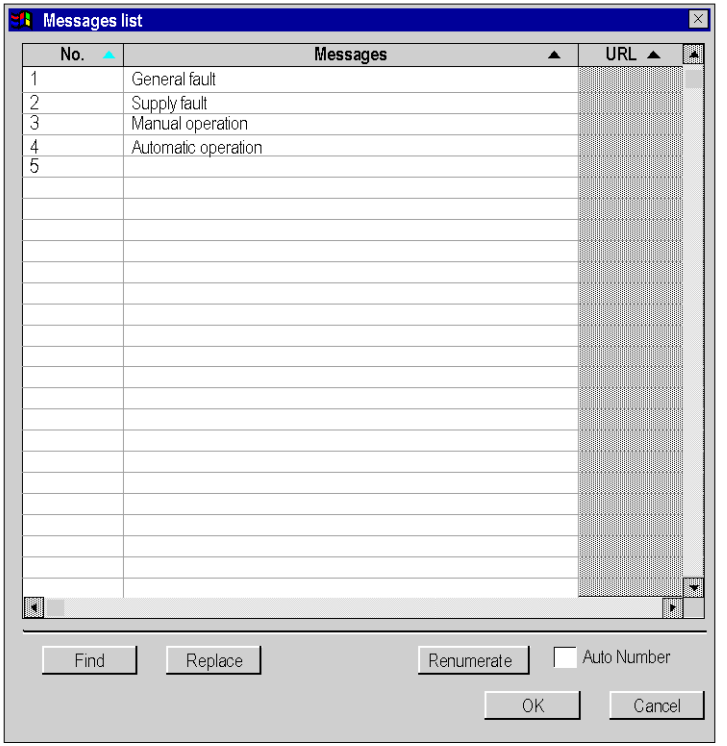
Displaying the message list

The following table describes the operations to perform in order to display the message list.

Step	Action
1	Select the **Operator screens** directory from the project browser.
2	Select the **Message list** > **Open** command in the contextual menu: **Result**: The **Message list** window appears on the screen.

Message screen

The screen below presents an example of a message list:



Description

The following table describes the commands accessible from the message screen.

Command	Description
Find	Allows you to search for a string of characters (or a message number).
Replace	Allows you to replace a string of characters (or a message number).
Auto Number	Allows automatic dialing of messages.
Renumerate	Gives access to a dialog box which allows you to define the first automatic dial number.

How to manage messages linked to operator screens

At a Glance

The dialog box for managing messages allows you to create, modify or delete messages in addition to linking them to a number or a hyperlink.

How to create a message

The table below describes the operations performed to create a message to be used by the operator screen editor.

Step	Action
1	Right click on the **Operator Screens** directory in the project browser. **Result:** The relevant contextual menu appears.
2	Click **Messages list > Open**. **Result:** The messages management dialog box appears.
3	Enter a message number in the **No.** column If you have checked the **Auto number** box, the number will be automatically entered, when the message is validated. **Result:** The number entered appears in the list.
4	Enter your message in the **Messages** column, then press **Enter** to validate. A maximum of 255 characters can be typed in this field. **Result:** Your message and its associated number appear in the list.
5	Repeat steps 3 and 4, to enter the other messages on the list. Click **OK** to confirm the command. **Result:** The messages management dialog box closes and the list is memorized.

How to link a hyperlink to a message

The table below describes the operations performed to link a hyperlink to a message.

Step	Action
1	Open the message list,
2	Right click on the **URL** message column. **Result:** The relevant contextual menu appears.
3	Activate the **Create** command.

Step	Action
	Result: The **Insert hyperlink** window appears.
4	Enter the name with the document path or the WEB page to link to the message. Or select it with the **Browse** button.
5	Click **OK** to validate the hyperlink. **Result:** The hyperlink appears in the **URL** column.
6	Click **OK** to confirm the command. **Result:** The messages management dialog box closes and the list is memorized.

How to modify a message

The following table describes the operations performed to modify an existing message.

Step	Action
1	Right click on the **Operator Screens** directory in the project browser. **Result:** The relevant contextual menu appears.
2	Click **Messages list > Open**. **Result:** The message screen appears.
3	Right click on the message to be modified. **Result**: A cursor appears in the text.
4	Move the cursor to where you want to modify your text. To delete a word, double click on it to select it and press **Del**. Enter the new text or the new number then **Enter** to validate. **Result:** The message is modified.
5	After you complete modifications, click **OK** to validate. **Result**: The new messages are backed up in the project.

NOTE: Press **ESC** to delete all the modifications made in the current session.

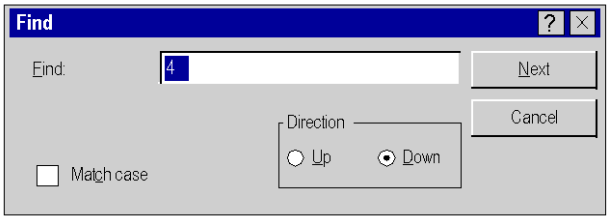
Find dialog box

At a Glance

The **Find** dialog box allows you to find a string of characters or a number in the list of messages.

Dialog box

The screen below allows users to set the find parameters.



Description

The following table describes the different fields offered by the dialog box.

Field	Description
Find	Allows you to enter the string of characters (or number) you want to find.
Match case	If this box is checked, find will match the case of the entered characters (upper or lower case).
Direction	This field allows you to define the search direction in the list: move up or move down.
Next	This command initializes find or starts the next find.
Cancel	This command cancels the search.

How to find and replace a string of characters or a number.

Follow the steps below to find or replace a string of characters (or a number) in the list of messages:

Step	Action
1	Right click on the **Operator Screens** directory in the project browser. **Result**: The relevant contextual menu appears
2	Click **Messages list > Open**. **Result**: The message manager dialog box appears.
3	Enable the **Find** command to find a string of characters or **Replace** to find and replace a string of characters. **Result**: A **Find** dialog box appears
4	Fill in the different fields proposed by the dialog box and start the find or replace, page 1629 of the string of characters. **Result:** The changes are integrated into the list of messages.
5	After you complete modifications, click **OK** to validate. **Result**: The new messages are backed up in the project.

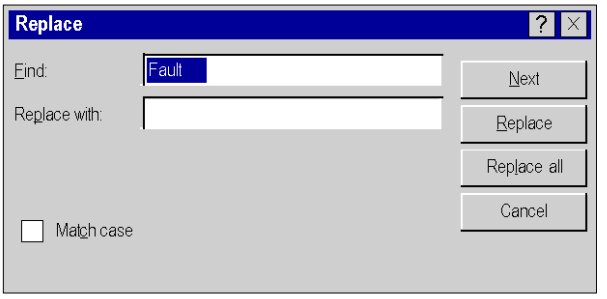
Replace dialog box

At a Glance

Dialog box **Replace** allows you to find a string of characters (or a number) in the list of messages and to replace it with another string of characters (or a number). You may select the item to replace or replace all.

Dialog box

The screen below allows users define the replace parameters.



Description

The following table describes the different fields offered by the dialog box.

Field	Description
Find	Allows you to enter the string of characters (or number) you want to find.
Replace	This field allows you to enter the string of characters (or number) you want to replace.
Match case	If this box is checked, find will match the case of the entered characters (upper or lower case).
Next	This command initializes find or starts the next find.
Replace	This command replaces the string of characters found with the new string of characters that you defined in the **Replace with** field.
Replace all	This command replaces the string of characters that you defined throughout the list of messages.
Cancel	This command cancels the find/replace command.

Message Renumbering Dialog Box

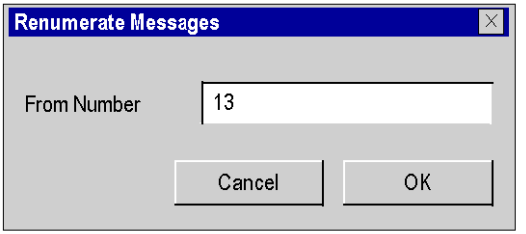
At a Glance

The renumbering dialog box allows you to change a number of one or several messages in the list automatically.

How to Modify a Message Number

To modify a message number, you can proceed in the same way as when modifying a message or use the **Renumerate** command. The table below describes the procedure for modifying a message number using the **Renumerate** command.

Step	Action
1	Right click on the **Operator Screens** folder in the project browser. **Result**: The relevant contextual menu appears.
2	Execute the **Messages list > Open** command. **Result**: The message manager dialog box appears.
3	Left click in the number field of the message to be modified. **Result**: The field is selected.

Step	Action
4	Activate the **Renumerate** command. **Result:** The following dialog box appears: 
5	Enter the new number and confirm with **Enter**. **Result**: The dialog box disappears and the selected message will be renumbered from the number entered in the **From Number** field.
6	After you complete modifications, click **OK** to validate. **Result**: The new numbers are backed up in the project.

Message editing and import/export function

At a Glance

The operator screen messages also have edit and import/export functions.

Information on editing functions

The editing functions are accessed by the commands:

- **Edit > Cut**
- **Edit > Copy**
- **Edit > Paste**
- **Edit > Delete**

NOTE: The paste function also allows messages copied in tools such as Excel or Word to be pasted directly into the message screen.

They are also accessible via the contextual menu: right click on the message.

There are also other functions in the contextual menu from the messages list.

Function	Description
Cancel	Cancels the last completed action.
Redo	Repeats the last completed action.
Clear message	Erases the content of the selected row.
Insert row	Inserts a blank row above the selection:
Delete row	Deletes the selected row.

You can **sort** the messages list by:

- the message number (increasing or decreasing order),
- alphabetical order (increasing or decreasing) of messages,
- alphabetical order (increasing or decreasing) of hyperlinks.

To start sorting, just click on the heading of the column you want to sort.

How to import a list of messages

The table below describes the procedure for importing a list of messages.

Step	Action
1	Right click on the **Operator Screens** directory in the project browser. **Result**: The relevant contextual menu appears.
2	Execute the **Messages list > Import** command. **Result**: A dialog box is displayed asking you to choose the file to be imported.
3	Select the file to be imported (directory tree and file) then confirm with the **Import** command. **Result:** Another dialog box is displayed asking you to choose the type of import to be made.
4	Check the **Replace**[1] box, or Check the **Merge (Based on new messages)**[2] box, or Check the **Merge (Based on new numbers)**[3] box.
5	Click **OK** to confirm the modifications to the list. **Result**: The new messages are backed up in the project.

Note[1]: All currently defined messages in the project are deleted and then the new messages are imported.

Note[2]: The currently defined messages in the project are kept. When one of the new imported messages has the same message text as a message already existing in the

project, this message is not imported. All the other imported messages are added at the end of the current message list of the project.

Note[3]: The currently defined messages in the project are kept. When one of the new imported messages has the same message number as a message already existing in the project, this message is not imported. All the other imported messages are added at the end of the current message list of the project.

NOTE: To import a list of messages always use the

- **Operator Screens > Messages list > Import** command

and **NOT** the

- **Operator Screens > Import** command.

Otherwise the dialog box for selecting the **Import type** will not be displayed (see step 3 and 4 in the table above).

How to export a list of messages

The table below describes the procedure for exporting a list of messages.

Step	Action
1	Right click on the **Operator Screens** directory in the project browser. **Result**: The relevant contextual menu appears.
2	Execute the **Messages list > Export** command. **Result**: A dialog box appears to allow you to specify the name of the export file.
3	Choose the directory (directory tree) and enter the name of the export file (the extension.XCR is automatically added). Click on the **Export** command to confirm. **Result:** Messages are exported in a text file, each line containing the number and label of each message. This file can then be used with any text editor.

Runtime screens in online mode

What's in This Chapter

Basic functions in online mode 1635
How to select an animated object 1636
How to use command objects to modify the variables associated with objects 1637
How to view the variables associated with graphic objects 1638
How to modify the variables associated with objects from an animation table 1639
How to search for cross-references 1640
How to manage the display of operator screens 1641
How to modify screen control 1643
How to obtain information on objects when in connected mode 1644

Object of this chapter

This chapter describes how the runtime screens editor works in online mode.

Basic functions in online mode

Introduction

After you create your operator screens, you must change to online mode, in order to monitor and control the automation process.

To do this, enable the relevant icon or **PLC > Connect** command

Editing online

A screen can only be edited in online mode if you have not given priority, page 602 to the **PLC driver**.

Display rules for a simple object

Animated objects which make up a screen are displayed when their display condition has been met.

Display rules for a compound object

When a compound object is animated but its display conditions have not been met, none of its component items are displayed (even if it is made up of items whose display requirements have been met).

Basic points about display of related objects

The objects on a page are displayed according to the positioning order assigned to them by the **Foreground** or **Background** commands.

NOTE: Command objects are always positioned above other graphic objects.

To avoid control errors or display problems, we advise you do not overlay command objects.

How to select an animated object

At a Glance

You may need to select an object in online mode. We advise you to follow the procedure described below for the process to remain unchanged during the selection.

How to select an animated object

The table below describes the procedure for selecting an object in online mode

If you want to select	then
an animated object using the function keys.	press **F4** or **F5**. **Result**: The **F4** key selects animated objects one after the other in the order that they were created or repositioned by the user. The **F5** key selects them in the reverse order.
an animated object using the tool bar.	click on the **next animated object** or **previous animated object** icons. **Result**: It is identical to pressing the **F4** and **F5** keys.
a control command object.	press and hold down the **Ctrl** key, then left click on the object. **Result**: The object is selected. It no longer commands the control and its properties can be modified.
any object.	press the **Space** bar. **Result**: The **Space** bar selects all the objects one after the other according to how they were created or repositioned. Simultaneously press **Caps Lock + Space** to select them in the reverse order.

How to use command objects to modify the variables associated with objects

At a Glance

In online mode, you may use the command objects in an operator screen to modify the variables associated with the animated objects. By default, you cannot modify the command objects when a screen is being opened. You must use the **F7** key or the **Services > Enable Variable Modification** command to allow these modifications.

How to use a command object to modify variables

The following table describes the procedures for using a command object to modify variables.

Step	Action
1	Open the operator screen in which the command objects are located.
2	Press the **F7** key or select the **Services > Enable Variable Modification** command.

Step	Action
3	Give control to the command object that enables the modification of the variable settings.
4	Modify the command object settings.

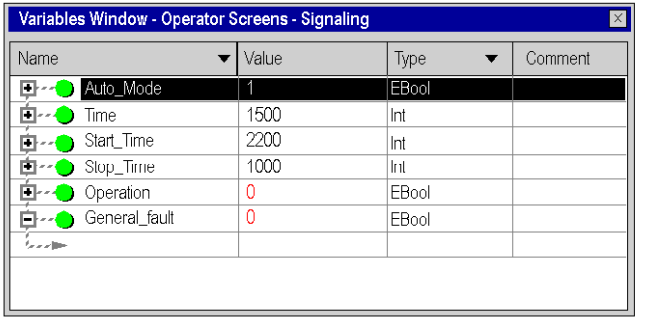
How to view the variables associated with graphic objects

At a Glance

In online mode you may view the settings of variables associated with an operator screen in the **Variables Window** tool. The **Variables Window** tool only allows you to view the variables. You cannot edit them.

How to view the variables associated with graphic objects

The following table describes the operations necessary to display the variables associated with the graphic objects.

Step	Action
1	Select the operator screen.
2	Select the **Tools > Variables Window** where the variables you want to view are located. **Result**: The **Variables Window** appears 
3	Double click on a variable in the **Variables Window**. **Result**: The object associated with this variable appears in the operator screen, surrounded by a dotted frame.

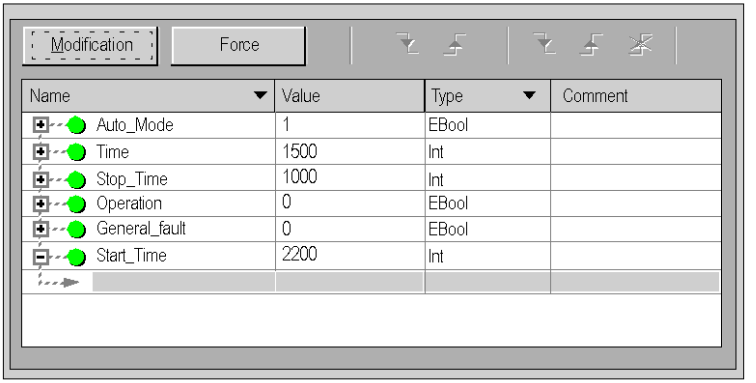
How to modify the variables associated with objects from an animation table

At a Glance

In online mode you may modify the variables associated with the animated objects from an animation table.

How to modify the variables associated with objects from an animation table

The table below describes how to modify the variables associated with graphic objects from an animation table.

Step	Action
1	Select the objects for which you want to modify the variables. **Result**: The selected objects appear with handles and have a dotted frame.
2	Use the **Ctrl** + **T** shortcut key or enable the **Services** > **Initialize Animation Table** command. **Result**: All the variables relating to the selected objects appear in a new animation table. You can now modify their value. The diagram below presents an example of an animation table produced by selecting animated objects. 

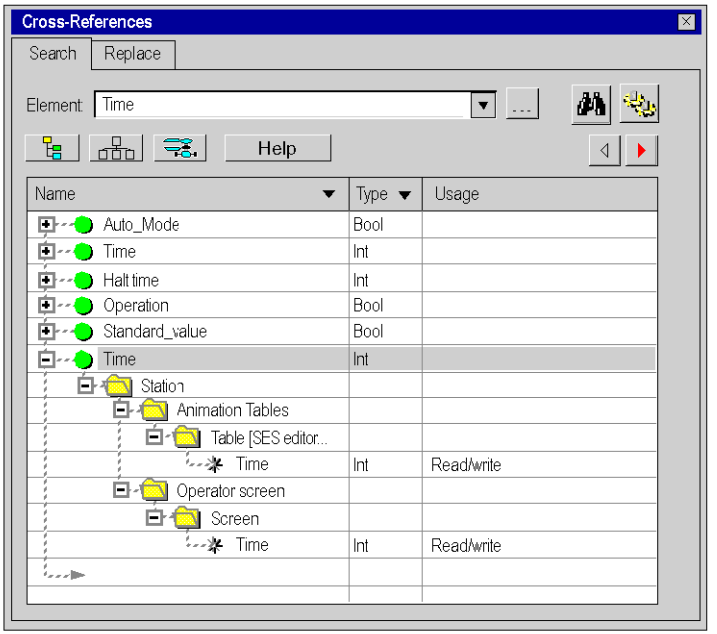
How to search for cross-references

At a Glance

In connected mode as in local mode, you can search in the project programs for the variables linked to an animated object. This is known as the cross references search.

How to search for cross references

The table below describes the procedures for searching for cross references in an animated object.

Step	Action
1	Select the object(s) in the operator screen. **Result**: The selected objects appear with handles and have a dotted frame.
2	Use the **Ctrl** + **U** shortcut or click **Services** > **Initialize Search**. **Result**: All the variables relating to the selected objects appear in the **cross references** search window. The diagram below is an example of a cross reference search performed by selecting the animated objects. 

How to manage the display of operator screens

At a Glance

You may need to modify the display of screens in connected mode. You may for example:

- Mask the horizontal and vertical scrollbars,
- Go to **full screen**mode.

How to mask the scroll bars

The table below describes the procedure for masking scrollbars displayed by the operator screens editor.

Step	Action
1	Click **Tools** > **Options**.
2	Select the **operator screen** tab. **Result**: The editor configuration, page 602 dialog box is displayed.
3	Uncheck the **Horizontal scroll bar** and **Vertical scroll bar** and validate with **OK**. **Result**: The graphic editor no longer has vertical and horizontal scrollbars.

How to switch to full screen mode

The table below describes the procedure for switching to **full screen** mode.

Step	Action
1	Click **Display** > **Full screen**. **Result**: The menu and toolbars disappear.

How to exit a full screen mode.

The table below describes the procedure for switching to full screen mode.

Step	Action
1	Press **Esc** to exit the full screen mode. **Result**: The screen changes from **full screen** mode to **window** mode.

How to start the operator screens editor automatically while in full screen mode

The table below describes the procedure for automatically starting the operator screens editor in full screen.

Step	Action
1	Click **Tools > Project settings**.
2	Select the **operator screen** tab. **Result**: The editor configuration, page 591 dialog box is displayed.
3	Check the **Full screen at opening** box and validate with **OK**. **Result**: The operator screens editor will be automatically started in full screen mode.

How to modify screen control

At a Glance

The control type is determined by the configuration, page 591 options. It can be modified in online mode.

NOTE: When the runtime screens are controlled via PLC it is not possible to edit them in online mode, even if the option **Enable Edit in online mode** is set via **Tools > Options > Operator Screens** tab.

How to set the control type in online mode

The following table describes the procedure for changing the control type for operator screens.

If you want	and if you are	then
the screens to be controlled by the PLC,	in operator control mode,	press **F2** or click on the **PLC controlled** icon.
the screens to be controlled by the operator,	in PLC control mode,	press **F3** or click on the **Operator controlled** icon.

How to obtain information on objects when in connected mode

Introduction

In connected mode, you can obtain two types of information via the mouse:

- Is the object a driving object,
- which variable is associated to this object.

How to ascertain whether an object is a driving object

The table below describes the procedure for ascertaining whether an object is a driving object.

Step	Action
1	Position the mouse cursor on the object. **Result:** if the object is a driving object, the cursor takes the shape of a hand.

How to find out the name and the value of a variable associated with an object

The table describes the procedure for finding out the name and value of a variable associated with an object.

Step	Action
1	Position the mouse cursor on the object. **Result:** the name of the variable and its current value in the PLC are displayed.

Runtime screens objects library

What's in This Chapter

Managing library objects..... 1646

Object of this chapter

You can access an objects library to create your runtime screens. This chapter presents a family-specific classification of all these objects: actuators, display panels, automation components, etc..

Managing library objects

Subject of this section

This section describes how to manage the library graphic objects.

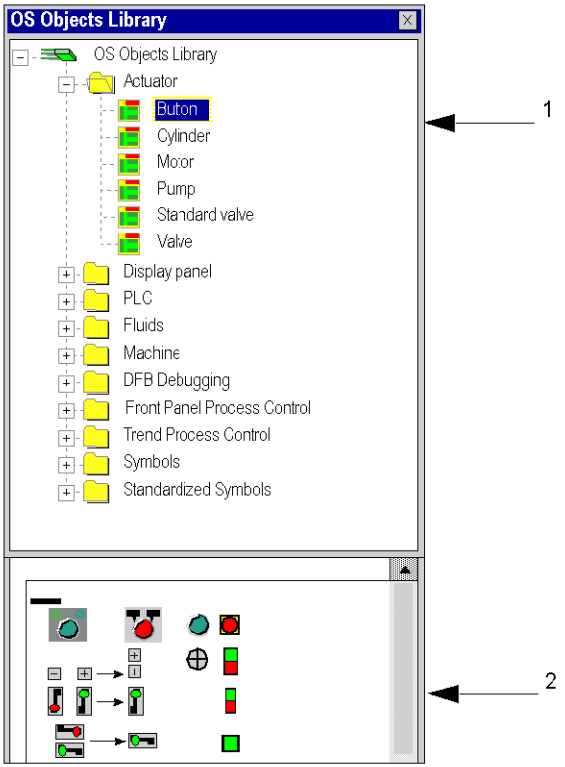
The Objects Library

At a Glance

The objects library provides manufacturer objects which can be inserted into the operator screens. The objects are arranged in families. The library allows operators to create their own objects and insert them into a library family.

The library may be opened with the **Tools > Operator Screen Library** command.

The figure below presents the objects library.



The table below provides a description of the objects library.

Number	Description
1	The browser displays all the families and objects.
2	The window displays the graphic representation(s) of the selected object.

How to manage the list of graphic items in the library

At a Glance

The library graphical objects are organized in families for optimum management. You cannot create new objects outside a family.

The main actions that can be performed on objects or object families are described below.

How to create an object family in the library

The following table describes the steps to be carried out to create a new object family in the library.

Step	Action
1	Execute the **Tools > Operator screens library** from the main Control Expert menu **Result:** The object library opens. The browser displays the objects and the families contained in the library.
2	Right click on the library root. **Result:** The relevant contextual menu appears.
3	Enable the command **New family**. **Result:** A new screen family appears in the objects library with a generic name.

How to create an object in the library

The following table describes the operations to be performed in order to create an object in the library.

Step	Action
1	Click **Tools > Operator screens library** in the toolbar. **Result:** The object library opens. The browser displays the objects and the families contained in the library.
2	Click with the right mouse button on an object family or library object. **Result:** The relevant contextual menu appears.
3	Enable the **New object** command. **Result:** A new object appears in the objects library with a generic name.

How to rename a library object or object family

The following table describes the operations to be performed in order to rename a library object family.

Step	Action
1	Click **Tools > Operator screens library** in the toolbar. **Result:** The object library opens. The browser displays the objects and the families contained in the library.
2	Click with the left mouse button on an object family or library object.
3	Press **F2**. **Result:** The label is highlighted and the text input cursor flashes at the end of the input field.
4	Enter the new name of the object or family using the keyboard. **Result:** The object or object family appears with its new name.

How to move an object in the library

The following table describes the operations to be performed in order to move an object from a family to another family in the library.

Step	Action
1	Click **Tools > Operator screens library** in the toolbar. **Result:** The object library opens. The browser displays the objects and the families contained in the library.
2	Left click on the library object to be moved. **Result:** The relevant contextual menu appears.
3	Hold down the left mouse button to maintain the selection.
4	Move the object to the desired family while holding down the button. **Result:** The object moves at the same time as the mouse cursor.
5	Release the mouse button. **Result:** The object appears in the desired family.

NOTE: You may move an object in the same family if the Manual sort, page 1650 option is checked.

How to open an object in the library

The following table describes the operations to be performed in order to open an object in the library.

Step	Action
1	Click **Tools > Operator screens library** in the toolbar. **Result:** The object library opens. The browser displays the objects and the families contained in the library.
2	Right click on a library object. **Result:** The relevant contextual menu appears.
3	Activate the **Open** command. **Result:** An operator screen window opens. It displays the graphic object or objects contained in the library object.

Information on editing functions

The editing functions are accessed by the commands:

- **Edit > Cut**
- **Edit > Copy**
- **Edit > Paste**
- **Edit > Delete**

Commands comply with Windows 32 bit ergonomics and apply to objects or families of objects in the library.

There is also a function for deleting the last completed action.

Properties of the object library and its elements

At a Glance

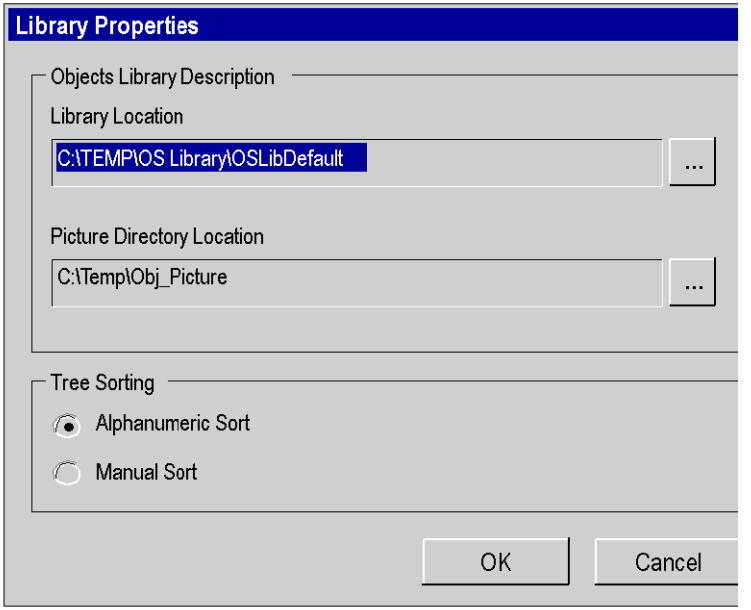
The objects library and its elements (family and objects) have properties. They allow the operator to modify:

- the access pathname of the library file,
- the access pathname of the library images directory,
- the type of family and object sorting in the library browser,
- the type of family and object sorting in the library browser,
- the name of families,
- the name of objects,

- the family linked to the object.

Library properties

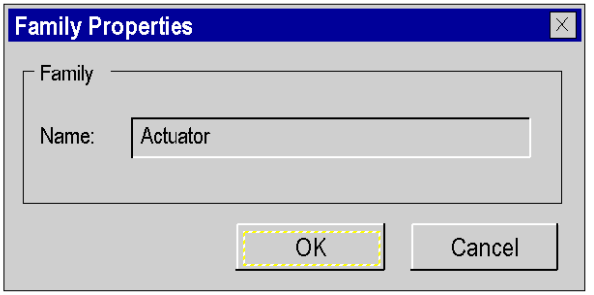
The table below presents the procedures for accessing the **Library Properties** dialog box.

Step	Action
1	Execute the **Tools > Operator screens library** from the main Control Expert menu. **Result:** The object library opens. The browser displays the objects and the families contained in the library.
2	Select the library root in the browser application.
3	Select the **Properties** command from the contextual menu. **Result:** The **Library Properties** window appears: 
4	The access pathname and the filename (.bib extension) of the library are indicated in the **Library Location** field. The  button allows the operator to browse the hard disk to change the pathname and the file.

Step	Action
5	The **Picture Directory Location** field contains the access pathname and file name (extension .bib) of the directory where the images used by the library are located. The  button allows the operator to browse the hard disk to change the pathname and the storage directory name.
6	In the **Tree Sorting** zone you may choose to sort the families and objects. Sorting can be: • in alphabetical order, by checking the **Alphanumeric Sort** box, • manual (user option), by checking the **Manual Sort** box.

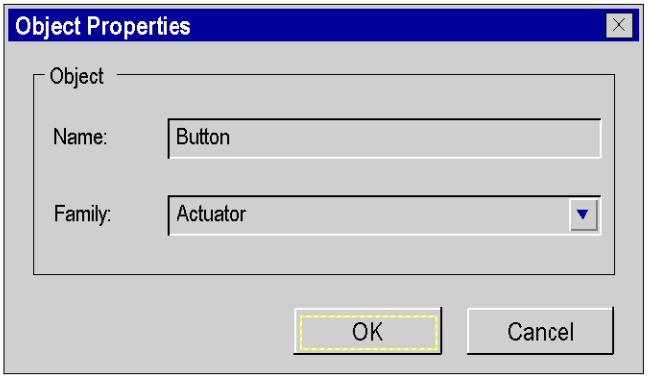
Library properties

The table below presents the procedures for accessing the **Family Properties** dialog box.

Step	Action
1	Execute the **Tools > Object Library** command from the Control Expert main menu. **Result:** The object library opens. The browser displays the objects and the families contained in the library.
2	Select a family in the library browser application.
3	Select the **Properties** command from the contextual menu. **Result:** The **Family Properties** window appears: 
4	In the **Name** field, give the family a name. It can be modified and can contain up to 255 characters.

Object properties

The table below presents the procedures for accessing the **Object Properties** dialog box.

Step	Action
1	Execute the **Tools > Object Library** command from the Control Expert main menu. **Result:** The object library opens. The browser displays the objects and the families contained in the library.
2	Select an object in the library browser application.
3	Select the **Properties** command from the contextual menu. **Result:** The **Object Properties** window appears. 
4	In the **Name** field, give the object a name. It can be modified and can contain up to 255 characters.
5	The **Family** field displays the family with which the object is associated. Select the arrow on the right side of the field to associate the object with another family.

Handling the Objects

At a Glance

When you open a library object, Control Expert opens an operator screen window in edit mode. The operator screen contains the different graphic objects. From the operator screen, you can

- use all editing functions in the screen
- select and copy a graphic object in another screen
- bind variables to graphic objects

Edit function

An object's operator screen editor works in the same way as the project operator screen editor. It has the same functions: The menu and toolbars, page 1559 are identical.

Copying an object

You can copy graphic objects between the operator screen of an object and:

- another object operator screen,
- a project operator screen.

In both cases the procedures are identical.

Procedure

The table below describes the procedure for copying a graphic object in another library object or in a project operator screen.

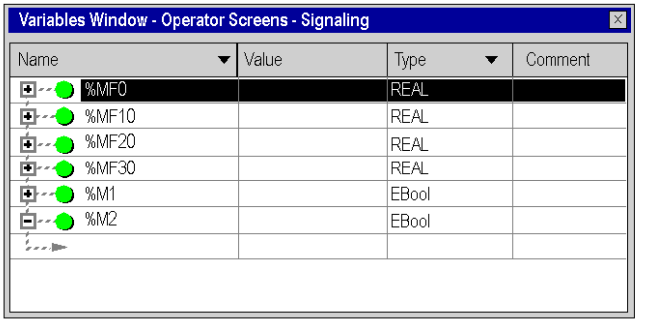
Step	Action
1	Open, page 1649 a library object. **Result:** The operator screen with the graphic objects appears.
2	Right click on a graphic object. **Result:** The relevant contextual menu appears.
3	Enable the **Copy** command.
4	Open a library object or an operator screen in which you want to copy the graphic object.
5	Right mouse click on the background of the operator screen.
6	Activate the **Paste** command. **Result:** The graphic object appears in the operator screen.

Association of variable

Graphic objects are animated with variables that have topological addresses. Simply replace these variables with your project variables to animate the objects.

Procedure

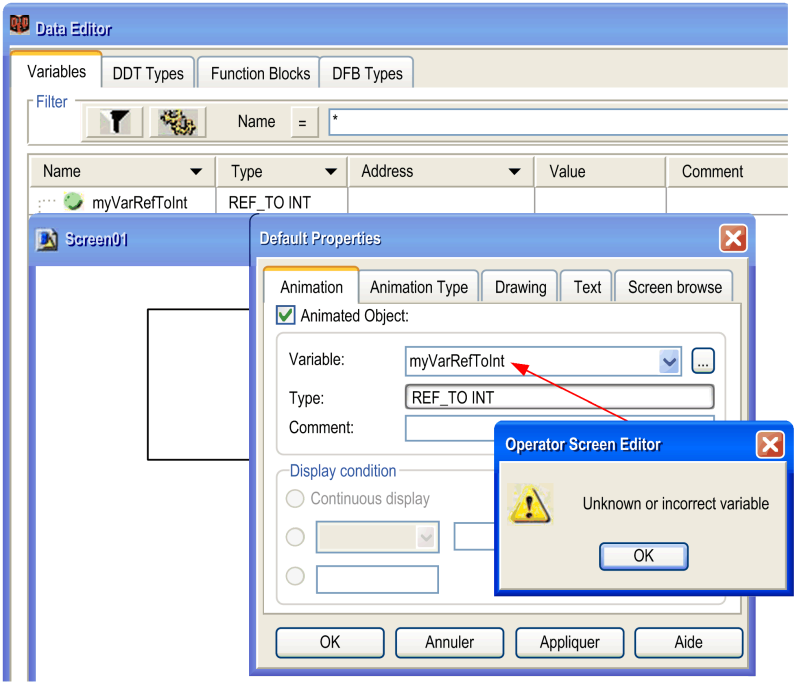
The following table describes the procedure for associating project variables with animation variables for graphic objects.

Step	Action
1	Copy a graphic object from the library into an operator screen, then select it.
2	Select the **Tools > Variables Window** command. **Result**: The **Variables Window** appears with all the variables of the graphic object 
3	Double click on a variable in the **Variables Window**. **Result**: The object associated with this variable appears in the operator screen, surrounded by a dotted frame.
4	Select the framed object.
5	Select the **Properties** command from the contextual menu. **Result**: The **Object Properties** window appears.
6	Select the **Animation** tab. **Result**: The variable that you selected initially appears in the **Variables** field.
7	Click on  to open the variables selection tool.
8	Select a project variable and click **OK**. **Result**: The selected variable appears in the **Variables** field in the **Animation** tab and in the variables window.
9	For each of the project variables to be grouped, double-click on the variable in the variables window and repeat steps 7 and 8.

Dereferencing in Operator Screens

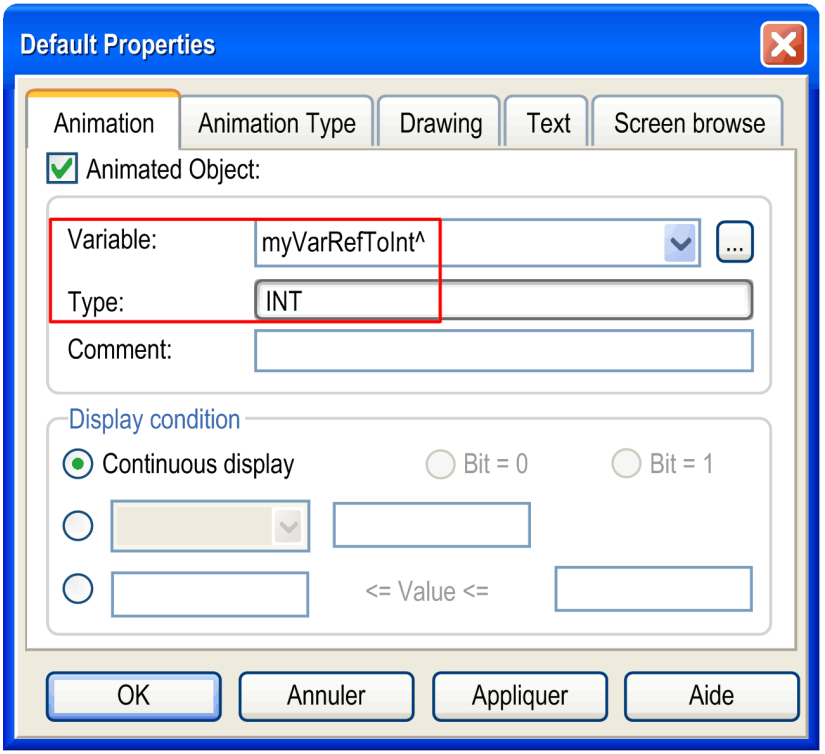
References

References, with a REF_TO ANY type, cannot be entered in an operator screen:



Dereferences

A dereferenced pointer (with the dereferencing operator “^”) can be entered; this is the actual variable itself:



NOTE: The dereferenced variable must have an Elementary Data Type.

NOTE: Only 1 level of dereferencing is allowed. This is not allowed: Myref^.MyInt^

Documentation, Import/export

What's in This Part

Documentation 1660
Import / Export 1686

Object of this part

This part presents the documentation and import/export functions.

Documentation

What's in This Chapter

Creating Project Documentation..... 1660
Documentation Structure..... 1661
Topics in Documentation 1663
Creating Project Documentation..... 1672
Print Preview..... 1675
Page Setup 1677
Print..... 1678
Printout Setup..... 1681

Overview

This chapter describes how project documentation is created.

Creating Project Documentation

Introduction

Control Expert lets you create the documentation for your project.

The subjects for the documentation are displayed in a tree structure. The structure contains all the project browser elements that were used. Elements that you have not used in the project do not appear as subjects in the documentation. The structure also contains some subjects specific to printing (for example the page title and table of contents).

In the tree structure, you can choose which subjects you want to print or display (print preview) and which you do not.

For some subject, parameters are available that you can use to define the printout.

For the printout you can choose between the **Structural View** or the **Functional View**.

You can print this documentation partially or fully after it has been created.

Displaying the documentation structure

To display the documentation structure in the project browser you can either double-click **Documentation** directory, or right-click the **Documentation** directory and click **Open**.

Documentation Structure

General Information

The documentation structure, page 1673 can be displayed with one of the two following views:

- Structural View, page 1662
- Functional view, page 1663

Selecting the Type of Documentation Display

The following options are available to display the documentation structure in the structure view:

- Select the symbol in the tools list in the documentation structure.

- Select the subject in the documentation structure and execute the menu command **View > Structural View**.

The following options are available to display the documentation structure in the functional view:

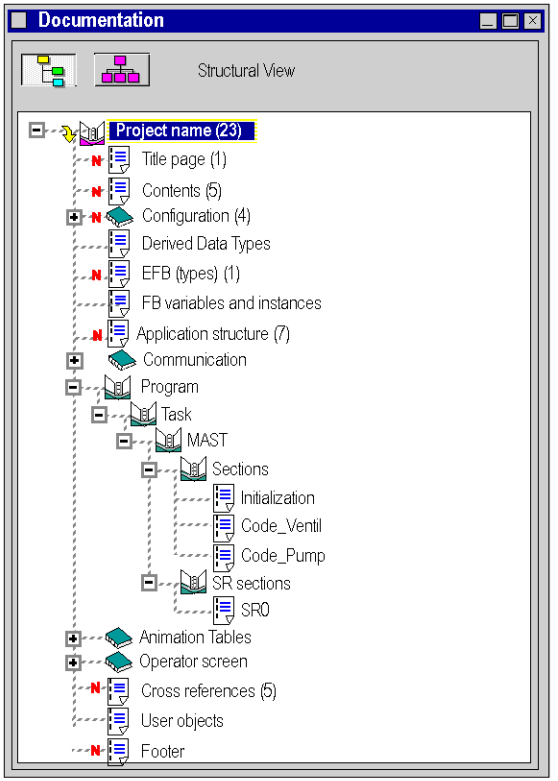
- Select the symbol in the tools list in the documentation structure.

- Select the subject in the documentation structure and execute the menu command **View > Functional View**.

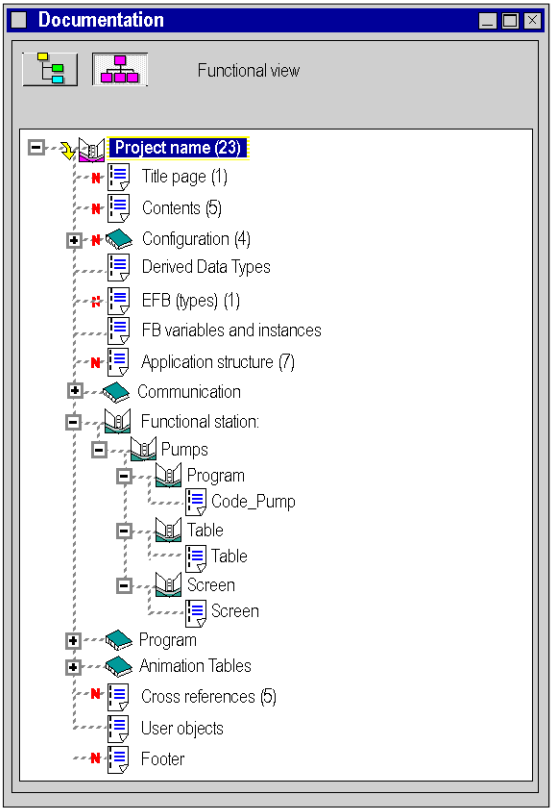
Structural View

The documentation structural view displays the project structure in the form of objects. All objects are connected to their headings. That means that the sections and events are in the header program and the animation tables and user windows are in the header animation table or operator windows.



Functional view

The documentation functional view displays the project structure in the form of functional units. Each functional unit contains the program, the animation table and the operator screens assigned to them. The objects that belong to a unit remain assigned to the appropriate headers in the structural view.



Topics in Documentation

Documentation Topics

The documentation folder of a project can contain the following topics:

- Title page

- Contents
- General information
- Configuration
- Derived Data Types
- Derived FB Types
- EFB types
- EF types
- FB variables and instances
- Application structure
- Functional project
- Communication
- Programs
- Animation table
- Operator screen
- Cross-References
- Footer

  NOTE: Using the menu commands **Edit > Include heading** and **Edit > Exclude heading** you can define the topics you want to use in your documentation, page 1672.

Title Page

This topic deals with printing the title page.

Topic contents:

- Name of the project
- Name of the designer
- Software version
- Creation date
- Date of last change
- Target PLC

  NOTE: In the **Title Page** area of **Printout Setup** dialog box you can define the name of the project and the designer. All other details are automatically generated.

Contents

The contents page is generated automatically according to the topics selected.

When a topic is selected, its corresponding number of pages appears in the contents. If the topic contains documentation but is not selected, **No print job** is displayed instead of the number of pages.

The total number of pages is displayed at the end of the contents.

General Information

This topic deals with printing user content entered in **Documentation > General Information** tab of the project browser.

This information can be comments on the project, hyper-links and/or bitmaps.

Configuration

This topic deals with printing the hardware configuration.

Topic contents:

- Bus topology
- Bus parameters (except for the X bus)
- PLC and rack configuration
- Parameterization of the input/output modules and the communication modules.

Derived Data Types

This topic deals with printing the derived datatypes for the project.

Topic contents:

- Names of the data structure and their elements
- Data structure types (for example, structure, array) and element datatypes
- Data structure comments

Derived FB Types

This topic deals with printing the DFBs used in the project.

Every DFB has a structure which contains the section of the DFB.

Topic contents:

- Topic **Derived FB types**
  - Table of all DFBs used in the project and their properties (name, version, creation / change date)
- Topic **"DFB name"**
  - Type name
  - Version
  - Short description of how the DFB work (if available)
  - Properties of the inputs, outputs, in/outputs and public variables (name, data type, initial value and comment)
- Topic **Sections**
  - Name of the section and the DFB
  - Section comment
  - Type of protection
  - Assigned functional module
  - Name of the condition for conditional call
  - Section code (except for the diagnostics DFBs provided by Schneider).

    Details on the section documentation can be found in the topic **Programs**.

EFB (Types)

This topic deals with printing the Elementary Function Block types used in the project.

Topic contents:

- Table of all EFBs used in the project and their properties (name, version, creation / change date)

- For every EFB type
  - Type name
  - Version
  - Short description of the how the EFB works
  - Properties of the inputs, outputs, in/outputs and public variables (name, data type, initial value and comment)

FB (types)

This topic deals with printing the Elementary Function types (FBs) used in the project.

Topic contents:

- Table of all FBs used in the project and their properties (name, version, creation / change date)
- For every FB type
  - Type name
  - Properties of the inputs, outputs, in/outputs and public variables (name, data type, initial value and comment)

Variables & FB instances

This topic deals with printing the following objects in a tabular format:

- Elementary datatypes (EDTs)
- Derived Data Types (DDTs)
- I/O derived data types (IODDTs)
- Instances of elementary functions
- Instances of elementary functions blocks
- Instances of derived functions blocks

Topic contents:

- Symbolic name (variable name or FB instance name)
- Constants (only for variables)
- Address (only for variables)
- Initial value
- Number of connections

- Global file (only for elementary datatypes and FB instances)
- Which attributes, for example, alias of, diag and so on (only if defined)

  NOTE: In the **Variables & FB-Instances** area of **Printout Setup** dialog box you can define the attributes printed for the variables and FB instances.

Application Structure

This topic deals with printing the project structure.

Topic contents:

- List of the Program Units and/or program sections:
  - Release conditions
  - Comments
  - Module
  - Language type
- Sub-programs and macro steps calling tree

Depending on the view selected the documentation is shown in the **Structural View** or the **Functional View**. If the **Functional View** is chosen the section documentation is sorted according to the assigned functional module. The documentation of sections not assigned to any functional module is shown in its own table.

Functional Project

This topic is only available in the **Functional View** and is used to print the sections assigned to a functional module.

Sections not assigned to documentation are not documented under the topic **Programs**.

Communication

This topic deals with printing the networks communication parameters.

Programs

This topic deals with printing the program elements.

Topic contents:

- MAST task
  - Properties (for example, configuration, configuration task duration, watchdog time) in the MAST task.
  - Program Units of the MAST task (**Interface & Variables**, sections)
  - Sections of the MAST task
  - Subroutine section of the MAST task
- FAST task
  - Properties (for example, configuration, configuration task duration, watchdog time) in the FAST task.
  - Program Units of the FAST task (**Interface & Variables**, sections)
  - Sections of the FAST task
  - Subroutine section of the FAST task
- AUX0 and AUX1 tasks when available
- Events
  - Sections of the **Timer Events** tasks
  - Sections of the I/O event tasks

The contents of the section documentation depend on their programming languages.

All sections:

- Name of the sections and the assigned tasks
- Section comment
- Type of protection
- Assigned functional module
- Name of the condition for conditional call
- Section code

FBD sections:

- Component tables with **Execution by** attribute.

  Blocks that have an **Execute by** reference to another block appear in a table that shows the instance names and the position of the referenced and referencing blocks.

LD sections:

- Table for all of the object names not fully shown.

  Because of the fixed column width it can happen that variable names or block types are not shown fully. The incomplete object names appear in a table that shows the full object names and their position(s).

- Using the checkbox **Mixed Display Mode** in the **Printout Setup** dialog, you can define whether the documentation in the LD section is displayed in Input mode, page 754 or in mixed display mode, page 755.

SFC sections:

- Tables for the step / action properties:
  - Step name
  - Position of the step object
  - Minimum and maximum supervision time
  - Delay Time
  - Step comment
  - Identifiers for the action(s)
  - Time literal / variable for the identifier (if required)
  - Action name
- Transition properties table:
  - Name of transition
  - Type of transition condition
  - Position of the transition object
  - Comment
- Jump properties table:
  - Target step name
  - Position of the jump object
  - Comment

Animation Tables

This topic deals with printing the animation tables.

Topic contents:

- Name of the animation table
- Comment
- Functional module
- Name of the object (for example, variable name, EFB/DFB/DDT instance name, address)
- Type
- Comment

Operator Screen

This topic deals with printing the operator screens.

This heading is used to print the properties of the operator screens, their graphic representations and the lists of messages associated with each screen.

Topic contents:

- Message lists
- Operator screen
  - Properties
  - Graphical representation
  - List of the animated variables

  NOTE: In the **Operator Screens** area of **Printout Setup** dialog box you can define the scope and display of the documentation and the variables and FB instances.

Cross-References

This topic deals with printing the cross-references for the application and the DFBs used.

Every cross-reference can contain the following tables:

- Addresses
- FB variables and instances
- FB objects
- Subroutine

Every table contains the following attributes:

- Name of the object
- Location of the reference (for example, section name, module name)
- Position used (for example, line and column in the section, type of module)

User Objects

This topic deals with printing the properties of the hyperlinks and the user directories.

Contents of the topic **Hyperlinks**:

- Nodes containing hyperlinks (for example, station\animation tables)
- Comment

- Target path

Contents of the topic **User directories**:

- Nodes containing hyperlinks (for example, station\animation tables)
- Comment

Footer

This topic deals with printing the footer.

Topic contents:

- Bitmap (logo)
- Author
- Dept.
- Target PLC
- Page title
- Two spare fields
- The date of printing
- Page number

NOTE: The target PLC, the page title, the printing date and the page number are automatically generated. In the **Footer** area of the **Printout Setup** dialog box you can define other attributes.

Creating Project Documentation

Creating Project Documentation

Carry out the following steps to create project documentation:

Step	Action
1	Open the documentation structure., page 1673
2	Select the documentation view, page 1673.
3	Select the individual subjects that you want to include or exclude (**Include heading** or **Exclude heading**).
4	Set up the page for printing, page 1677.

Step	Action
5	Define the parameters for the printout, page 1681.
6	Prepare the subjects for printing, page 1674.
7	Check the settings with the print preview, page 1675.
8	Select a printer and start printing, page 1678.

Opening the Documentation Structure

To open the documentation structure choose one of the two following options:

- Double-click **Documentation** directory in the project browser.
- Right-click the **Documentation** directory and click **Open**.

Selecting the Documentation View

After opening the documentation structure you can choose between the Structural View, page 1662 and the Functional View, page 1663.

Including a Subject in the Documentation

You have the following options to include a subject in the documentation:

- In the documentation structure, right-click the subject to include and click **Include heading**.
- In the documentation structure, select the subject to include and execute the menu command **Edit > Include heading**.
- In the documentation structure, click beside the subject symbol to toggle between **Include heading** and **Exclude heading**.

Subjects included in the documentation are marked with a red symbol in front of the subject.

Excluding a Subject From the Documentation

You have the following options to exclude a subject from the documentation:

- In the documentation structure, right-click the subject to exclude and click **Exclude heading**.

- In the documentation structure, select the subject to exclude and execute the menu command **Edit > Exclude heading**.
- In the documentation structure, click beside the subject symbol to toggle between **Include heading** and **Exclude heading**.

Adding all Subjects in the Documentation

You have the following options to include all the subjects in the documentation:

- In the documentation structure, right-click the project and click **Include all headings**.
- In the documentation structure, select the project and execute the menu command **Edit > Include all headings**.
- In the documentation structure, click beside the project symbol to toggle between **Include all headings** and **Exclude all headings**.

Subjects included in the documentation are marked with a red symbol in front of the subject.

Excluding all Subjects From the Documentation

You have the following options open to exclude all subjects from the documentation:

- In the documentation structure, right-click the project and click **Exclude all headings**.
- In the documentation structure, select the project and execute the menu command **Edit > Exclude all headings**.
- In the documentation structure, click beside the project symbol to toggle between **Include all headings** and **Exclude all headings**.

Updating and Preparing

The documentation is updated dynamically whenever a change is made in the project directory.

After every update the updated or new subjects are marked with a **(?)** symbol.

You have the following options to prepare the project for printing:

- In the documentation structure, right-click the project and click **Generate Printout**.
- In the documentation structure, select the project and execute the menu command **Edit > Generate Printout**.

The subjects correctly prepared have the page number **(n)** displayed after the subject.

NOTE: It is only when the project subject has a page number that all the subjects are prepared.

NOTE: Printable documents are generated for every subject during preparation for printing. By default the documentation is not saved, it is deleted at the end of the Control Expert session and must be generated again in the next Control Expert session. If you want to use these documents in other sessions, check the **Save Generated Printout** checkbox in the **Printout Setup** dialog box.

Print Preview

Introduction

You can call up a print preview for every subject.

You have the following options to open the print preview:

- In the documentation structure, right-click the desired subject and click **View**.
- In the documentation structure, select the desired subject and execute the menu command **View > View**.

Previous Page

You have the following options to show the previous page:

- In the documentation structure, right-click the displayed page and click **Previous Page**.
- In the documentation structure, select the desired displayed page and execute the menu command **Edit > Previous Page**.

Next Page

You have the following options to show the next page:

- In the documentation structure, right-click the displayed page and click **Next Page**.
- In the documentation structure, select the desired displayed page and execute the menu command **Edit > Next Page**.

Expand View

You have the following options to expand the view:

- In the documentation structure, right-click the displayed page and click **Zoom In**.
- In the documentation structure, select the desired displayed page and execute the menu command **Edit > Zoom In**.

Contract View

You have the following options to contract the view:

- In the documentation structure, right-click the displayed page and click **Zoom Out**.
- In the documentation structure, select the desired displayed page and execute the menu command **Edit > Zoom Out**.

View Full Page

You have the following options to show the whole page:

- In the documentation structure, right-click the displayed page and click **Zoom to Fit**.
- In the documentation structure, select the desired displayed page and execute the menu command **Edit > Zoom to Fit**.

Close

You have the following options to close the print preview:

- In the documentation structure, right-click the displayed page and click **Close**.
- In the documentation structure, select the desired displayed page and execute the menu command **Edit > Close**.
- Press the **Esc** key.

Print

Refer to the print section, page 1678.

Page Setup

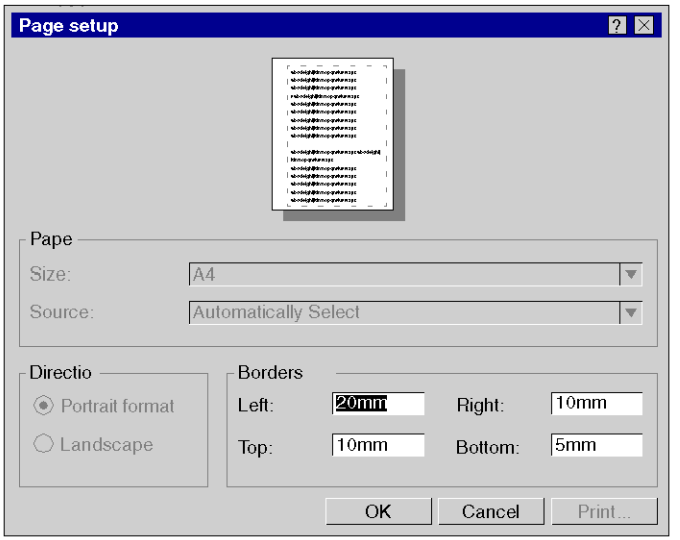
Introduction

The following options are available to open the setup dialog box:

- In the documentation structure, right-click the subject to include and click **Page Setup...**.
- In the documentation structure, select the subject to include and execute the menu command **Edit > Page Setup...**.

Representation

Representation of the dialog box:



Items

Elements of the dialog box:

Element	Description
Preview	Shows the preview for the documentation using the current settings.
Paper	**Size:** Shows the paper size (the paper format). This setting depends on the current printer and cannot be modified here. **Source:** Shows the paper source. This setting depends on the current printer and cannot be modified here.
Direction	Shows the orientation of the document on the page. This setting depends on the current printer and cannot be modified here.
Borders	**Left:** Enter the desired border distance from the left-hand edge of the page. **Right:** Enter the desired border width to the right-hand edge of the page. **Top:** Enter the desired border width to the top edge of the page. **Bottom:** Enter the desired border width to the bottom edge of the page.
OK	Saves all changes and closes the dialog box. You are not asked if you want to accept the changes, as this requires a new generation of documentation, page 1674.
Cancel	Cancels all changes and closes the dialog box.

Print

Introduction

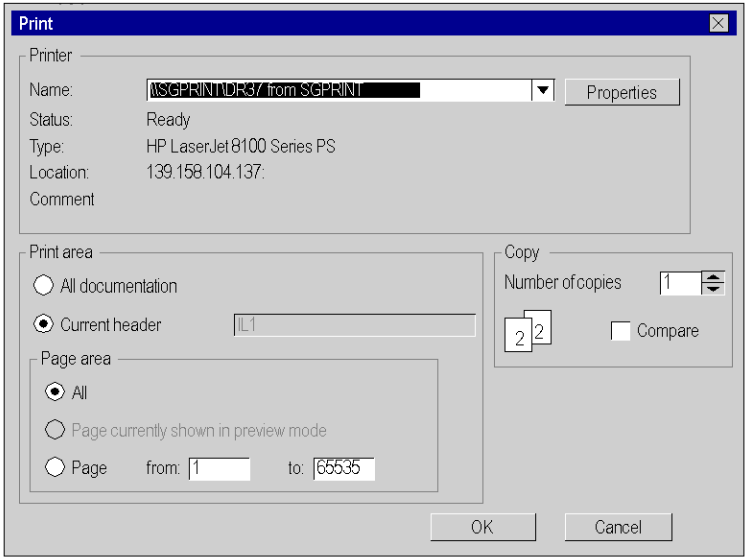
The following options are available to open the printer selection dialog box:

- In the documentation structure, right-click the subject to print and click **Print**.
- In the documentation structure, select the subject to print and execute the menu command **View > Print**.
- In the **print preview**, right-click the preview page and click **Print...**.

- In the documentation structure, select the subject to print and press the key combination **Ctrl**+**P**.

Representation

Representation of the dialog box:



Items

Elements of the dialog box:

Element	Description
Printer	**Name** Shows the name of the currently selected printer. You can select a different printer from the list box. **Status** Shows the status of the currently selected printer. **Type** Shows the type of the currently selected printer. **Position** Shows the path of the currently selected printer if you are connected to a network printer, or the connection to the computer if you are using a local printer. **Comment** Shows the comment for the currently selected printer. **Properties** Opens the properties dialog box for the currently selected printer.
Print range	**Whole documentation** If you click this option button, all the subjects, page 1673 in the document structure are printed out. **Current heading** If you click this option button, the currently selected subject and all its sub-sections are printed out. (The header for the current subject is shown in the text box).
Page range	This area is only available when the option button **Current heading** is activated. **All** If you click this option button, all the pages in the currently selected subject and all its sub-sections are printed out. **Currently previewed page** If you click this option button, the page currently showing in the preview is printed out. (This option is only available if Print preview, page 1675 is active.) **Pages from / to** If you click this option button, you can set which pages in the currently selected subject and/or all its sub-sections are printed out.
Copies	**Number of copies:**

Element	Description
	Here you can enter the number of copies to print. **Collate** If you check this checkbox, when several copies are printed, each set of document is printed together. If you uncheck this checkbox, when several copies are printed, several copies of each page are printed together.
OK	Closes the dialog box and starts the printout.
Cancel	Closes the dialog box without starting the printout.

Printout Setup

Introduction

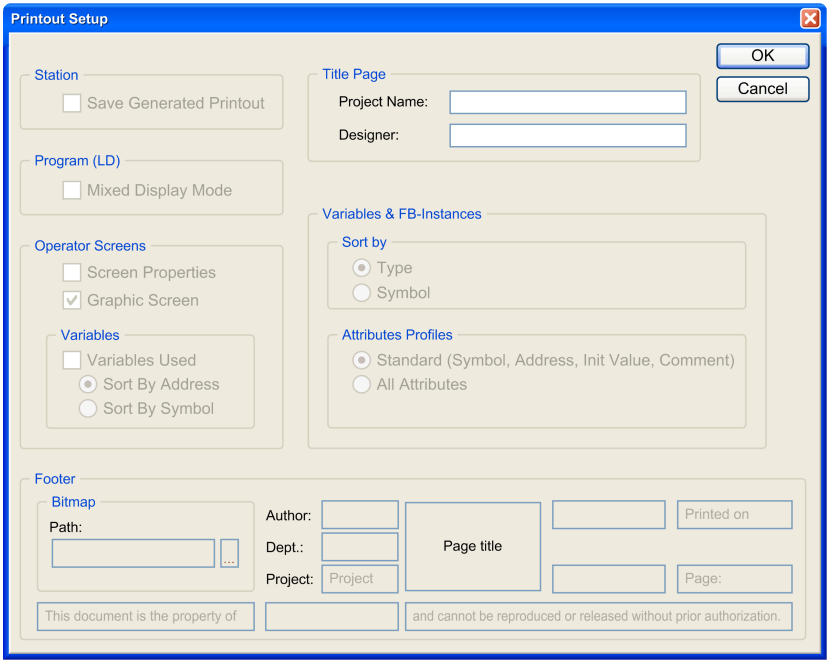
The following options are available to open the printing parameters dialog box:

- Right-click **Documentation > Title page** directory in the project browser and click **Printout Setup...**
- In the documentation structure, right-click a subject and click **Printout Setup...**.
- In the documentation structure, select a subject and execute the menu command **Edit > Printout Setup...**.

NOTE: For the first option, only the title can be modified, if you want to modify the other parameters, use the second or the third option to open the parameters dialog box.

Representation

Representation of the dialog box:



Items

Elements of the dialog box:

Element	Description
Station	**Save Generated Printout** You can save documents that have been prepared, page 1674 between Control Expert sessions (not saved by default). If you want to save documents, check the checkbox.
Program (LD)	This area is used to define the display for the documentation for the LD section, page 1668. **Mixed Display Mode** When you activate this checkbox, the documentation for the LD section is shown in combined display mode, page 755.

Element	Description
	When you deactivate this checkbox, the documentation for the LD section is shown in input mode, page 754.
Operator Screens	This area is used to define the scope and display for the operating screen documentation, page 1671. **Screen Properties** When you activate this checkbox, the properties for the operating screens are documented. **Graphic Screen** When you activate this checkbox, the graphical displays for the operating screens are documented. **Variables > Variables Used** When you activate this checkbox, the variables used in the operating screens are documented. **Variables > Sort By Address** If you click this option button, the variables used are sorted according to the address. **Variables > Sort By Symbol** If you click this option button, the variables used are sorted according to the symbol.
Title page	This area is used to define the title page, page 1664 for the documentation. **Project Name** Enter the name of the project. **Designer** Enter the name of the designer.
Variables & FB-Instances	This area is used to define the scope and display for the variables and FB instances, page 1667. **Sort by > Type** If you click this option button, the variables and FB instances used are sorted according to type. **Sort by > Icon** If you click this option button, the variables and FB instances used are sorted according to symbol. **Attributes Profiles > Standard (Symbol, Address, Init Value, Comment)** If you click this option button, only the default attributes for the variables and FB instances used are documented.

Element	Description
	Attributes Profiles > All Attributes If you click this option button, all attributes for the variables and FB instances used are documented.
Footer	This area is used to define the footer, page 1672 for the documentation. **Bitmap > Path:** Here you can enter the path for the bitmap file (for example, logo) to appear in the footer. **Author:** Enter the name of the author. **Dept.:** Enter the name of the department. **Project:** Displays the project name entered in **Title Page**. **Page title** Automatically generated and contains the title of the subject printed. **Free fields:** You can enter any text here. **Printed on** Automatically generated. **Page:** Automatically generated
OK	Saves all changes and closes the dialog box. **NOTE:** Subjects whose settings have to be changed, need to be generated again, page 1674.
Cancel	Cancels all changes and closes the dialog box.

Import / Export

What's in This Chapter

General information on the Import/Export function 1686
Importing/Exporting the various elements of Control Expert 1696
Wizard for importing the various elements of Control Expert 1727

Subject of chapter

This chapter describes the procedures for importing or exporting the project program or elements thereof. For example:

- import / export of the inputs / outputs configuration,
- import / export of DFB types,
- import / export of a program section,
- import / export of runtime screens,
- import / export of an animation table, etc.

It is also possible to Import/Export Project settings, page 549 and Import/Export Options, page 595.

General information on the Import/Export function

Object of this sub-chapter

This sub-chapter presents general information on the Import/Export function available under Control Expert.

General Information on Import/Export

At a Glance

The import / export functions are accessible from the structural view and the functional view of the project browser:

- The import function allows you to retrieve part or all of a project program for use in your project,
- The export function allows you to copy part or all of your project program to a file. You can then retrieve this program for use in a new project using the import function.

Export Function

The export function generates a file that contains **unprotected data** and **references to protected data**. You must define the name of this file and its location (directory), the extension of which is determined by the export type (for example, XDB if you export a DFB type).

Partial Program Export With File Encryption Active

When file encryption option is enabled, the export of program elements with hardened encrypted DFB type (such as section, Program unit, task, variable, DFB type, ...), requires entering the file encryption password.

Import Function

When you perform a partial import, the software asks you if you want to save the current project. This allows you to have a pre-import version of the project.

If during an import an element with the same name is found in the project, a dialog box allows you to choose between a number of operating modes:

- **Keep**: the element that is already present in the project is kept (the element with the same name is not imported),
- **Replace**: the element that is already present in the project is replaced by the imported element with the same name,
- **Rename**: you can change the name of the element to be imported in order to remove the conflict.

Which of these three operating modes you choose depends on the element you wish to import.

Following an import, you need to confirm the imported data (analysis and generation). In fact, import works like a manual input and **is not confirmed automatically**.

> **NOTE:** When importing elements with leading digits or unicode characters, adjust project settings (**Tools > Project Settings... > Variables > Allow leading digits** and **Tools > Project Settings... > Variables > Character set**) to avoid detected errors during import or at build.

Import / Export Files

At a Glance

The files generated during an export contain **unprotected data** and/or **coded protected data**.

The extension of each file is defined by its content, that is to say by your position in the directory tree of the project browser at the time of the export.

Import / Export Files

The files generated during an export are as follows:

Element to be exported	File type
Inputs/outputs configuration	XHW
Inputs/outputs configuration with global DTM configuration	ZHW
Program	XPG
Program Unit	XPU
Section in LD language	XLD
Section in IL language	XIL
Section in ST language	XST
Section in FBD language	XBD
Section in SFC language	XSF
Segment in LL984 language	X9S
Network in LL984 language	X9N
DFB type	XDB
DDT	XDD
Variables	SCY / TXT / XSY / XVM
Communication network(s)	XCM
Runtime screen(s)	XCR
Animation table	XTB
Functional module	XFM

Element to be exported	File type
Global project	XEF
Global project with global DTM configuration	ZEF

NOTE: When the first letter of the file extension is 'X', it means that the file is in XML format (extensible markup language).

Type of dialog box for the export function

At a Glance

When you select the export function from the Control Expert menu or from an element to be exported, two types of dialog boxes appear on the screen. These dialog boxes allow you to define:

- The name of the export file,
- The location of the file,
- The type of file,
- The export options.

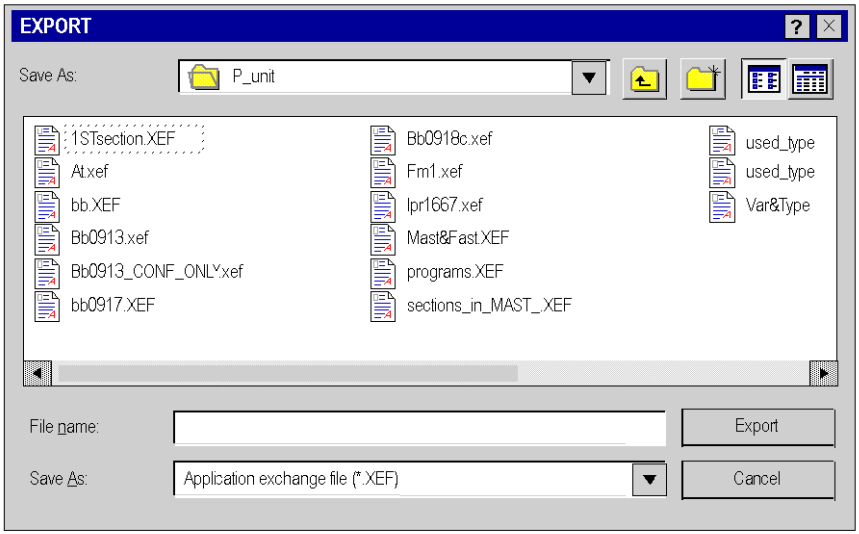
Standard dialog box

The following dialog box appears when you export one of the following element types:

- Inputs/outputs configuration, page 1696,
- DFB type, page 1703,
- DDTs, page 1705,
- Communication network, page 1711,
- Operator screen, page 1713,

NOTE: For all these elements to be exported, simply define the name of the export file and its location. The type of file under which the export is saved cannot be modified.

The following example shows the dialog box for exporting a global project:



Dialog box and options

To export the following elements, you must define the name of the file with its location. You can choose 1,2 or 3 options:

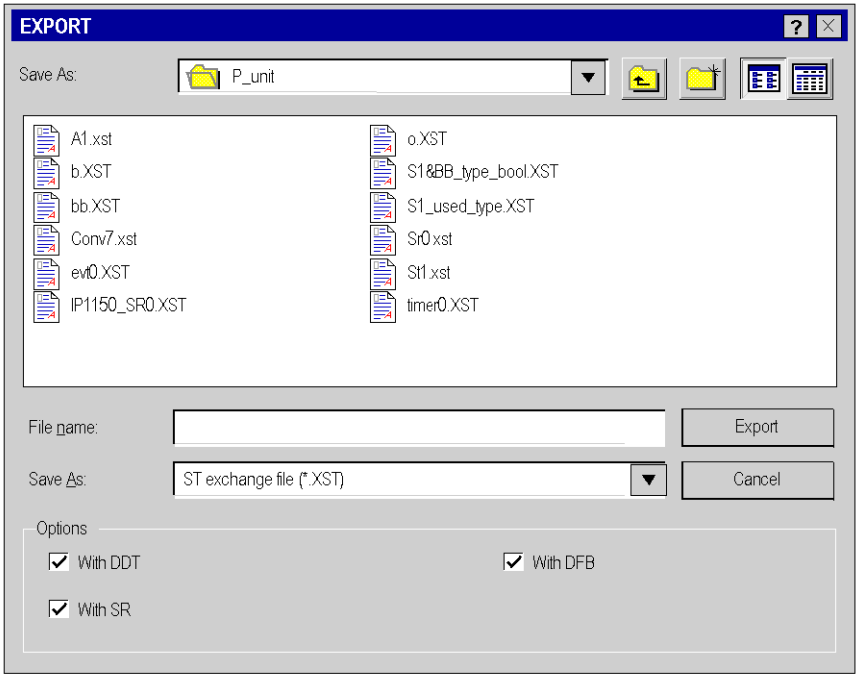
- Program, Program Unit and section, page 1699,
- Variables, page 1707,
- Animation table, page 1716,
- Functional module, page 1718.
- Global project, page 1720.

The following table describes the available options as a function of the elements to be exported:

Type of element to be exported	Choice of file type	Export mode with or without DDT option	Export mode with or without DFB type option	Export mode with or without SR option	Export mode with or without conf option
Program	no	yes	yes	yes	no
Program Unit	no	yes	yes	no	no
Section	no	yes	yes	yes	no

Type of element to be exported	Choice of file type	Export mode with or without DDT option	Export mode with or without DFB type option	Export mode with or without SR option	Export mode with or without conf option
Event Section	no	yes	yes	no	no
Variables	yes	yes	yes	no	no
Animation table	no	no	no	no	no
Functional module	no	yes	yes	no	no
Global project	no	no	no	no	yes

The following example shows the export dialog box for a section element, including the checkbox options:



Type of dialog boxes for the import function

At a Glance

When you select the import function from the Control Expert menu or from an element to be imported, various dialog boxes appear on the screen. These dialog boxes allow you to define:

- The name of the import file,
- The location of the file,
- The type of file,
- The import option,
- The import procedures (Keep, Replace, Rename).

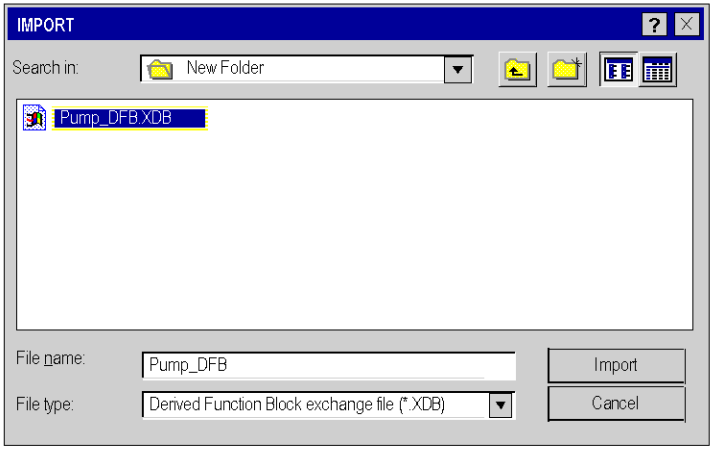
Standard dialog box

The following dialog box appears when you import one of the following element types:

- Inputs/outputs configuration, page 1698,
- DFB type, page 1704,
- DDTs, page 1706,
- Variables, page 1710,
- Communication network, page 1712,
- Animation tables, page 1717,
- Operator screens, page 1715.

NOTE: For all these elements to be imported, simply define the name of the import file and its location. The type of file under which the import is chosen cannot be modified except for importing variables.

The following example describes the dialog box for importing a DFB type:



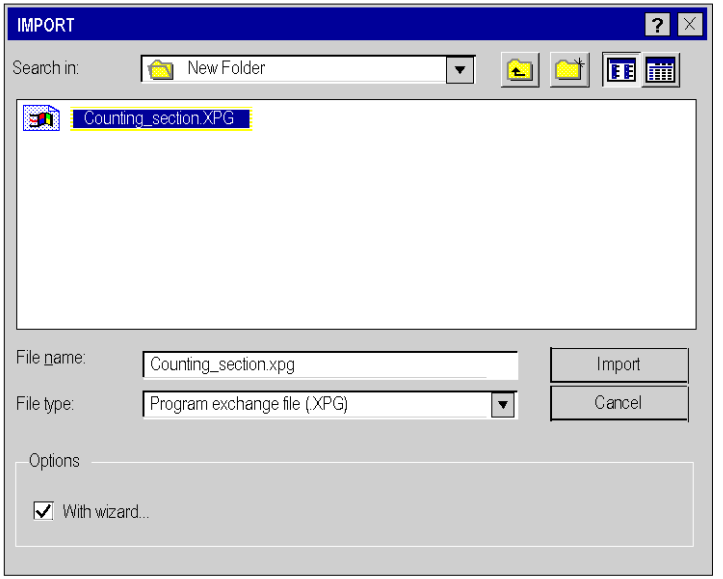
Dialog box with wizard

In order to facilitate reassignments, you can choose to import the following elements using wizard, page 1727:

- Global project, page 1722,
- Program Units, page 1701,
- Sections, page 1701,
- Functional modules, page 1719.

NOTE: For all these elements to be imported, simply define the name of the import file and its location and check the option **With wizard**.

The following example shows the dialog box for importing a section:



Managing Conflicts

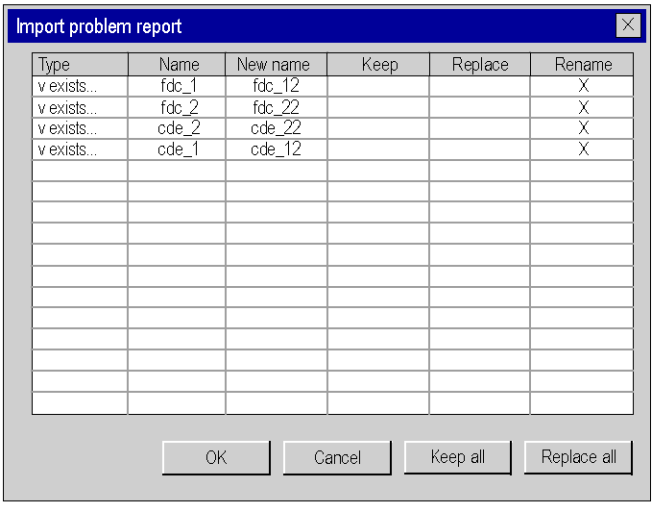
At a Glance

When importing an element with the same name, if it already exists in the project, an **Import problem report** dialog box lets you choose from several operating modes, page 1687:

- Keep,
- Replace,
- Rename.

These operating modes depend on the element to be imported. Excluding the global project import function, all the other elements have these operating modes.

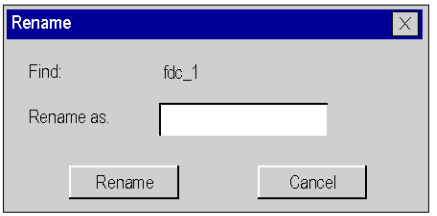
The following example shows the dialog box with conflicts when importing variables:



How to Manage Conflicts

You should perform the following steps in order to import an element and manage conflicts during the import process:

Step	Action
1	In the project browser, select the element to be imported. **Result**: The selected element appears highlighted.
2	Activate the **Import** command from the **File** menu or from the contextual menu (by clicking the right mouse button). **Result**: A dialog box appears on the screen:
3	Choose the source folder for the import (directory tree) then select the file to be imported. For the **Variables** element, the file type can be modified (XSY or TXT). **Result**: The name of the file then appears in the **File name** field.
4	Activate the **Import** command. **Result**: If there is no conflict, a progress indicator appears during the import, otherwise the **Import problem report** dialog box appears on the screen.
5	If you do not want to import the elements, check the **Keep** mode.
6	If you want to replace the elements with the imported ones, check the **Replace** mode. For further information on the replacement of Function Block (FB) instances see Changing the type of an instance, page 387.

Step	Action
7	If you want to rename the elements, check the **Rename** mode. **Result**: A dialog box appears on the screen where you can enter the name for the element to be modified. 
8	Start the import with your chosen option (Keep, Replace, Rename) using the **OK** command. If a conflict still exists, the **Import problem report** appears on the screen. Repeat steps 5, 6 and 7. If you want to cancel the import, activate the **Cancel** command.

Importing/Exporting the various elements of Control Expert

Object of this sub-chapter

This sub-chapter presents the Import/Export functions that apply to the various elements of Control Expert.

Export of the input/output configuration

At a Glance

This function can be accessed from the **Configuration** folder in the project browser. It allows you to export the complete configuration of the inputs/outputs of an existing project, that is to say its physical description, in order to be able to use it in a new project. You can, for example, export:

- the bus parameters
- the racks
- the modules

- the channel parameters
- the DTM configuration (in .ZHW file type only)

NOTE: Partial configuration export (.XHW) is available for M580 projects so that the .XHW file can be imported by a CAD electrical drawing software. However, the partial configuration import (.XHW) remains forbidden.

File generated

When you export the input/output configuration, you copy **the entire configuration** of the project to an .XHW or .ZHW file. The .XHW file is in XML format.

File type:

- .XHW: Input/output configuration file
- .ZHW: DTM + input/output configuration file

Exporting the input/output configuration

To export the input/output configuration, perform the following operations:

Step	Action
1	In the project browser, select the **Configuration** folder. **Result**: The selected element appears highlighted.
2	Activate the **Export** command from the **File** menu or from the context-sensitive menu (accessible with a right-click). **Result**: A dialog box appears on the screen.
3	Select the destination directory for the export (directory tree).
4	Select the file type (.XHW or .ZHW) and enter the file name.
5	Activate the **Export** command. **Result**: A progress indicator lets you know how the export is proceeding.
6	A message in the **output window** tells you that export is complete.

Import the Input/Output Configuration

At a Glance

This function is accessible from the **Configuration** folder in the project browser. It allows you to import the entire input/output configuration of a project. If you have already configured your project (for example if you have modified the default configuration), the software offers you a choice between:

- Replace the existing input/output configuration with the one contained in the import file.
- Cancel the import to keep your existing input/output configuration.

File type:

- `.XHW`: Input/output configuration file
- `.ZHW`: DTM and input/output configuration file

NOTE: Despite partial configuration export (.XHW) is available for M580 projects so that the .XHW file can be imported by a CAD electrical drawing software. The partial configuration import (.XHW) is not available.

Import the Input/Output Configuration

Step	Action
1	In the project browser, select the **Configuration** folder. **Result**: This folder appears highlighted.
2	Activate the **Import** command from the **File** menu or from the contextual menu (accessible with a right-click). **Result**: A dialog box appears on the screen.
3	Choose the source directory for the import (directory tree).
4	Choose the file type (`.XHW` or `.ZHW`) and select the file to import. **Result**: The name of the file appears in the **File name** field.
5	Activate the **Import** command. **Result**: A message tells you that the import is complete. If any error is detected during import, you are informed by a message in the **Output Window**.

I/O Data Type Import Behavior

The imported objects keep their original I/O data type.

The imported object behavior at import depends on its I/O data type:

- Topological I/O data type:

  The imported object respects its initial topological addressing.
- Device DDT I/O data type:
  - There is no variable with the same name in the application.

    A device DDT is created with the initial name.
  - A variable with the same name exists and it is managed in the application.

    A device DDT is created with a new name assigned to the object.
  - A variable with the same name exists but it is not managed in the application.

    The existing device DDT is linked to the imported element and becomes managed in the application.

Exporting the program

At a Glance

This function is accessible from the project browser.

Depending on the element that you have selected in the directory tree of the project browser, it allows you to export:

- either all the program elements of a sequential task (MAST, FAST, AUX),
- or a Program Unit of a sequential task (excluding the activation conditions),
- or a section of a sequential task (excluding the section activation conditions),
- or a section of a Program Unit (excluding the section activation conditions),
- or an SR program module of a sequential task,
- or **Timer** or **Input/Output** event processing.

  NOTE: When you export an SFC section, the sections containing the actions and transitions (whether referenced or not) are also exported.

  NOTE: When you export a conditioned section, the conditions are not exported. They are exported only if you export the complete task.

Options associated with exporting the program

Depending on the program elements you are exporting, you have a choice of 3 options:

- export with or without the content of all the DDTs used in the program element,
- export with or without the content of all the DFB types used in the program element,
- export with or without the content of all the SR program modules called in the program element.

These 3 **Options** are selected by default.

Files generated

When you export a set of program elements or event processing elements, the software generates a **.XPG** file which contains the following **unprotected information**: Program Units, sections, SR program modules, event processing elements, unprotected DFB types used (if the option is activated), DDTs used (if the option is activated).

When you export a Program Unit, the software generates a **.XPU** file which contains the following **unprotected information**: sections, SR program modules, event processing elements, unprotected DFB types used (if the option is activated), DDTs used (if the option is activated).

When you export a section or SR program module or an event processing element, the software generates a file according to the programming language used:

- **.XLD**, if the language of the element exported is LD
- **.XIL**, if the language of the element exported is IL
- **.XST**, if the language of the element exported is ST
- **.XSF**, if the language of the element exported is SFC (sections only)
- **.XBD**, if the language of the element exported is FBD

The file contains the **unprotected information**: program section, module or current event processing, unprotected DFB types used (if the option is activated), DDTs used (if the option is activated), program modules called (if the option is activated).

NOTE: The sections protected (read/write) by the active project protection, page 192 cannot be exported.

Exporting a program from the project browser

To export a program from the project browser, perform the following operations:

Step	Action
1	Select the element to be exported. **Result**: The selected element appears highlighted.
2	Activate the **Export** command from the **File** menu or the contextual menu. **Result**: A dialog box appears on the screen.
3	Select the destination directory for the export (directory tree) and enter the file name.
4	Select export mode: with or without DDT, DFB types, SR called. **Result**: When a **With** option is selected, the corresponding box is checked.
5	Activate the **Export** command. **Result**: A progress indicator lets you know how the export is proceeding.
6	A message in the **output window** tells you that export is complete.

Import the Program

At a Glance

This function is accessible from the Project Browser. According to the element you selected in the project browser tree, you can use it to import:

- All of the program elements (Program Units, sections and program modules) of a sequential task (MAST, FAST, AUX). The program elements are created in the same order as they were when memorized during export.
- One or all of the Program Units of a sequential task. These Program Units are imported after the program element that already exist in the list.

  If you need to perform reassignments, a wizard is available (except for protected Program Units which cannot be modified).
- One or all of the sections of a Program Unit. These sections are imported after the sections that already exist in the list of the Program Unit.

  If you need to perform reassignments, a wizard is available (except for protected sections which cannot be modified).
- One or all of the sections of a sequential task. These sections are imported after the sections that already exist in the list.

  If you need to perform reassignments, a wizard is available (except for protected sections which cannot be modified).
- One or all of the program modules of a sequential task. These modules are imported after the modules that already exist in the list.

- One or all of the **Timer** or **Input/Output** event processing operations. These processing operations are imported after the event processing operations that already exist in the list.

If one of the previous elements already exists in the current project, the software asks you to choose between the following options:

- **keep** the component of the current project,
- **replace** the project component by the one from the import file,
- **rename** the imported element.

NOTE: When importing a program element, if you wish to **keep** an element which has the same name in the project (program), then you need to rename the imported element (the name of an element is unique in a project).

NOTE: If the DDT, DFB types, and SR known as associated were exported (options selected during export), they are imported with the program element.

NOTE: After importing the Program Unit, section or program module, validate them.

Import a Program Element

Perform the following operations to import a program element:

Step	Action
1	In the project browser, select the folder into which you wish to import your program element (Program Units, sections, SR or events). **Result**: This folder appears highlighted.
2	Activate the **Import** command from the **File** menu or from the contextual menu. **Result**: A dialog box appears on the screen.
3	Choose the source folder for the import (directory tree) then select the file to be imported (XPG, XLD, XIL, XST, and so on). **Result**: The name of the file then appears in the **File name** field.
4	Select the **With wizard...** option, page 1727 to benefit from various options in reassigning elements.
5	Activate the **Import** command. **Result**: Aprogress indicator shows how the import is progressing. A message tells you when the import is complete. If errors are detected during the import, they are indicated by a message in the **display window**.

Exporting DFB types

At a Glance

This function is accessible via the **Derived FB types** folder in the project browser.

Depending on the element that you have selected in the directory tree of the project browser, it allows you to export:

- either all the DFB types of the current project (even those that have not been used),
- or one DFB type.

  NOTE: if you export a DFB type, which uses nested DFBs or DDTs, these are also exported automatically.

Protected DFB.

A **protected** DFB type (in write or read/mode) can be exported and keeps the same type of protection.

File generated during export of all DFBs

When you export all the DFB types of the project, the software generates an **.XDB** file containing

- all the **unprotected information**: unprotected nested DFB types, DDTs used, and references to the protected DFB types

File generated during export of one DFB type

When you export one DFB type, the software generates an**.XDB** file containing

- all the **unprotected information**: current DFB type, nested DFB types, DDTs used, and references to the protected DFB types.

Exporting one or more DFB type(s) from the project browser

To export one DFB type or all the DFB types of a project, perform the following operations from the project browser:

Step	Action
1	Select the element to be exported. **Result**: The selected element appears highlighted.
2	Activate the **Export** command from the **File** menu or the contextual menu. **Result**: A dialog box appears on the screen.
3	Select the destination directory for the export (directory tree) and enter the file name.
4	Activate the **Export** command. **Result**: A progress indicator lets you know how the export is proceeding.
5	A message in the **output window** tells you that export is complete.

Importing DFB types

At a Glance

The **Derived FB Types Import** function allows the user to import one or several DFB types. The imported DFB can be either write protected or not write protected. If the DFB is write protected, the DFB elements can only be accessible in read only mode. If the DFB has version protection, the version number is only accessible as read only. Any nested DFB or DDT that are used to compose the imported DFB is also be imported.

If the imported DFB type already exists in the application, a pop-up window appears to allow the user to either:

- **keep** the element from the current project,
- **replace** the project element with the one in the import file,
- **rename** the imported element.

Importing a DFB type

The following steps can be used to import the DFB:

Step	Action
1	In the project browser, select the **Derived FB Types** folder. **Result**: This folder appears highlighted
2	Activate the **Import** command from the **File** menu or the contextual menu. **Result**: A dialog box appears on the screen.

Step	Action
3	Choose the source directory for the import (directory tree), then select the file to be imported (XDB). **Result**: The name of the file then appears in the **File name** field.
4	Activate the **Import** command. **Result**: A progress indicator lets you know how the import is proceeding or the popup window appears to choose between **keep**, **replace** or **rename**, in that case, choose and validate by the **OK** button.
5	A message tells you that the import is complete. If any errors occur during import, you are informed of these by a message in the **Output Window**.

NOTE: If the imported duplicate DFB is not renamed, the DFB might not function as expected. Multiple reuse of the same instance of either the imported DFB or copied DFB is not recommended because it could lead to improper execution of the DFB.

Exporting DDTs

At a Glance

The export functionality is accessible from the **Derived Data Types** folder in the project browser.

Depending on the element that you have selected in the directory tree of the project browser, it allows you to export:

- All the project DDTs (even those that have not been used).
- One DDT.

  NOTE: If you export a DDT that uses nested DDTs, these are also exported automatically.

File Generated During Export of all DDTs

When you export all the DDTs, the software generates an .XDD file that contains the **unprotected information** for the project DDTs.

File Generated During Export of One DDT

When you export one DDT, the software generates an .XDD file that contains the **unprotected information** of the DDT and the nested DDTs used.

Exporting One or More DDTs From the Project Browser

To export one DDT, or all the DDTs, of a project, perform the following operations from the project browser:

Step	Action
1	Select the element to be exported. **Result**: The selected element appears highlighted.
2	Select the **Export** command from the **File** menu or the contextual menu. **Result**: A dialog box appears on the screen.
3	Select the destination directory for the export (directory tree) and enter the file name.
4	Select the **Export** command. **Result**: A progress indicator lets you know how the export is proceeding and a message in the **output window** informs you that the export is complete.

Importing DDTs

At a Glance

This function is accessible from the **Derived Data Types** folder in the project browser. It allows you to import one or several DDTs.

NOTE: If the DDT uses nested DDTs, these are also imported.

If a DDT type already exists in the current project, the software proposes the following options:

- **keep** the element from the current project.
- **replace** the project element with the one in the import file.
- **rename** the imported element.

Importing a DDT

To import a DDT, perform the following operations:

Step	Action
1	In the project browser, select the **Derived Data Types** folder. **Result**: This folder appears highlighted.
2	Activate the **Import** command from the **File** menu or the contextual menu. **Result**: A dialog box appears on the screen.
3	Choose the source directory for the import (directory tree), then select the **.XDD** file to be imported. **Result**: The name of the file then appears in the **File name** field.
4	Activate the **Import** command. **Result**: A progress indicator lets you know how the import is proceeding.
5	A message tells you that the import is complete. If any errors occur during import, you are informed of these by a message in the **Output Window**.

Exporting Variables

At a Glance

This function is accessible from the **FB Variables and instances** folder in the project browser.

According to the element you selected in the project browser tree, you can use it to export:

- all the variables of the project (even those that are not used),
- all the variables of a given family: EDT, IODDT, and so on (even those that are not used).

To export a subset of the variables, see also the sub-section Export of a subset of variables, page 437.

Excel Tool for Variables Management

An Excel template file is available in Control Expert extra tools to provide parameters management from files generated in *.XSY* format. The Excel features allow to create, edit, sort, or filter Control Expert application variables, and import them back in Control Expert.

The Excel template and its manual are installed with Control Expert: **Start > All Programs > EcoStruxure Control Expert > Extras > Excel Import Export Tool**. They are also available in the Control Expert software DVD (**Extras > Excel Import Export Tool**).

Format of Variable Files

Variables can be saved:

- either in source format, in an *.SCY* file compatible with the PL7 program,
- or in text format with tab separators, in a *.TXT* file,
- or in XML format, in an OFS-compatible *.XVM* file,
- or in XML format, in an *.XSY* file.

Options Associated with Exporting Variables

Whatever the variables you export, you can choose two options:

- export with or without the content of all the DDTs used,
- export with or without the content of all the DFB types used.

By default, the two "export with" options are selected.

NOTE: These two options are available only when exporting files to *.XSY* format.

Purpose of Importing Data

Importing data into a specific format as *.XSY*, *.SCY*, *.TXT* or *.XVM* can be useful for different purposes.

Those functionalities are described in the table below:

File type	Available for
.XSY	Export / Import
.SCY	Export / PL7

File type	Available for
.TXT	Export / Import / Excel
.XVM	Export / OFS

Files Structure For .XSY, .SCY, .TXT And .XVM Formats

When importing elementary variables from the data editor screen, the data structure with extension *.XSY*, *.SCY*, *.TXT*, and *.XVM* is displayed in the following order: *Name*, *Address*, *Type*, *Comment*, and *Value*.

NOTE: The structure of imported data from a *.XSY* file can have additional information. If the options are set in column configuration screen, the complete structure for this type of file is the following: *Name*, *Address*, *Type*, *Comment*, *Value*, *Save*, *RW Program*, *Constant* and *Custom*.

Files Generated In .XSY And .XVM Formats

When you export variables, the program generates an *.XSY* or an *.XVM* file containing the unprotected information, such as the list of current variables, the unprotected DFB types of which an instance is exported (if the option is set), and the DDTs of which an instance is exported (if the option is set).

Files Generated In .SCY, .XVM, .TXT Formats

When you export variables, the program generates a *.TXT* or an *.SCY* file containing the unprotected information about current variables.

If you want to use Microsoft Excel to open the exported *.TXT* file, please follow the steps:

- Open Microsoft Excel
- With **Open** command of Excel, open the **.TXT** file by selecting in the open window **Text files** as type of file.
- When the Excel file import assistant window opens, select **Delimited** for the type of file. Click on the **Next** button.
- In the next window, check **Tab** option box as column separator. Click on the **Next** button.
- In the next window select in the display the column that contain the values of the variables and click on the **Text** option button for the column data format. Doing so keeps the same values as in the *.TXT* file. If this format is not used for the column, positive values will appear as negative when opening the *.TXT* file in Excel.

NOTE: The variables must not exceed 32 characters in length. Once you have modified the *.TXT* file, save it with the tab-separated value option before importing.

How to Export a List of Variables from the Project Browser

Perform the following operations to export all the variables or a family of variables from the project browser:

Step	Action
1	Select the element to be exported in the **FB Variables and instances** folder. **Result:** The selected element appears highlighted.
2	Activate the **Export** command from the **File** menu or from the contextual menu. **Result:** A dialog box appears on the screen:
3	Select the target folder for the export (directory tree) then enter the file name.
4	Select the export mode. with or without DDT, DFB type. **Result:** When a "with" option is selected, the corresponding box is checked.
5	Activate the **Export** command. **Result:** A progress indicator shows how the export is progressing.
6	A message is displayed in the **display window** to indicate that the export is finished.

Importing Variables

At a Glance

This function is accessible from the **FB Variables and instances** folder in the project browser. It enables you to import a list of variables into your project.

NOTE: If the DDTs or associated DFB types have been exported (options selected on export), they are imported with the list of variables.

If the list of variables already exists in the current project, the software asks you to choose between the following options:

- **keep** the component of the current project,
- **replace** the project component by the one from the import file,
- **rename** the imported component, allowing you to keep both components.

Excel Tool for Variables Management

An Excel template file is available in Control Expert extra tools to provide parameters management from files generated in *.XSY* format. The Excel features allow to create, edit, sort, or filter Control Expert application variables, and import them back in Control Expert.

The Excel template and its manual are installed with Control Expert: **Start > All Programs > EcoStruxure Control Expert > Extras > Excel Import Export Tool**. They are also available in the Control Expert software DVD (**Extras > Excel Import Export Tool**).

How to Import a List of Variables

Perform the following operations to import a list of variables:

Step	Action
1	In the project browser, select the **FB Variables and instances** folder. **Result**: The **Variables** folder appears highlighted.
2	Activate the **Import** command from the **File** menu or from the contextual menu (by clicking the right mouse button). **Result**: Adialog box appears on the screen:
3	Choose the source folder for the import (directory tree) then select the file to be imported (XSY or TXT). **Result**: The name of the file then appears in the **File name** field.
4	Activate the **Import** command. **Result**: A progress indicator shows how the import is progressing.
5	A message tells you when the import is complete. If errors occur during the import, they are indicated by a message in the **display window**.

NOTE: You can only import **.XSY** or **.TXT** files. Files in **XVM** and **SCY** format cannot be imported.

Exporting the communication network configuration

Introduction

This function is accessible from the **Communication** folder of the project browser.

Depending on the element selected in the directory tree of the project browser, it allows you to export:

- All communication networks with their configurations.
- A specific communication network with its configuration.

File generated during export of networks

When you export one or all of the communication networks, the software generates a **.XCM** file containing all the information on each network: common data, bridge.

How to export an operator screen

To export the communication networks, perform the following operations:

Step	Action
1	In the **Project** browser, select the element to be exported. **Result**: The selected element appears highlighted.
2	Activate the **Export** command from the **File** menu or the contextual menu. **Result**: A dialog box appears on the screen.
3	Select the destination directory for the export (directory tree) and enter the file name.
4	Activate the **Export** command. **Result**: A progress indicator lets you know how the export is proceeding.
5	A message in the **output window** tells you that export is complete.

NOTE: If you check the **With Conf.** option box before the export, the networks are exported with the links to the modules of the project. When you import the networks in another project they are automatically linked to the existing modules. This option works only if the modules of the other project are compatible with those of the previous one and if they are placed in the same positions in the rack.

Importing the configuration of a communication network

At a Glance

This function is accessible from the **Communication** folder of the project browser.

You can import one or more communication network configurations.

If a communication network already exists in the current project with the same name, the software proposes the following options:

- **keep** the element from the current project.
- **replace** the project element with the one in the import file.
- **rename** the imported element.

How to import a functional module

To import a communication network, perform the following operations:

Step	Action
1	In the function view of the project browser, select the **Communication** folder. **Result**: This folder appears highlighted.
2	Activate the **Import** command from the **File** menu or the contextual menu. **Result**: A dialog box appears on the screen.
3	Choose the source directory for the import (directory tree), then select the file to be imported (XCM). **Result**: The name of the file then appears in the **File name** field.
4	Activate the **Import** command. **Result**: A progress indicator lets you know how the import is proceeding.
5	A message tells you that the import is complete. If any errors occur during import, you are informed of these by a message in the **Output Window**.

Exporting operator screens

Introduction

This function is accessible from the **Operator Screens** folder in the project browser.

Depending on the element that you have selected in the directory tree of the project browser, it allows you to export:

- All the operator screens of the project.
- All operator screens in the same family.

- A specific operator screen.

  NOTE: The messages associated with the screens are only exported if you export all the operator screens, page 1714 of the project.

Files generated during the export of an operator screen

When you export one or several operator screens, the software generates an **.XCR** file for each family of screens. This file contains all the information relating to the graphic objects and the reference of the image objects (.bmp, .jpg). The image object references correspond to their names and their access paths.

How to export an operator screen

To export an operator screen (all screens, a family of screens or a single screen), perform the following operations:

Step	Action
1	In the **Project** browser, select the element to be exported. **Result**: The selected element appears highlighted.
2	Activate the **Export** command from the **File** menu or the contextual menu. **Result**: A dialog box appears on the screen.
3	Select the destination directory for the export (directory tree) and enter the file name.
4	Activate the **Export** command. **Result**: A progress indicator lets you know how the export is proceeding.
5	A message in the **output window** tells you that export is complete.

Import/Export of messages

The messages associated with the screens are only exported if you export all the operator screens of the project. If you export a single operator screen or an operator screen family, the messages are not exported.

That means when you export a single screen from project A and import it to project B, you need to redefine all messages used inside this screen in the **Messages list** of Project B.

As a workaround you can export and import all messages as described in Messages editing and import/export function, page 1631.

Another way to use messages created in project A in screens imported to project B, is to open project A and project B in parallel and to copy/paste the messages from the **Messages list** in project A to the **Messages list** in project B.

You have to take care that the **Message numbers** meet the demands of your new project.

Import Operator Screens

At a Glance

This import functionality is accessible from the **Operator Screens** folder in the project browser.

In the **Operator Screens** folder, if you are importing one or several screens with their families you can:

- Import a family of screens, in which case it is then inserted in the folder. The screens belonging to this family continue to be located in their families of origin.
- Import the screens without families directly in the root of the folder (they do not belong to a family).

  NOTE: If you import a file containing references to images and the access path to these images is no longer correct, the images do not appear in the imported screen.

  NOTE: Screens are only reattached to the functional modules to which they were attached before export when you import a complete ZEF, XEF or XFM file. If you import an operator screen in the structural view (XCR file), the attachment between operator screen and functional module is discarded. If you import an operator screen in the functional view (XCR file), it is attached to the functional module from which the import was launched.

Import and Export Messages

Refer to *Import/Export of Messages*, page 1714.

Import an Operator Screen

Step	Action
1	In the project browser, select the **Operator Screens** folder into which you wish to import your screen or family of screens. **Result**: This folder appears highlighted.
2	Select the **Import** command from the **File** menu or the contextual menu. **Result**: A dialog box appears on the screen.
3	Choose the source directory for the import (directory tree), then select the file to be imported (. XCR). **Result**: The name of the file then appears in the **File name** field.
4	Select the **Import** command. **Result**: A progress indicator lets you know how the import is proceeding and a message tells you that the import is complete. If errors are detected during import, you are informed by a message in the **Output Window**.

Exporting animation tables

At a Glance

This function is accessible from the **Animation Tables** folder of the project browser.

Depending on the element that you have selected in the directory tree of the project browser, it will allow you to export:

- all the animation tables of the project,
- a single animation table.

File generated during export of a single animation table

When you export an animation table, the software generates a **.XTB** file that contains all the **unprotected information** such as the current animation table.

Exporting a single animation table

To export one or all of the animation tables, perform the following operations from the project browser:

Step	Action
1	Select the element to be exported. **Result**: The selected element appears highlighted.
2	Activate the **Export** command from the **File** menu or the contextual menu. **Result**: a dialog box appears on the screen.
3	Select the destination directory for the export (directory tree) and enter the file name.
4	Activate the **Export** command. **Result**: a progress indicator lets you know how the export is proceeding.
5	A message in the **output window** tells you that export is complete.

Importing animation tables

At a Glance

This function is accessible from the **Animation Tables** folder of the project browser. It allows you to import one or more animation tables into your project.

If an animation table already exists in the current project with the same name, the software proposes the following options:

- **keep** the element from the current project,
- **replace** the project element with the one in the import file,
- **rename** the imported element.

Importing a single animation table

To import a single animation table, perform the following operations:

Step	Action
1	In the project browser select the **Animation Tables** folder. **Result**: This folder appears highlighted.
2	Activate the **Import** command from the **File** menu or the contextual menu. **Result**: A dialog box appears on the screen.
3	Choose the source directory for the import (directory tree), then select the file to be imported (XTB).

Step	Action
	Result: The name of the file then appears in the **File name** field.
4	Activate the **Import** command. **Result**: A progress indicator lets you know how the import is proceeding.
5	A message tells you that import is complete. If any errors occur during import, you are informed of these by a message in the **Output Window**.

Exporting a functional module

At a Glance

This function is accessible from the **function view** of the project browser.

It allows you to export all the elements from the functional module selected:

- functional module properties (name, comment),
- Program Units and/or sections (with the variables used, program code, activation condition),
- event processing,
- functional sub-modules (nested functional modules),
- operator screens,
- animation tables.

Options associated with exporting a functional module

As with the program export procedure, you have the choice of 3 options:

- export with or without definition of all the DDTs used in the functional module,
- export with or without definition of all the DFB types used in the functional module,
- export with or without the content of all the SR program modules called in the program element.

The 3 "export with" options are selected by default.

Files generated

When you generate a functional module, the software generates a **.XFM** file. This contains all the **unprotected information** such as the sections, event processing, DFB types used (if the option has been activated), DDTs used (if the option has been activated) and the SR program modules called (if the option has been activated). It also contains the references of the operator screen images.

How to export a functional module

To export a functional module, perform the following operations from the function view of the project browser:

Step	Action
1	Select the functional module to be exported. **Result**: The selected functional module appears highlighted.
2	Activate the **Export** command from the **File** menu or the contextual menu. **Result**: A dialog box appears on the screen.
3	Select the destination directory for the export (directory tree) and enter the file name.
4	Select export mode: with or without DDT, DFB types called. **Result**: When a "with" option is selected, the corresponding box is checked.
5	Activate the **Export** command. **Result**: A progress indicator lets you know how the export is proceeding.
6	A message in the **output window** tells you that export is complete.

Importing a functional module

At a Glance

This function is accessible from the **functional view** of the project browser.

You can import a functional module under:

- the project,
- in the existing module, the new module is then a nested module.

In all cases, the assistant is offered as an option.

NOTE: Screens are only re-attached to the functional modules to which they were attached before export when you import a complete ZEF, XEF or XFM file. If you import an operator screen in the structural view (XCR file), the attachment between operator screen and funtional module is lost. If you import an operator screen in the functional view (XCR file), it is attached to the functional module from which the import was launched.

How to import a functional module

To import a functional module, perform the following operations:

Step	Action
1	In the function view of the project browser, select the documentation file in which you wish to import your functional module (station or a functional module). **Result**: This folder appears highlighted
2	Activate the **Import** command from the **File** menu or the contextual menu. **Result**: A dialog box appears on the screen.
3	Choose the source directory for the import (directory tree), then select the file to be imported (XFM). **Result**: The name of the file then appears in the **File name** field.
4	Activate the **Import** command. **Result**: A progress indicator lets you know how the import is proceeding.
5	A message tells you that import is complete. If any errors occur during import, you are informed of these by a message in the **Output Window**.

Export a Global Project

At a Glance

This function is accessible from the **Project** folder of the project browser. It allows you to export the following elements of the project:

- all the DDTs
- all the variables (used and unused)
- all the unprotected DFB types

- the entire program: sections, SR program modules (with execution condition), event processing
- all the functional modules
- all the operator screens
- all the animation tables
- the project properties

This function optionally allows to export the following elements of the project:

- the input/output configuration
- the communication network configuration
- the global DTMs configuration (DTMs topology tree and device configurations)

NOTE: The EFs, EFB types, protected DFB types, documentation and user preferences are not exported.

Generated Files

These formats are available for file export:

- **.XEF**: Use this format to export a global project that does not have a global configuration for DTMs. (You cannot export an encrypted ..XEF file.)
- **.ZEF**: Use this format to export a global project that has a global configuration for DTMs. (You can export an encrypted .ZEF file.)

The export file contains the unprotected information: configuration of the inputs/outputs, sections, SR program modules, events processing, unprotected DFB types, DDTs, variables, animation tables, references to the protected DFB types, global DTMs configuration (ZEF file only).

When exporting protected DFBs, private variables and sections of the DFB are encrypted. However, the variables of type IN, OUT, IN/OUT and PUBLIC are still visible.

NOTE:

- Files in the .XEF or .ZEF formats do not contain the configuration of the TSXCPP110. Back up the .CO file (Sycon) for CANopen after a project export.
- When you export a project in the .XEF or .ZEF formats, the application password is cleared.

Maintain Output Links on Disabled EF

For compatibility of the project setting **Maintain output links on disabled EF (EN=0)**, refer to the respective section, page 581.

Export a Global Project

Use these operations from the **Project Browser** to export a global project:

Step	Action
1	Select the **Project** folder. **Result**: This folder appears highlighted.
2	Select the **Export project** command from the **File** menu or the contextual menu. **Result**: A dialog box appears on the screen.
3	Select the export format from the **Type** option: • **.ZEF** format to export a global project with global DTMs configuration • **.XEF** format to export a global project without global DTMs configuration. If the application contains DTMs, the global DTMs configuration is not exported.
4	Select the destination directory for the export (directory tree) and enter the file name.
5	If the ZEF export format is chosen, the option **With Encryption** is selected by default when application file encryption is enabled (only for EcoStruxure Control Expert version 15.0 Service Pack 1 (and any subsequent supporting version(s)). **NOTE:** Exporting an non-encrypted application file is possible by de-selecting the option **With Encryption**. In this case you need to enter encryption file password to perform the export and the exported file is no longer protected.
6	If the XEF export format is chosen, select the option **With Conf** to include the data relating to the hardware configuration in the file. **NOTE:** Managed Device DDT become unmanaged if the **With Conf** option is unchecked. Manually link the unmanaged Device DDT instances with each device.
7	Select the **Export** command. **Result**: A progress indicator lets you know how the export is proceeding and a message in the **output window** informs you that the export is complete.

Importing a global project

At a Glance

This function is accessible only from the **File > Open** menu of Control Expert.

The following files can be opened:

- **.ZEF** file (Full Application Exchange Files) to import a global project with global DTMs configuration

- **.XEF** file (Application Exchange Files) to import a global project without global DTMs configuration

Before opening the imported project, Control Expert proposes to save the current project before closing it.

Libset Version

When importing a **.ZEF** or **.XEF** file, the latest **Libset** version is always selected.

Importing protected elements

If the project you want to import contains protected sections, these are imported and their content is coded.

If the project you want to import contains protected DFB types, these are referenced in the .ZEF or .XEF file but not imported.

NOTE: .ZEF or **.XEF** files do not contain the configuration of the TSX CPP 110.

If the **.CO** file (**Sycon**) does not exist:

- a warning is displayed in the **Output Window** after the **Open** command,
- an error is displayed in the **Output Window** after the **Analyze** or **Generate** command.

In the configuration screen of the TSX CPP 110 PCMCIA card, select the required **.CO** file to update the CANopen configuration.

Importing operator screens

Screens are only re-attached to the functional modules to which they were attached before export when you import a complete ZEF, XEF or XFM file.

If you import an operator screen in the structural view (XCR file), the attachment between operator screen and functional module is lost.

If you import an operator screen in the functional view (XCR file), it is attached to the functional module from which the import was launched.

The operator screens are imported with the same structure as in the source project. If families of screens do not exist, they are automatically created.

Maintain Output Links on Disabled EF

For compatibility of the project setting **Maintain output links on disabled EF (EN=0)** please refer to the respective section, page 581.

How to import a global project

To import a global project, perform the following operations:

Step	Action
1	Activate the **Open** command from the **File** menu. **Result**: A dialog box appears on the screen.
2	Choose to save or not save your current project. **Result**: • Your project is saved or not saved, according to your selection. • A dialog box appears on the screen.
3	Select the file to be imported (.ZEF or .XEF). **Result**: The filename appears in the **File name** field.
4	Select the **Open import File with Wizard** option, if you require help reassigning elements.
5	Activate the **Open** command. **Result**: • Control Expert closes your current project. • A progress indicator lets you know how the import is proceeding.
6	A message tells you that the import is complete. If any errors occur during import, you are informed of these by a message in the **Output Window**.

Conversion Wizard

To convert a legacy project partially or as a whole you can use the **Conversion Wizard**.

The **Conversion Wizard** provides options to remap I/O objects (channels, variables etc.) during conversion and to adapt the hardware configuration of the new project concurrently in Control Expert.

Exporting/Importing LL984 Segments/Networks

Introduction

You can export/import LL984 segments and networks from the **Structural View** of the **Project Browser**.

Restrictions

In contrast to the **Export** dialog for other language sections, the dialog for LL984 segments/networks does not provide the **With SR** check box, because the subroutine networks called in the exported networks are not exported.

This is done because the called subroutine number can be given as a parameter at runtime, so you do not know in advance which subroutines are really called.

If you want to use the called subroutines in another application, you must export and re-import the whole LL984 subroutine segment.

Files Generated

Files with the following extensions are generated during export:

- **.X9S* for LL984 segments
- **.X9N* for LL984 networks

The created files also include the definitions of variables and variable types used inside the exported segments and networks.

Exporting LL984 Segments/Networks

Step	Action
1	Select the element to be exported (LL984 segment/network).
2	Right-click and select **Export** from the context menu or use the **File > Export** menu command.
3	Select the destination directory for the export and enter the file name.
4	Select / deselect the **With DDT**/**With DFB** options.
5	Click the **Export** button.
6	A message in the **Import/Export** tab of the output window informs you when export is completed.

Importing LL984 Networks

To import a **.X9N* file (LL984 network) you have to select an LL984 segment node in the **Structural View** of the **Project Browser**

Right-click and select **Import** from the context menu or use the **File > Import** menu command.

In case of naming conflicts for the network name or the used variables and variable types an **Import Trouble Report** dialog will be displayed. Please refer to Managing Conflicts, page 1694.

Importing LL984 Segments

To import a **.X9S* file (LL984 segment) you have to select **Sections** or **SR Sections** in the **Structural View** of the **Project Browser**.

Right-click and select **Import** from the context menu or use the **File > Import** menu command.

In case of naming conflicts an **Import Trouble Report** dialog will be displayed.

Importing a **.X9S* file to the **SR Sections** node can end up in the existence of more than one LL984 segment under **SR Sections**.

During **Analyze** this will be checked and will lead to an analyze error.

In this case move the LL984 subroutine networks needed into one LL984 SR segment and delete the other LL984 SR segments.

Even if the **Ladder Logic (LL984)** option under **Tools > Project Settings > Program > Languages** is not set, **.X9S* files can be imported without error messages and the corresponding segments and networks will be created. But in this case later during **Analyze** an `incorrect language` message will be raised for these segments and networks.

In the context menu of segments the menu entry to import networks is disabled in case the **Ladder Logic (LL984)** option is not set.

Wizard for importing the various elements of Control Expert

Object of this sub-chapter

This sub-chapter presents the functions of the wizard for importing the sections and functional modules under Control Expert.

Using the assistant

At a Glance

You can select the assistant to import the following elements:

- A global project
- A Program Unit
- A section
- A functional module

During the import you can use the assistant to facilitate the **reassignment** of the following types of element:

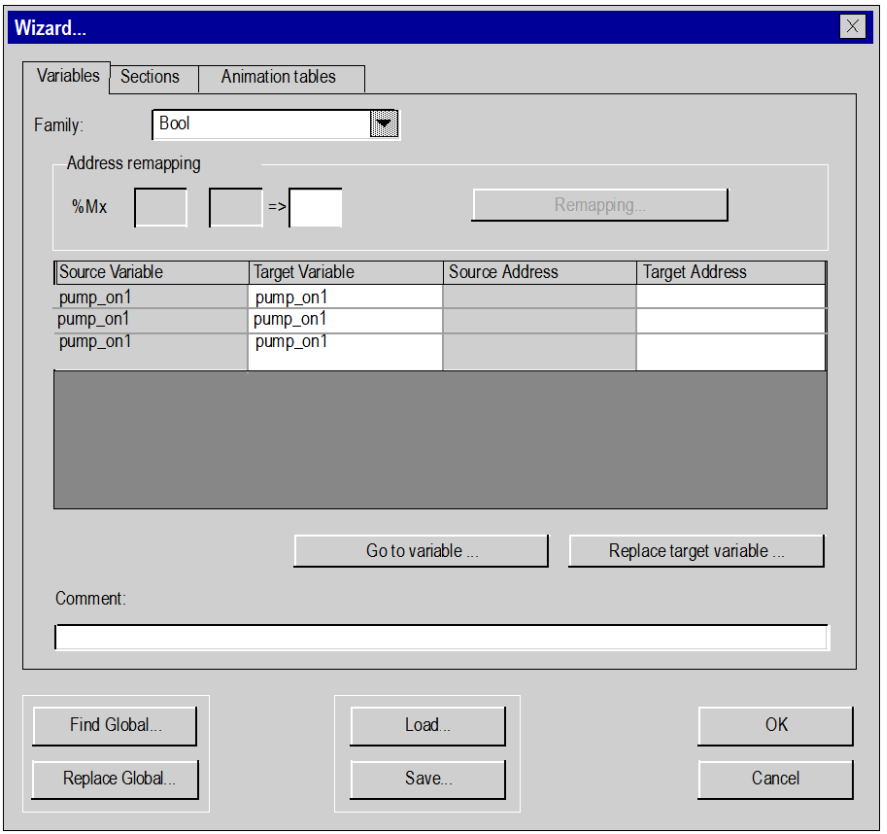
- Variables
- DDTs
- DFBs
- Sections (SR, macro SFC)
- Event processing (timer or Input/Output)
- Animation table
- Functional module

To do this, check the **With wizard** box in the import dialog box.

Description of the wizard

The wizard is accessible via a dialog box. This box contains one or more tabs. Each tab corresponds to an element to be imported.

The illustration that follows is an example of the dialog box when a functional module is imported:



The list of elements that cannot be reassigned (shaded columns) and the contextual functions allowing the reassignments are offered via the following tabs:

Tab	Description
DDT	Provides a list of the DDTs used. For each DDT, the screen gives the name of the DDT before and after reassignment, as well as the type of DDT and the comment.
DFB	Provides a list of instances of DFBs used. For each instance, the screen gives the name of the instance before and after reassignment, as well as the DFB type and the comment.
Variables	Provides a list of the variables used. For each variable, the screen gives the address and the symbol (before and after reassignment), as well as the comment.
Sections	Provides a list of program modules.
SR	Provides a list of SRs.
Macro steps	Provides a list of the sequential steps called.

Tab	Description
Events	Provides a list of Timer or System event processing elements called.
Animation tables	Provides a list of animation tables used.
Functional modules	Provides a list of elements (variables, sections, animation tables) that are in the functional module to be imported.

Common commands

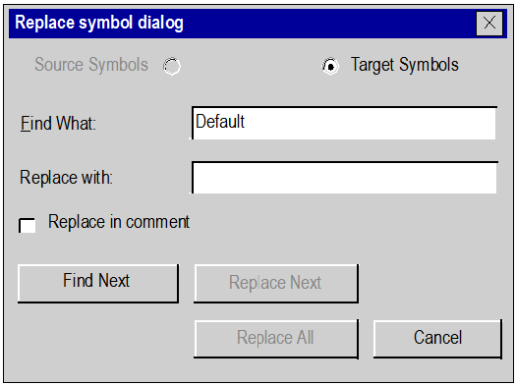
The wizard contains global commands:

- **Find global...** and **Replace Global...**
- **Load...** and **Save...**, page 1738

A symbol or character string search or replace action can be performed in all the wizard tabs or in a specific tab:

- To address all tabs shown in the wizard, select **Find global** or **Replace Global**.
- To address a specific tab, right-click in the selected tab and choose **Find** or **Replace**. The commands in the dialog box only affect the selected tab.

Dialog box associated with the **Replace** and **Replace Global...** commands:



To replace a symbol or character string:

- In the **Find What** field, enter the symbol or character string to replace.

  The **Find What** command is case-sensitive. The special character * is not allowed in character strings and does not act as a wildcard character to search any string or part of string.
- In the **Replace with** field, enter the new symbol or character string.

- Select **Replace Next** to replace the occurrences found one by one.

  Select **Replace All** to replace all symbol occurrences in one operation.

  NOTE: If the dialog box is opened using the **Replace** command in a tab, the **Replace All** command only affects the selected tab. If the dialog box is opened using the wizard **Replace Global...** command, the **Replace All** command affects all the tabs.

New addresses reassignment using the wizard

The procedure below explains how to reassign a range of addresses using the import assistant wizard (only symbolic variables can be modified using the assistant wizard):

Step	Action
1	Select the **Variable** tab from the wizard window.
2	Select the first source address from the range of addresses to re-allocate.
3	Right-click on the first source address.
4	Select **Set Low Limit**. **Result:** In the **Address remapping** section, the low limit is set.
5	Select the last source address from the range of addresses to re-allocate.
6	Right-click the last source address.
7	Select **Set High Limit**. **Result:** In the **Address remapping** section, the high limit is set.
8	In the blank field from the **Address remapping** section: • Enter the first new remapping address.
9	Click **Remapping...**. **Result:** • The original range of addresses is updated. • The **Target Address** column is updated with the new range of addresses.

Variables tab of the wizard

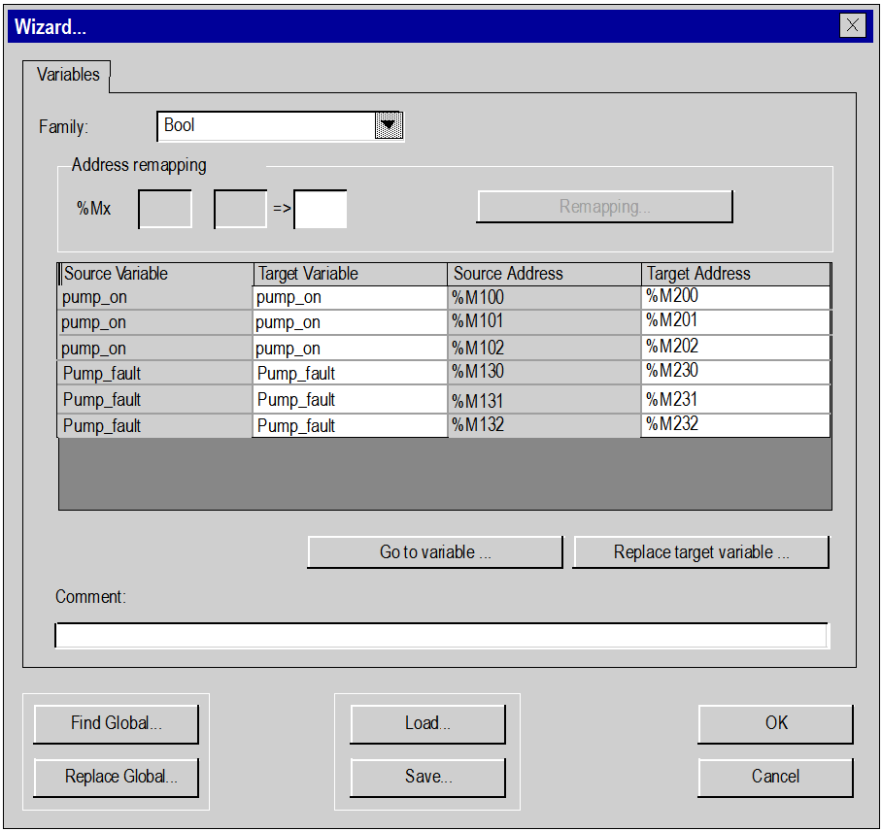
At a Glance

This tab allows you to edit the name of the symbols or to reassign an object or a group of objects to be imported.

If you want to import input/output variables, this editing allows you to reassign them in order to adapt to the new current configuration.

Reassignment of variables

The **Remapping** command is used to remap a group of variables. For example, you can remap the objects %M100...%M132 into %M200...%M232:

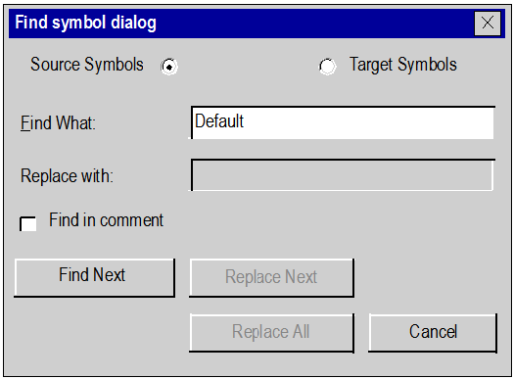


Sort by type

In the **Family** field, you can select the type of variables to display in the window.

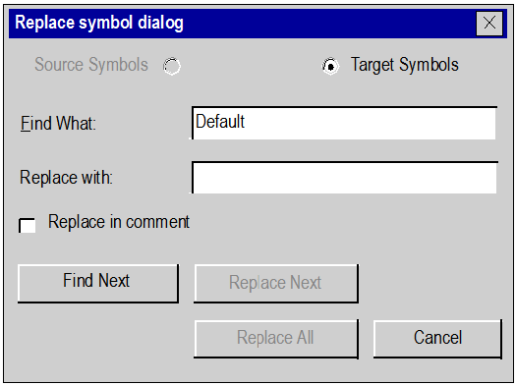
Modifying a symbol

The **Go to variable** command allows you to find a symbol or a character string in the symbols (source or destination):



If you select the **Find in comment** box, the variable is also searched in the variable comments.

The **Replace target variable** command allows you to modify a symbol or replace a character string in the symbols to be imported:



If you check the **Replace in comment** box, the variable is also replaced in the variable comments.

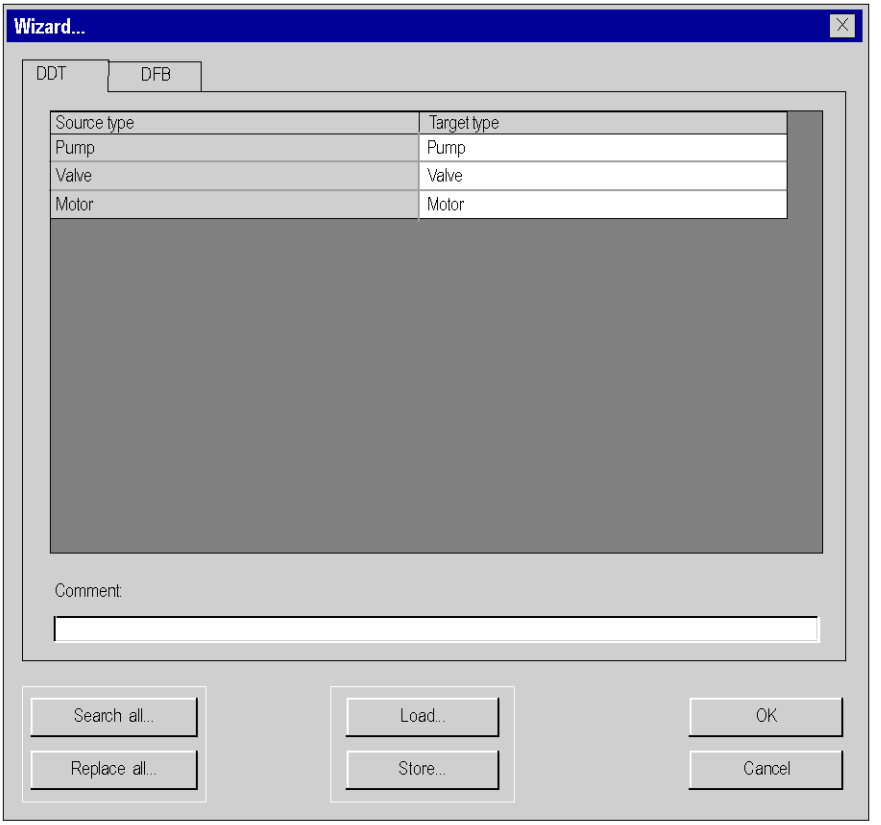
NOTE: The **Find What** command is case-sensitive. The special character * is not allowed in character strings and does not act as a wildcard character to search any string or part of string.

NOTE: The **Find symbol dialog** and **Replace symbol dialog** dialog boxes are also available using **Find global** and **Replace global** buttons.

DFB and DDT Tabs of the Wizard

At a Glance

These Two Tabs Present the DFB or DDT Types to Be Imported.



Modifying an Element Name

To modify the name of an element, you must perform the following operations:

Step	Action
1	Double click on the element to be modified.
2	Input the new name of the element.
3	Confirm your input by pressing the **Enter** button. The **Escape** button cancels the input.

Elements You May Modify

The various elements that you are able to modify are the following:

- DFB:
  - **Target type** enables you to modify the name (**Source type**) of a DFB type. This name may contain a maximum of 32 alphanumeric characters (‘A’...‘Z’, ‘0’...‘9’ and underscore character); the first character must be a letter.
  - The new name of a type must be unique.
  - The DFB type may not be modified.
  - **Comment** enables you to modify comment associated with a selected DFB type. This comment, written on a single line, may contain a maximum of 1024 characters.
- DDT:
  - **Target type** enables you to modify the name (**Source type**) of a DDT type. This name may contain a maximum of 32 alphanumeric characters (‘A’...‘Z’, ‘0’...‘9’ and underscore character); the first character must be a letter.
  - The new type names must be unique.
  - **Comment** enables you to modify comment associated with a selected DDT type. This comment, written on a single line, may contain a maximum of 1024 characters.

SR, Sections, Events and Macro steps tabs of the wizard

At a Glance

These four tabs present the sections, SR program modules, event processing elements and macro steps to be imported.

Modifying the name of an element

To modify the name of an element, you must perform the following operations:

Step	Action
1	Double click the element to be modified.
2	Enter the new name for the element. You can only modify the **name** of a program module or of a macro step and the **number** of an event processing element.
3	Confirm your input by pressing **Enter**. Press **Esc** to cancel the input.

Elements that you can modify

The various elements that you can modify are as follows:

- SR program sections and module
  - **Target Address** allows you to modify the name of an SR program section or module. This name comprises a maximum of 32 alphanumeric characters (‘A’...’Z’, ‘0’...’9’ and underscore character); the first character is a letter.
  - **Comment** allows you to modify the comment associated with the SR program section or module. This comment written on a single line may contain a maximum of 512 characters.
  - The new SR program section or module must be unique.
- Event processing
  - **Target Address** allows you to modify the event processing number.
  - **Comment** allows you to modify the comment associated with the event processing element. This comment written on a single line may contain a maximum of 512 characters.
  - The new event processing must be unique.
- Macro step
  - **Target Address of the Macro** allows you to modify the name of a macro step.
  - **Comment** allows you to modify the comment associated with the macro step. This comment written on a single line may contain a maximum of 512 characters.
  - The new macro step must be unique.

Wizard animation tables tab

At a Glance

This tab presents the animation tables to be imported.

Modifying the name of an element

To modify the name of an element, you must perform the following operations:

Step	Action
1	Double click the element to be modified.
2	Enter the new name for the element. You can only modify the **name** of an animation table.
3	Confirm your input by pressing **Enter**. Press **Esc** to cancel the input.

Elements that you can modify

The various elements that you can modify are as follows:

- Animation table
  - **Target address** allows you to modify the name of the animation table. This name comprises a maximum of 32 alphanumeric characters (‘A’...’Z’, ‘0’...’9’ and underscore character); the first character is a letter.
  - **Comment** is used to modify the comment associated with the animation table.
  - The new animation table must be unique.

Wizard functional module tabs

At a Glance

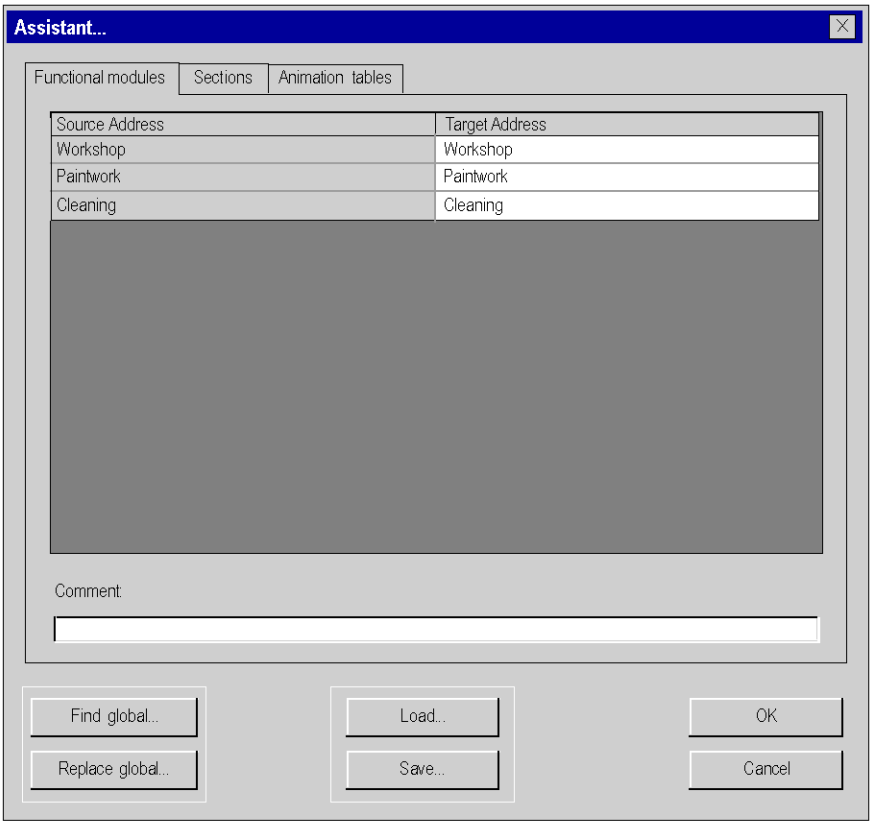
When importing a functional module with the wizard, the dialog box appears with the following tabs:

- Functional modules,
- Variables, page 1731,
- Sections, page 1734.
- Animation tables, page 1736,

- etc.

The tab of the element to be imported appears only if this element exists in the functional module.

The following illustration presents the dialog box when a functional module is imported:



Elements you can modify

The various elements you can modify are as follows:

- Functional module:
  - **Target Address** allows you to modify the target name of the functional module. This name comprises a maximum of 32 alphanumeric characters (‘A’...’Z’, ‘0’...’9’ and underscore character); the first character is a letter.
  - The new functional module name must be unique.
  - **Comment** is used to modify the comment associated with the selected functional module. This comment written on a single line may contain a maximum of 1024 characters.

Correspondence File

At a Glance

This file in text format contains all the information concerning the modifications or reassignments made in the various tabs of the wizard: Variables, DFB, DDT, Sections, Macro steps, etc.

NOTICE
LOSS OF DATA Save the Correspondence file before launching the import command. **Failure to follow these instructions can result in equipment damage.**

Saving/Restoring Modifications

The **Load** and **Save** commands allow you to:

Command	Description
Load	Allows you to automatically restore the reassignments previously saved to a file.
Save	Allows you to save the modifications made, via the wizard, to a correspondence file. The following dialog box allows you to enter the name of the file: 

Conversion Wizard

What's in This Part

Conversion Wizard..... 1742

Subject of this Part

This part describes the conversion wizard.

Conversion Wizard

What's in This Chapter

General Description of the Conversion Wizard 1742
Structure Tab of the Conversion Wizard 1748
I/O Remapping Tab of the Conversion Wizard 1752
CPU Configuration Tab of the Conversion Wizard 1770

Overview

This chapter describes the conversion wizard.

General Description of the Conversion Wizard

Overview

This section provides a general description of the conversion wizard and how to access the wizard.

Introduction

Overview

The conversion wizard is an integrated part of Control Expert.

The conversion wizard is used to:

- convert applications, exported out of legacy applications (PL7 and Concept) to Control Expert
- convert legacy applications partially or as a whole
- remap I/O objects (channels, variables etc.) during conversion by means of the wizard
- adapt the hardware configuration of the new application concurrently in Control Expert
- modify the amount of used memory in the CPU

The conversion wizard is available if you have chosen to install a converter (e.g. PL7 Application Converter) during the setup of Control Expert.

Application Conversion as a Whole

To convert an application as a whole, while keeping the same PLC family and selection of application parts or remapping of I/O objects is not needed, you have to use the specific converters directly via the Control Expert menu **File > Open**.

Additional documentation is provided for the specific converters (*EcoStruxure™ Control Expert, Concept Application Converter, User Manual* and *EcoStruxure™ Control Expert, PL7 Application Converter, User Manual*).

Partial Application Conversion

To convert an application partially and/or the PLC family must be changed or remapping of I/O objects is needed, use the conversion wizard via the Control Expert menu **Tools > Convert Partially**.

General Procedure

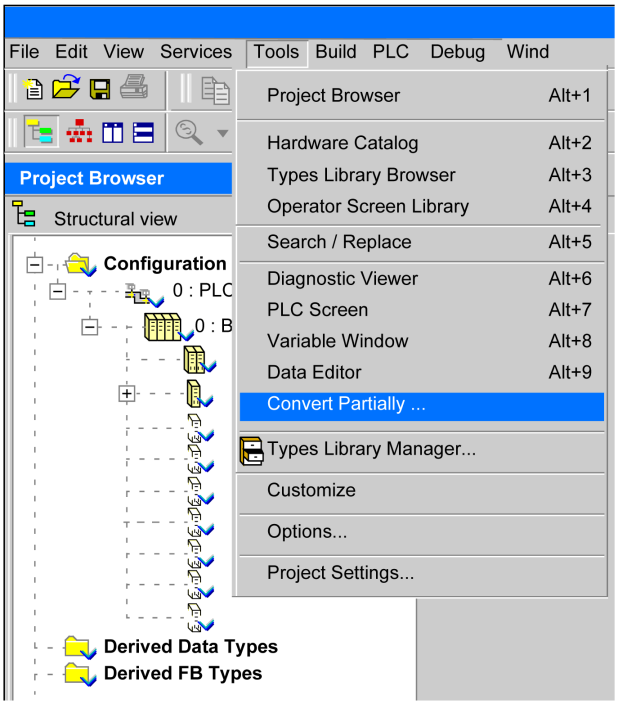
General procedure to convert a legacy application to Unit Pro

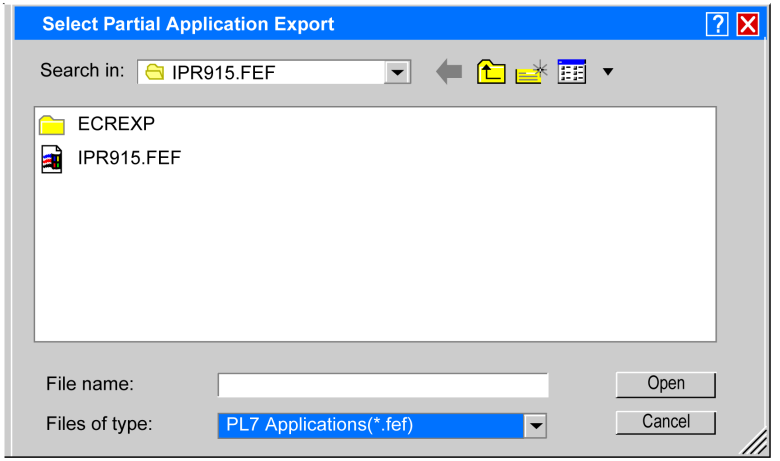
Step	Action
1	Export your application out of your legacy programming system (e.g. as an FEF file out of PL7 or an ASC file out of Concept).
2	Create a new application in Control Expert selecting a CPU with enough memory and the I/O access capabilities needed (e.g. Modicon M340). Optionally you can configure the I/O modules expected to be needed but you can modify the hardware configuration even later (see step 6).
3	Launch the conversion wizard in Control Expert via **Tools > Convert Partially**. **Result:** The conversion wizard asks you to select the exported legacy source file.
4	Select the exported legacy source file. **Result:** The converter analyzes the source file and displays the result in the 3 tabs of the conversion wizard.
5	Select the parts of the application (or the complete application) to be converted in the **Structure** tab.
6	Remap the I/O objects for getting them compliant with the new hardware configuration. Concurrently you can modify the hardware configuration of the new application in Control Expert. **Note:** To save a backup file of your intermediate I/O mapping you can use the **Save** button. With **Load** you can reload your latest saved intermediate I/O mapping.

Step	Action
7	After finishing all your selections and manual modifications click **OK**. **Result:** The converter applies the defined remapping to the selected parts of the source file and imports the results into the opened Control Expert Application (e.g. an application with an Modicon M340 PLC).
8	Continue working on the opened application, save it or export it as a ZEF or XEF file, page 1721.

Access to Conversion Wizard

Accessing the Wizard

Step	Action
1	Create a new application in Control Expert selecting a CPU with enough memory and the I/O access capabilities needed (e.g. Modicon M340).
2	Launch the conversion wizard in Control Expert via **Tools > Convert Partially**. 

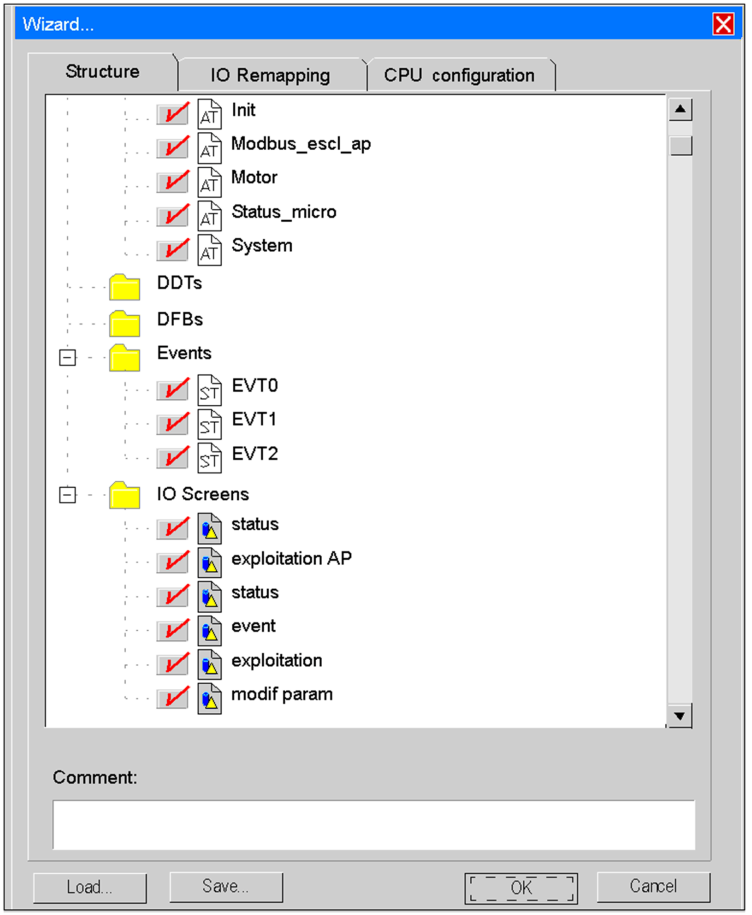
Step	Action
3	A file selection dialog is displayed, offering a choice of exported legacy application files (e.g FEF files for PL7, ASC files for Concept etc.). 
4	Select a file and confirm with **Open**. **Result:** Depending on the kind of file selected the respective converter (PL7, Concept, LL984) becomes active and analyzes the content of the selected file. The result of the analysis is displayed in the 3 tabs of the conversion wizard.

Exported Files

Software	Extension	Exported
PL7	**.fef*	complete application
Concept	**.asc*	complete application
Concept	**.sec*	user-defined subset of sections
Concept	**.txt*	user-defined subset of variables
Concept	**.ccf*	configuration export to extract communication definitions
ProWORX32	**.pwx*	complete application
MODSOFT	**.cfg*	complete application
Control Expert	**.xpg*	tasks (without configuration)

General Representation

Overview



Elements of the Wizard

Element	Description
tabs	The conversion wizard shows 3 tabs: • **Structure** In this tab you can select the parts of the application to be converted. Please refer to Structure Tab of the Conversion Wizard, page 1748. • **IO Remapping** In this tab you can remap I/O objects before conversion. Please refer to I/O Remapping Tab of the Conversion Wizard, page 1752. • **CPU configuration** In this tab you can modify the amount of used memory in the CPU. Please refer to CPU Configuration Tab of the Conversion Wizard, page 1770.
Save...	Saves all your input made in the different tabs, as selecting the parts to be converted or remapping of I/O objects, to a backup file. Each **Save...** will overwrite your last backup file.
Load...	Restores all input, made in the different tabs, from a backup file. There is only 1 level of backup file created with your last **Save...**.
OK	Starts conversion taking into account all the input made in the different tabs of the wizard.
Cancel	Cancels conversion and closes the wizard.

The content of the tabs may differ depending on the source code that is converted.

Conversion and Result

Starting Conversion

Click the **OK** button to start converting the already prepared application.

XPG File

An XPG file containing all user corrections is saved during conversion.

- In case of PL7 conversion an XPG file is saved to a user-related directory:

  My Documents\PL7 Converter\<app name>\<app name.xpg>

  This directory is different from the directory where the input file has been taken from. This allows to read also from network directories with write-protection.

- In case of Concept and LL984 conversion an XPG file is saved to the same directory as where the source file is located.

  NOTE: XPG format is used as output file format because it contains programs only, but no configuration.

Automatic Import

The import of the converted application into the opened application is automatically launched to free you from an otherwise additional operation step.

Report File

A report of conversion that gives the list of new mapping, warnings and what is not converted, accompanies the XPG output file in the same user-related directory.

Application Build

To be able to build the application in Control Expert correct the I/O mapping completely and remove all other errors displayed during analysis of the application.

Structure Tab of the Conversion Wizard

Overview

This section provides a description of the conversion wizard´s **Structure** tab and how to select the parts of the application to be converted.

Introduction

Overview

The **Structure** tab of the conversion wizard shows the analyzed legacy application in a browser view.

In this browser you can select all parts of an application that should be converted.

Selectable Parts

The following parts of an application can be selected for conversion:

- program tasks and sections of programs in all programming languages, subset list of sections
- subset list of subroutines
- subset list of animation tables
- subset list of DDTs and DFBs
- subset list of events
- subset list of I/O screen families and I/O screens (operator screens)
- variables

Depending on the legacy project, not all parts may exist.

NOTE: By default all parts of an application are selected for conversion.

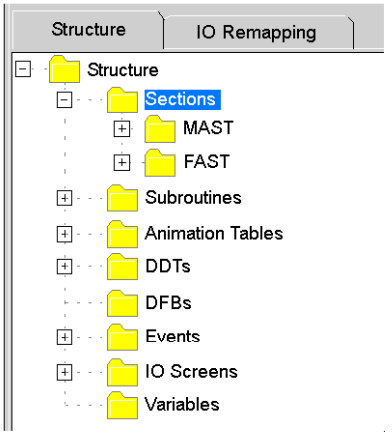
I/O Screens (operator screens) can not be selected.

Variables

If you select the **Variables** part for conversion, all the variables that are declared data (having a symbol or a comment) are taken into account during conversion.

Otherwise only data, called inside the selected sections, subroutines and events, are converted.

Browser View



The extension sign (+) in front of a directory indicates that there are selectable subdirectories.

Selecting Application Parts to be Converted

Overview

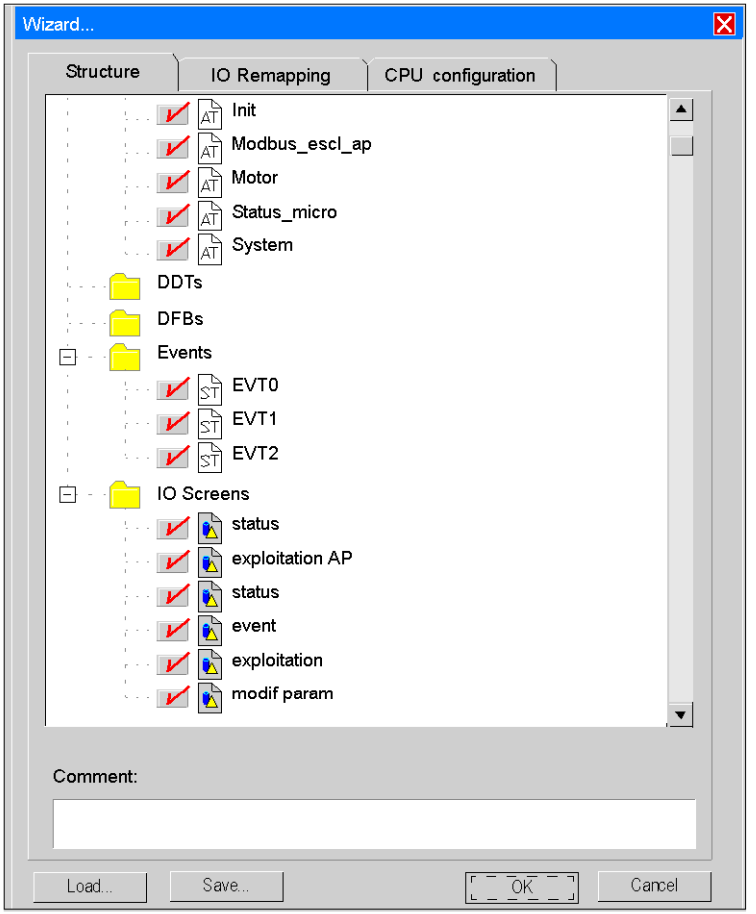
The **Structure** tab of the conversion wizard shows the analyzed legacy application in a browser view.

In the browser view you can select the parts of an application to be converted.

You can either select single parts or all parts of a directory at once.

You can not select a directory itself.

Structure Tab



Elements of the Tab

Element	Description
checkbox	A checkbox with a red checkmark indicates that this part should be converted. An empty checkbox indicates that this part should not be converted. Click the checkbox or the icon beside to toggle from checked to unchecked and vice versa.
Comment	Displays the comment of a selected part in the browser view (read only).

Context Menu

Right click on an application part in the browser view to open a context menu.

This context menu provides the **Select All** and the **Select None** items.

Element	Description
Select All	Click on any part of a directory and then click **Select All** to select all parts of the respective directory.
Select None	Click on any part of a directory and then click **Select None** to select all parts of the respective directory.

I/O Remapping Tab of the Conversion Wizard

Overview

This section provides a description of the conversion wizard´s **I/O Remapping** tab and how to remap I/O objects before conversion.

Introduction

Overview

The **I/O Remapping** tab of the conversion wizard shows the I/O objects of an analyzed legacy application in a table view.

By means of this table view you can remap I/O objects before conversion.

The **I/O Remapping** tab provides additional features to ease special remapping operations (**Communication -1**, **Exclude**, **Type Restricted**).

The **I/O Remapping** tab is applicable for all applications coming from PL7, Concept or ProWORX32.

Syntax of I/O Objects

For the detailed syntax of I/O objects and their correspondences in Control Expert please refer to the specific documentation to be found under:

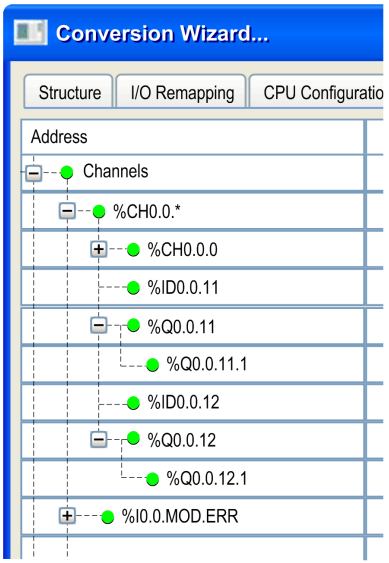
- For Concept Converter, refer to chapter *System* (see EcoStruxure™ Control Expert, Concept Application Converter, User Manual).
- For PL7 Converter, refer to chapter *Correspondences between PL7 and Unity Pro: language objects* (see EcoStruxure™ Control Expert, PL7 Application Converter, User Manual).

Address Column

Directly addressed I/O variables (topological addresses) are represented in the **Address** column of the conversion wizard as shown in the following example.

Please also refer to chapter *Topological Addresses of PL7 Micro in the Conversion Wizard* (see EcoStruxure™ Control Expert, PL7 Application Converter, User Manual).

Embedded counter channels of a PL7 TSX Micro CPU



For detailed address description please refer to chapter *Direct Addressing Data Instances* (see EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual).

Corrected Column

The addresses in the **I/O Remapping** tab are formally adapted to Control Expert syntax, but have the same numerical values as before in PL7.

For details please refer to chapter *Memory objects (variables and constants)* (see EcoStruxure™ Control Expert, PL7 Application Converter, User Manual).

In the **Corrected** column only direct addresses following the Control Expert syntax may appear.

If modules do not exist in Control Expert, existing I/O objects have to be mapped manually to other regular modules of the targeted PLC.

I/O Object Remapping

I/O Remapping is applicable to all selectable source formats and PLC families coming from PL7, Concept or ProWORX32.

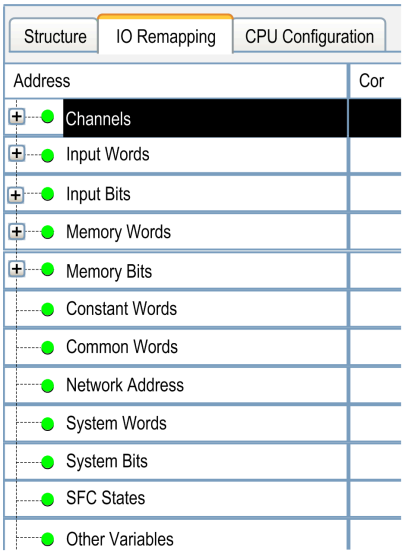
Variable Families

Because applications can have very many variables, the variables are structured into families.

A variable family is characterized by direct addresses of the same semantically meaning, e. g. Word addresses or addresses of boolean variables.

Table View

The names of all variable families are listed in the table view.



If there is no extension sign (+) in front of a variable family, no object of this family is present in the application.

Remapping Table

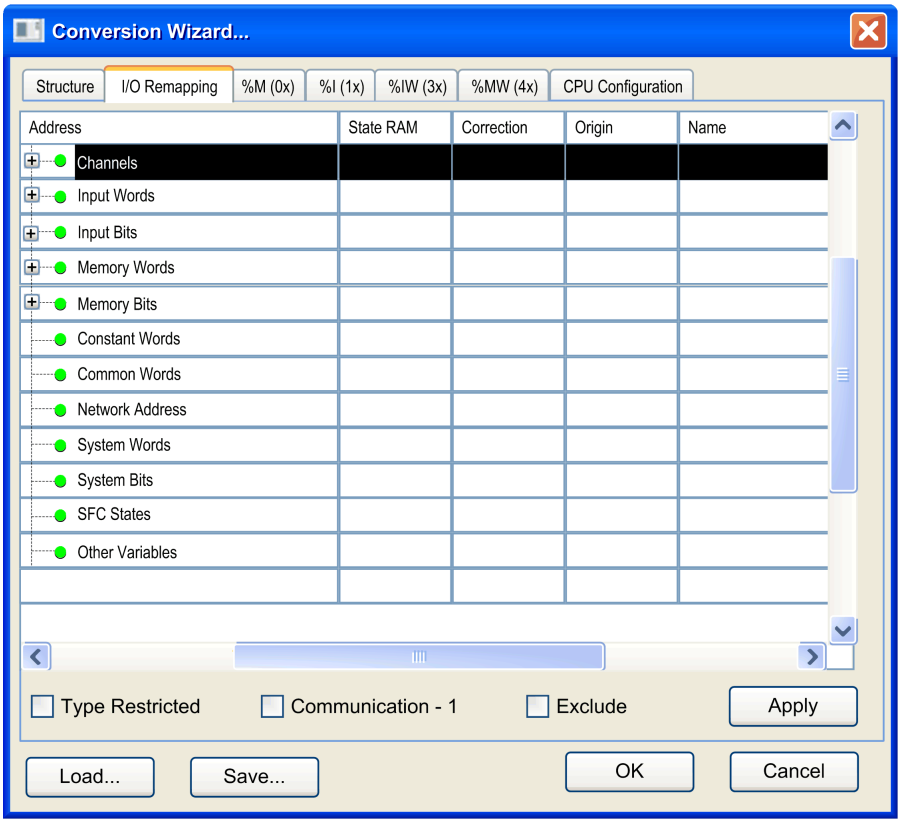
Overview

The remapping table shows all variables with direct addresses used in the application whether they are declared or not.

Even if directly addressed variables appear only in programs, they are detected and displayed in this table.

Representation

I/O Remapping tab



Elements of the Remapping Table

To see all columns of the table scroll horizontally.

Element	Description
Address	Topological or direct variable address For detailed information please refer to paragraph below.
State RAM	This column is only available for Concept, Modsoft, ProWORX32 or ProWORX Nxt applications with State RAM assignments. If the **State RAM** column is visible, the **Correction** column only accepts State RAM addresses.

Element	Description
Correction	In this column you can enter corrections for the values of the **Address** column. Use the same format as in **Address** column. Example: **Address:** `%I\1.1\1.4.1` **-Correction:** `%I\1.1\1.3.1`
Origin	Source module of the imported legacy application
Section	A comma-separated list of section names in which the variable is used or the description of the hardware module, if available in the application file.
Name	Symbolic name
Type	Variable type
Comment	Variable comment
Apply	Starts the defined remapping for the currently selected variables.
Communication -1	Please refer toExclude, Communication -1, Type Restriction, page 1766.
Exclude	
Type Restricted	

Address

For data that have hierarchical elements, like channels for hardware modules in TSX Premium and TSX Micro, there is an extension sign (+) to expand the list. For example, an overview of the channels of the hardware modules is displayed.

The **Address** column shows the hierarchy of the topological addresses in tree form. The rows are sorted according to the module addresses (e.g. `%CH0.1*`), which are shown as:

`%CH{\<Bus alias>.<Connection point>\}<Rack>.<Module>.*`

The module addresses are derived from the hardware modules present in the configuration part of the application file.

The addresses are composed with a wildcard character "`.*`". The wildcard character stands for all channels of the hardware module (e.g. `%CH0.1*`).

In the next level (e.g. `%CH0.6.1.*`), that can be expanded by clicking the extension sign (+) in front of a module row, the channels of the module are listed.

This is done by additionally showing the channel number, followed by a wildcard character "`.*`". The wildcard character stands here for the various data of the channel, which can have more than 1 level.

Below the **Channels** item, there are the items for **Input** and **Memory** that are both subdivided into **Words** and **Bits**. These 4 groups directly represent the State RAM of the PLC. They only show those variables that are used in the entire program. Unused I/O variables are not displayed here.

The 2 **Memory** groups contain clusters of memory items with consecutive addresses. In these groups it is allowed to rearrange consecutive areas. This is not allowed in the 2 **Input** groups.

Correction

You can define new start addresses for the memory items in the **Memory Words** and **Memory Bits** groups. The new addresses are displayed in the **Correction** column of the **I/O Remapping** tab. For further information on the remapping process, refer to the chapter Clustering and Remapping of Memory Variables, page 1761.

Section

The **Section** column shows a comma-separated list of section names in which the variable is used or the description of the hardware module, if available in the application file.

Remapping Channels, Groups of Channels or Specific I/O Types

Overview

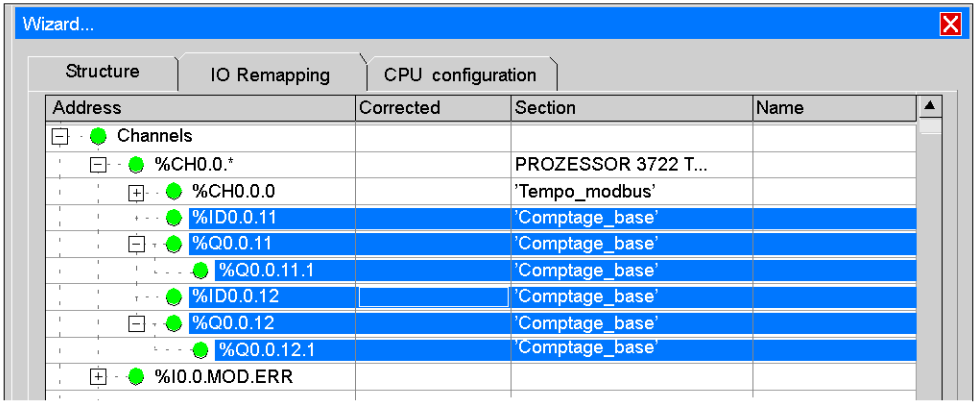
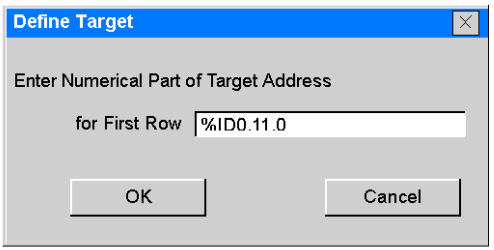
There are several kinds of objects that are managed in hardware modules, for example %IW, %QW, %MW, %I, %Q, etc.

By means of the conversion wizard you are able to

- move all objects from one slot to another slot.

  This is equivalent to search any kind of objects with e.g. addresses `\4.0\2.*` and replace in all kind of objects with `\5.2\3.*`.
- move a subset of channels from one location to another one.

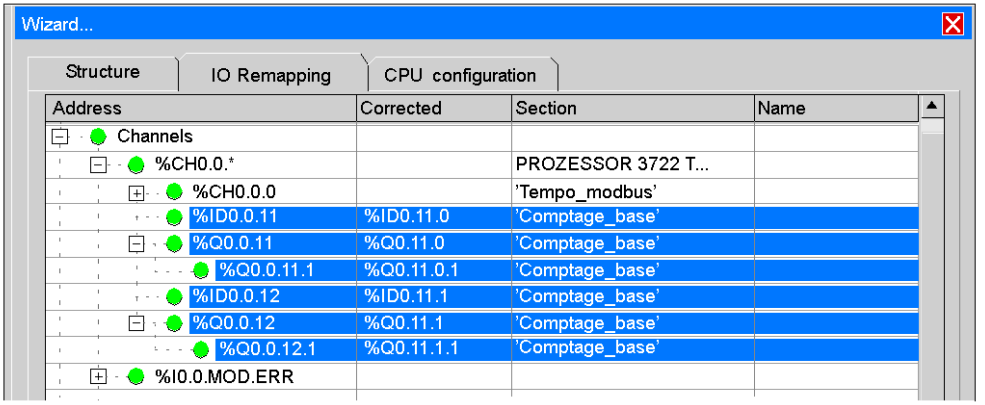
Procedure

Step	Action
1	Select the rows of the data to be remapped either at the channels level or in levels below. 
2	Right click anywhere in the selected area to open the context menu. 
3	Select **Define Target** from the context menu. The **Define Target** dialog opens. 

Step	Action
4	Enter the address that should be assigned to the first selected row in the remapping table.
5	Click **OK** to confirm. **Result:** The selected rows are remapped incrementally starting from the new address entered for the first selected row. Even addresses in levels below are remapped. The new addresses are displayed in the **Corrected** column. See example below.

Example

Embedded counter channels of a PL7 TSX Micro CPU moved to a counting module of a Modicon M340



Validity Check

There is no validity check of the new mapping with the opened application during editing.

The check of consistency is done after converting the application, during the analysis of the application in Control Expert.

Clustering and Remapping of Memory Variables

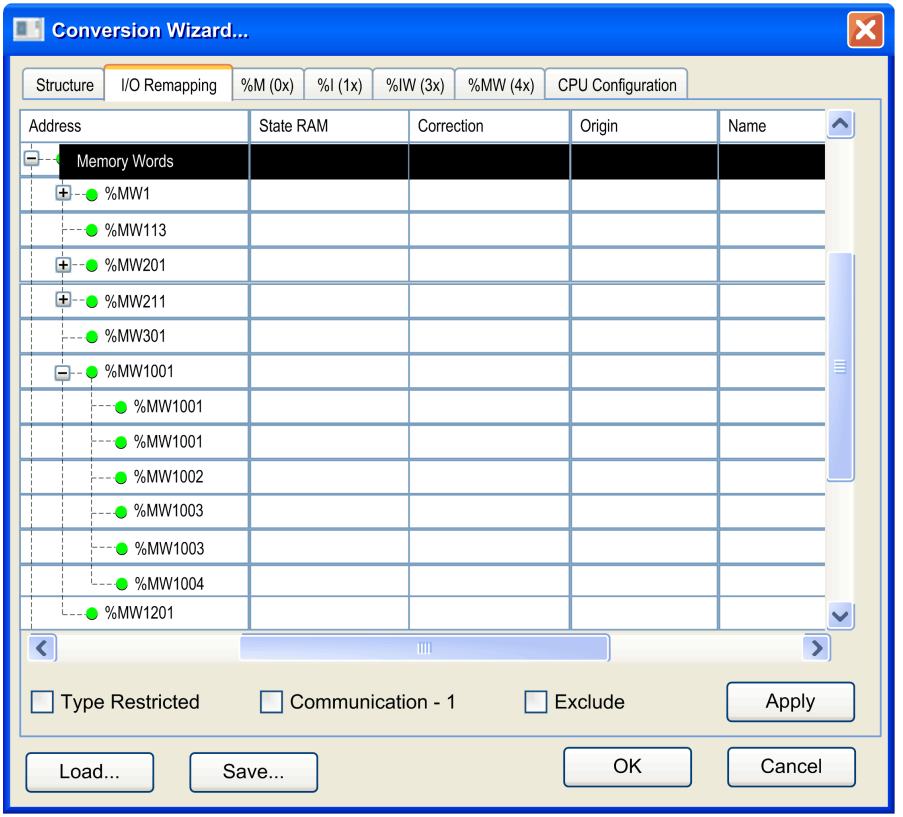
Overview

Not only topological addressed I/O objects, but also directly addressed memory is structured similarly into clusters.

A cluster is a sequence of memory cells, in which immediate neighbor cells with an address difference of 1 are collected.

Representation

The graphic below shows a cluster example with `%MW1001`.



Remapping

Remapping can be applied to channels and such clusters.

Subsequent sets of clusters, subsequent sets of elements on lower levels, single clusters or single elements can be selected and remapped by invoking the command from the context menu. The new address will be displayed in the **Correction** column.

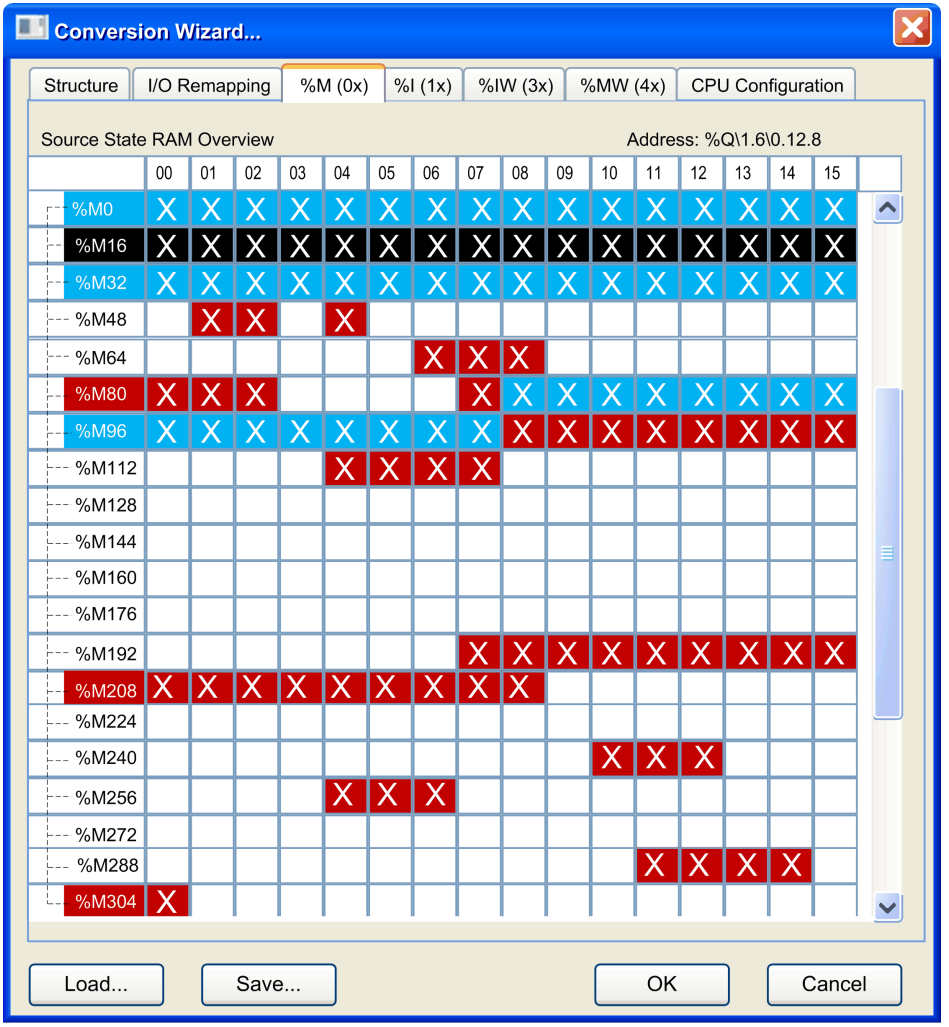
Alignment of variables with type `DINT`, `UDINT` and `REAL` is automatically performed, if a Modicon M340 configuration has been initially defined.

Memory Area Tabs of the Conversion Wizard

Overview

The memory area tabs (**%M (0x) to %MW (4x)**) are only available for Concept, Modsoft, ProWORX32 or ProWORX Nxt applications with State RAM assignments. They show the variable allocation in the State RAM of the source application.

Memory Area Tabs (%M (0x) to %MW (4x))



The memory area tabs (**%M (0x) to %MW (4x)**) indicate the variable allocation of the State RAM in the source application.

The elements of the dialog are color-coded as follows:

Element	**Color**
module allocated variables	blue
language allocated variables	red

In this dialog, module allocated variables are dominant. This means that only module allocation is indicated in the dialog, even if the variable is language accessed.

Select a cell in the table to indicate the topological address of the variable in the upper right corner of the dialog.

To display different memory areas, select the respective memory area tab (**%M (0x), %I (1x), %IW (3x), %MW (4x)**).

Concurrent Editing

Overview

The **Conversion Wizard** is a modeless window, i.e. you can work either in the **Conversion Wizard** or for example in the Control Expert hardware configuration window.

Therefore you can adapt the remapping and modify the hardware configuration concurrently.

- You can adapt the default mapping done by the converter according to the real hardware.
- You can concurrently modify the configuration of the new Control Expert hardware.
- You can adapt the hardware configuration to the legacy application, while remapping I/O objects according to the modularity of the modules that may be different to legacy hardware.

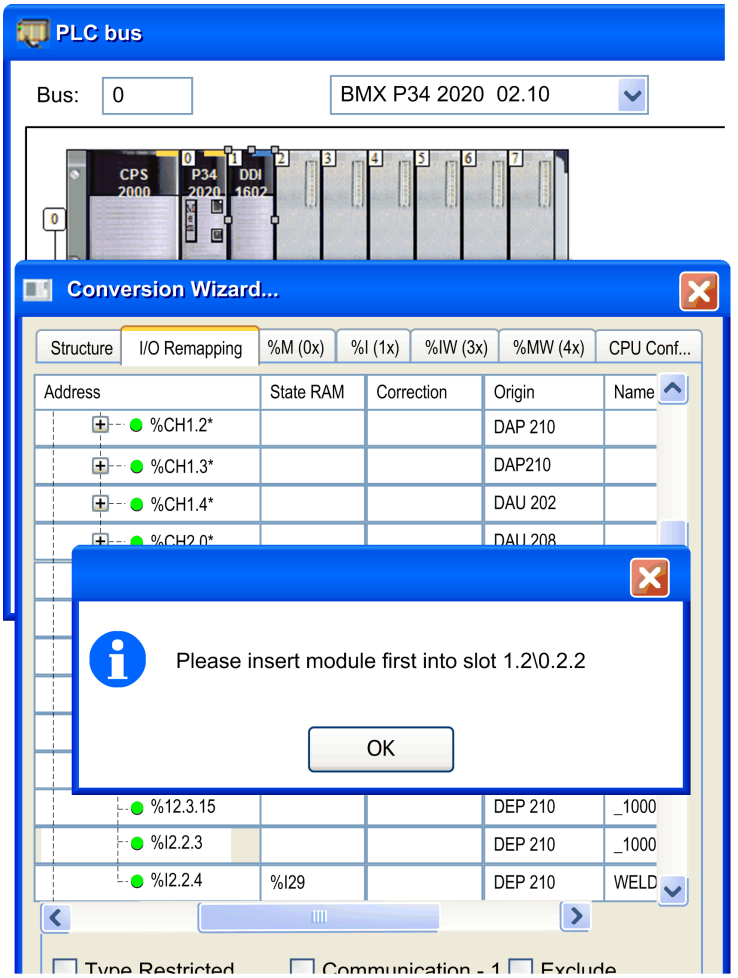
Initial Situation

A new Modicon M340 configuration appears in the Control Expert configuration editor.

The application of which parts have to be added is opened in the **Conversion Wizard**.

The I/O channels used in the application appear in the **I/O Remapping** tab.

Representation



Interactive I/O Remapping

Step	Action
1	Select a channel group representing a module, e.g. CH1.1.*
2	Enter the corresponding target module into the configuration editor.

Step	Action
3	Open the target address dialog for the chosen channel group.
4	Enter the new channel assignment.

NOTE: It is required to enter a module to which a correction order refers into configuration before giving the referring correction command.

The **Conversion Wizard** checks the Control Expert configuration for the presence of the module - if it is not there, the correction is not executed, leaving the target field blank.

If the module is present in the configuration, the **Conversion Wizard** reads out its State RAM assignments done by the Control Expert configuration and replaces the assignments in the State RAM column by the new addresses.

Internally it keeps the old addresses and forms correction commands to the Control Expert-internal correction processor, which replaces all occurrences of the old addresses with the new addresses, when acknowledgement is given with the **OK** button.

Corrections for all modules can be collected before acknowledging, giving the ability to correct also previous corrections.

No Automatic Notification

Even though you can adapt the remapping and modify the hardware configuration concurrently, there is no automatic notification between hardware configuration and **Conversion Wizard** about changes on either side.

NOTE: Following the conversion of the project, some adjustments may be needed depending on the differences of behavior between the original hardware module and the selected module in the Control Expert platform.

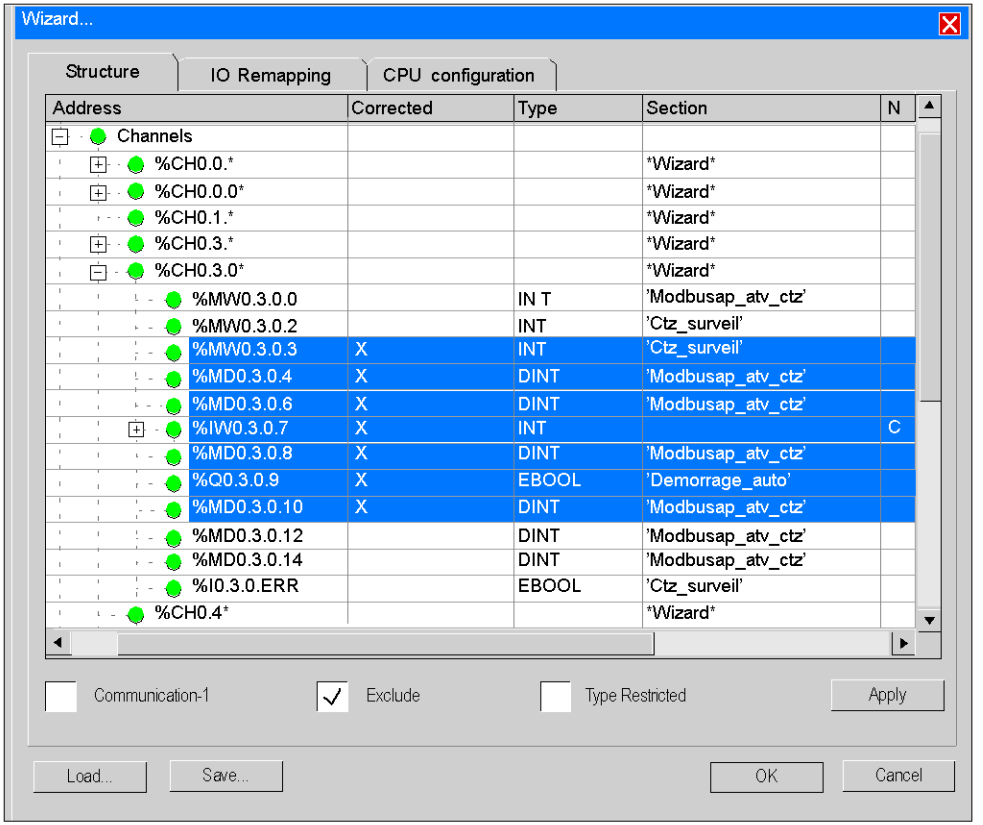
Exclude, Communication -1, Type Restriction

Overview

The **IO Remapping** tab provides additional features to ease special remapping operations (**Communication -1**, **Exclude**, **Type Restricted**).

These operations can be executed by checking the appropriate checkboxes below the remapping table.

Representation



Communication -1

Premium or Modicon M340 memory is indexed starting with 0, whereas Compact or Quantum memory indexing starts with 1.

If an application for the Compact or Quantum family is converted partially into a Premium or Modicon M340 family application, and if this PLC must communicate through the Modbus protocol with another PLC, it will be necessary to adapt the addresses during the conversion to recover consistent exchanges with other PLCs.

The communication software introduces this offset.

To subtract a 1 of all selected addresses, which must be determined to be Modbus communication related, activate the **Communication -1** checkbox.

Exclude

To exclude I/O objects from conversion proceed as described below.

Step	Action
1	Select the I/O objects in the remapping table you want to exclude from conversion.
2	Activate the **Exclude** checkbox.
3	Click the **Apply** button. **Result:** An X character is displayed in the **Corrected** column of the I/O objects excluded from conversion.

Type Restricted

In some cases not all information of a channel has to be remapped commonly.

To remap variables that match to one type within a selection only, activate the **Type Restricted** checkbox. See example below.

Type restriction applies to all kind of directly addressed variables.

%M internal variables

Variable	Type
%M	bit
%MW	word
%MD (for Premium only)	double word
%MF (for Premium only)	real (floating point)

%K constants

Variable	Type
%KW	word constant
%KD (for Premium only)	double word constant
%KF (for Premium only)	real (floating point) constant

%I inputs

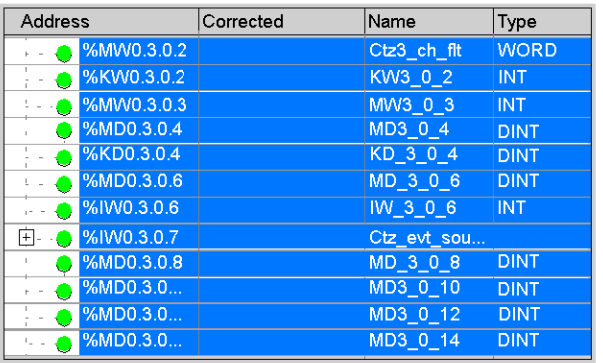
Variable	Type
%I	BOOL or EBOOL type bit
%IW	INT type word
%ID (for Premium only)	DINT type double word
%IF (for Premium only)	REAL type real (floating point)

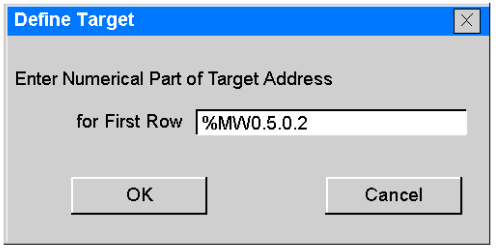
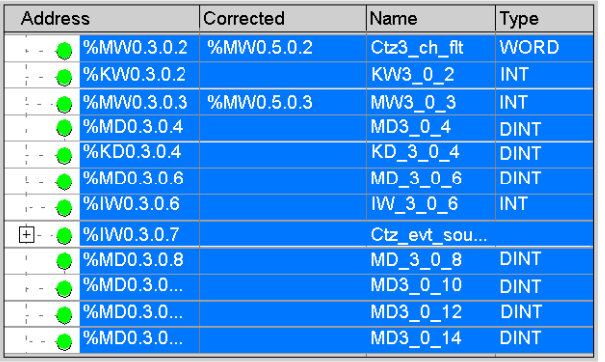
%Q outputs

Variable	Type
%Q	EBOOL type bit
%QW	INT type word
%QD (for Premium only)	DINT type double word
%QF (for Premium only)	REAL type real (floating point)

Type Restricted Example

For example the **Type Restricted** feature allows to remap all %MW variables at once to a module in the target family.

Step	Action
1	Select the variables to be remapped. 
2	Activate the **Type Restricted** checkbox.

Step	Action
3	Select **Define Target** from the context menu and enter the new address for the first variable to be remapped. 
4	All addresses, which have the same type (e.g. %MW) as the first one selected, are remapped. 

Address	Corrected	Name	Type
%MW0.3.0.2	%MW0.5.0.2	Ctz3_ch_flt	WORD
%KW0.3.0.2		KW3_0_2	INT
%MW0.3.0.3	%MW0.5.0.3	MW3_0_3	INT
%MD0.3.0.4		MD3_0_4	DINT
%KD0.3.0.4		KD_3_0_4	DINT
%MD0.3.0.6		MD_3_0_6	DINT
%IW0.3.0.6		IW_3_0_6	INT
%IW0.3.0.7		Ctz_evt_sou...	
%MD0.3.0.8		MD_3_0_8	DINT
%MD0.3.0...		MD3_0_10	DINT
%MD0.3.0...		MD3_0_12	DINT
%MD0.3.0...		MD3_0_14	DINT

CPU Configuration Tab of the Conversion Wizard

Overview

This section provides a description of the conversion wizard´s **CPU Configuration** tab and how modify the amount of used memory in the CPU.

Introduction

Overview

The **CPU Configuration** tab of the conversion wizard allows to modify the amount of used memory in the CPU.

The merge of corrected values and original values is put directly to the opened application.

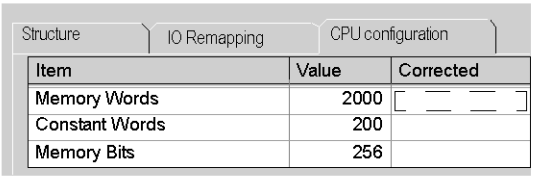
If there is a corrected value, it is used; otherwise the original value is taken.

The limits are check after converting the application, during the analysis of the application in Control Expert only.

The **CPU Configuration** tab is available for all PLC families with the exception that the input for **Constant Words** is not available for the Quantum family.

CPU Configuration Tab

To modify the values type the new values directly into the respective rows of the **Corrected** column.



Appendices

What's in This Part

Hyperlinks in Control Expert 1774
User Interface 1825
Keyboard Shortcuts 1857

Subject of this Part

This part presents the hyperlink functions, the ergonomics, and the shortcuts of Control Expert.

Hyperlinks in Control Expert

What's in This Chapter

General 1774
Principles 1777
Hyperlinks in the project browser 1786
Hyperlink in the data editor 1790
Hyperlinks in the DFB type attributes 1796
Hyperlinks in animation tables 1804
Hyperlinks in runtime screens 1808
Hyperlinks in the documentation folder 1814
Hyperlinks and drag-and-drop 1819

Subject of this Chapter

This chapter describes how to use hyperlinks (hypertext) in Control Expert.

General

Object of this sub-section

This sub-section provides general information about hyperlinks.

Introduction to Hyperlinks

General

You can use the hyperlink function available in the Control Expert software to create links between your project and external documents.

Hyperlinks are shortcuts to such documents as:

- installation and wiring diagrams,
- help screens,
- miscellaneous notes,
- images,

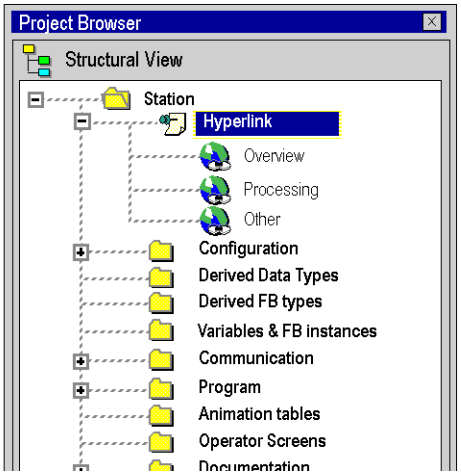
- web pages,
- etc.

For example, in an operator screen, you can associate an alarm that is displayed on a variable controller to a hyperlink that opens a document explaining the alarm's characteristics.

Hyperlinks and Project Browser

You can create hyperlinks in the project browser. They will be managed like shortcut sheets in a directory.

Example:



NOTE: To drop a hyperlink in a blank user directory, you must press the ‘shift’ key during drag-and-drop.

Hyperlinks and Comments

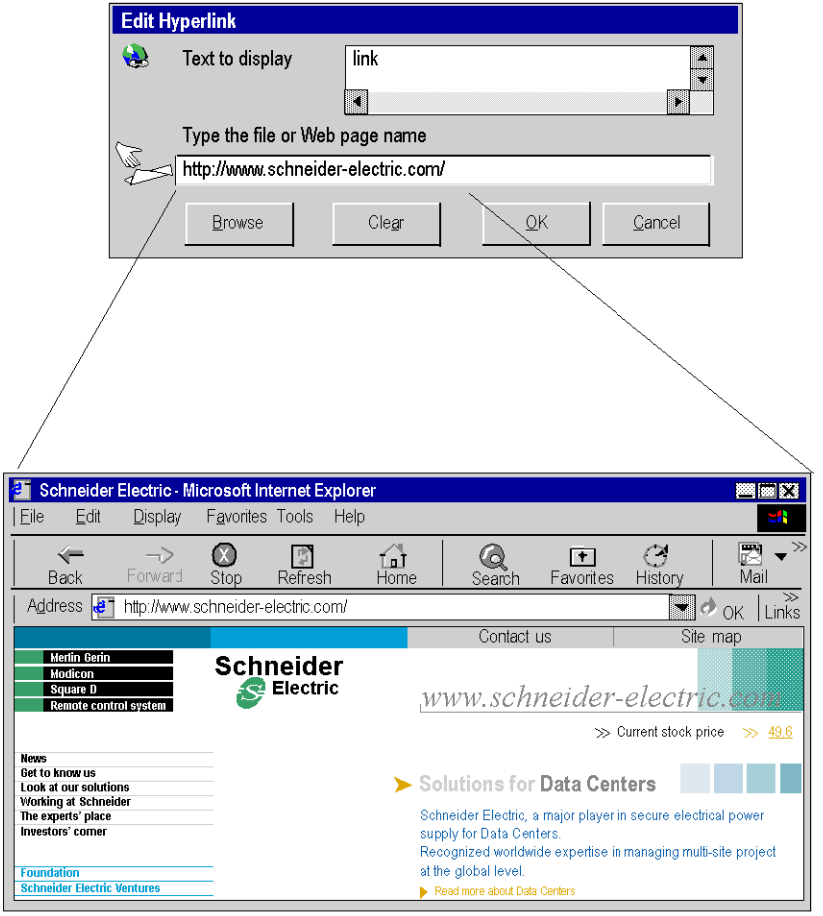
You can also create hyperlinks from a comment field. A hyperlink in a comment is used to extend the comment to other data outside the controller project which can take on all forms to represent the information.

In the Control Expert functions below, you can create a hyperlink in a comment belonging to:

- data editor variables,
- the animation tables,
- the operator screens,

- documentation.

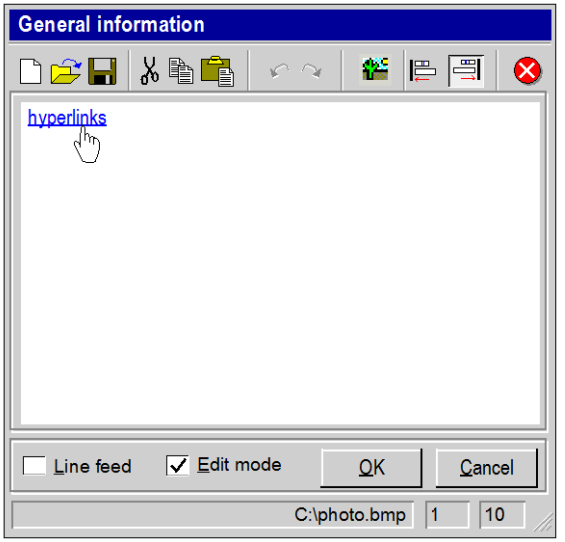
Example:



Representation and opening of a hyperlink

Representation

A hyperlink is represented by an underlined text in blue. When you move the mouse cursor over the hyperlink, it changes into a hand.



Opening a link

Right-click on the underlined part to open the linked document.

Principles

Object of this sub-section

This sub-section presents the main operate-modes and characteristics of hyperlinks.

Definitions

Target document

You CANopen the target document by clicking on the hyperlink. There are no restrictions on the type of document that can be opened. The system simply has to know the document's suffix. In other words the application at the origin of the creation of the document or which can view must exists and be available on the computer.

Hyperlink target

The hyperlink target is the tool (software) that is used to open the document. For example, if the document has a .doc extension, the hyperlink target is Microsoft Word.

The table below presents the different hyperlink targets depending on the file extensions. This list is not exhaustive.

File extensions	Hyperlink target
.doc	Microsoft Word
.xls	Microsoft Excel
.txt	NotePad
.htm .html .asp .shtml	Microsoft Internet Explorer Netscape WEB browser
.bmp .jpg .gif	Paint Paint shop
.png	Microsoft Photo Editor

Overview of operate-modes

Operate-modes

There are two operate-modes for creating and accessing a hyperlink:

- 1-step, page 1778mode,
- 2-step, page 1780mode.

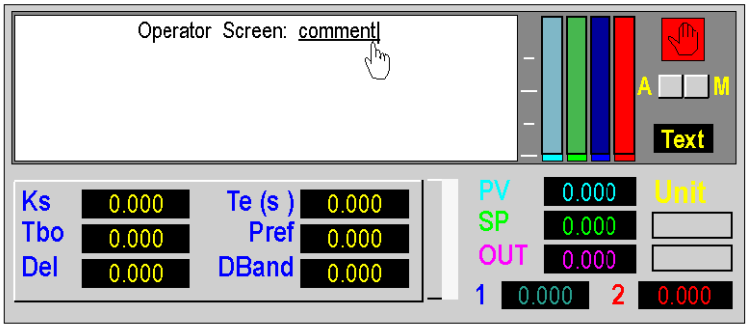
Each of these modes is applied to specific functions of the Control Expert application.

1-step mode

Principle

The 1-step mode is the mode that is available for most Control Expert functions.

It consists of associating a hyperlink with a comment. The principle of the 1-step mode is that the target document can be accessed directly with a single click from the comment. Simply click on the underlined comment to open the linked document.



Available functions

You can create a hyperlink in comments and objects that belong to the following functions:

- In the project browser:
  - directories and sub-directories of the browser,
  - hyperlink directory.
- In the data editor on:
  - elementary variable (EDT) comment,
  - function block (DFB/FB) instance comment,
  - derived variable (DDT) comment.
- In animation tables on:
  - variable (EDT/DDT/FB) comment.
- In operator screens on:
  - text graphical object,
  - animated text objects with a message.

2-step mode

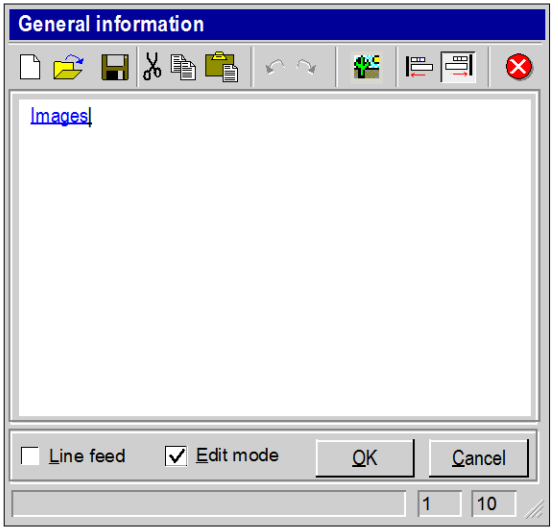
Principle

This mode is described as 2-step because you cannot access the target document directly. An intermediary window opens (comment editor) before you can access the hyperlink.

NOTE: In 2-step mode, you can create several hyperlinks in the comment editor. You can therefore have different target documents with one hyperlink source

Accessing the 2-step mode

The table below describes how to access the 2-step mode:

Step	Action
1	Use the project browser to select the **General Information** sub-directory of the **Documentation** directory.
2	Select **Open** in the contextual menu. **Result**: The **General Information**, page 1781 window appears. 
3	Click on the underlined comment in blue to open the linked document.

Available functions

You can create a hyperlink with the 2-step mode in comments and objects that belong to the following functions:

- In the project folder:
  - general information about the folder.
- In DFB type attributes.

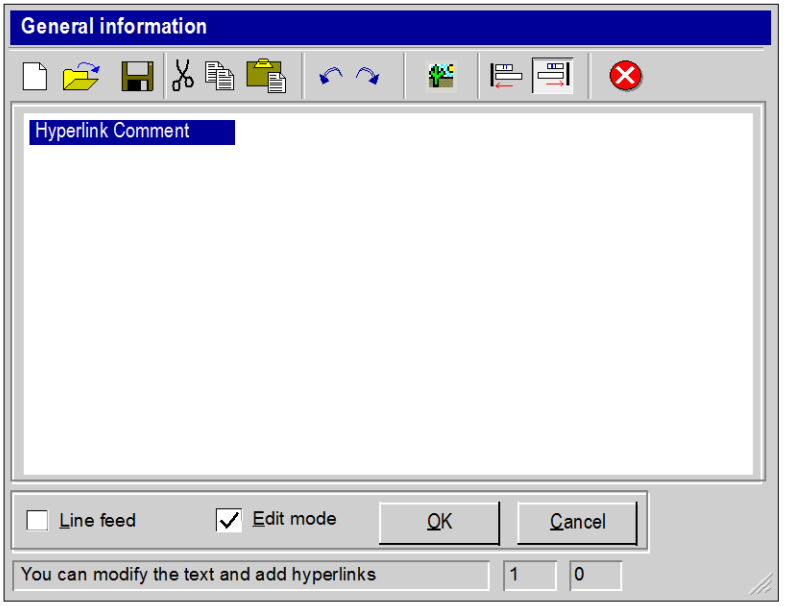
Description of the Hyperlink Comment Window

Illustration

In 2-step mode, the **General Information** or **DFB type description** window is used to enter the comment on which the hyperlink will be applied.

Illustration

The functions of the icons and elements that make up the window are described in the paragraphs below.



Toolbar

The toolbar is made up of different icons.



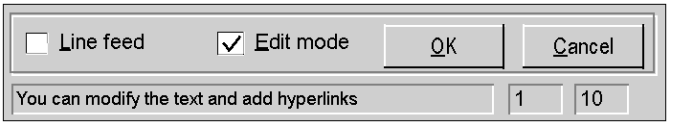
The table below presents the various functions of the icons.

Icon	Shortcut	Description
	Ctrl + N	Creates a new comment by deleting the comment that has already been written.
	Ctrl + O	Opens the dialog box for opening a .htm file to retrieve a hyperlink text saved on the hard disk in a HTML page outside the controller project. **Note**: You can in this way use a hyperlink that has already been created.
	Ctrl + S	Saves the comment and the hyperlink(s) to the hard disk in a HTML page outside the controller project. **Note**: You can in this way save a hyperlink to use again later on. This function is optional. When you save the controller project, the hyperlink comments are also saved in the .STU project. You therefore do not need to save a specific hyperlink comment in an .html page, except if you wish to use it again later on. The hyperlink comments are also saved in the STA archive files.
	Ctrl + X	Cuts the selected text.
	Ctrl + C	Copies the selected text.
	Ctrl + V	Pastes the cut or copied text.
	Ctrl + Z	Undoes the last action.
		Repeats the action made before it was canceled.
	Ctrl + K	Opens the **Insert hyperlink** dialog box.

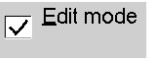
Icon	Shortcut	Description
		Aligns the text of the status bar to the left.
		Aligns the text of the status bar to the right. **Note**: When you move the mouse over a hyperlink without clicking on it, the full path name of the document associated with the hyperlink is displayed in the status bar. However, if the path name is too long, the status bar will not be large enough to read the name of the document. The right alignment function can be useful if you want to view the name of the document associated with a hyperlink.
		Cancels and closes the window.

Status Bar

The status bar is made up of different elements.



The table below presents the various functions of these elements.

Element	Description
	Automatic line skip with a long comment. Check to validate this function.
	Allows the comment to be entered. If you don't check this box, you will not be able to write or modify the comment. Check to validate this function.
	Confirm the editing of the comment.
	Cancel.

Element	Description
You can modify the text and add hyperlinks Explains the description of checked boxes.	
1 0	Shows the position of the cursor in the comment entry field (row - column).

Properties

Storage

Comments with hyperlinks, but not the linked documents, are stored in the controller. The linked documents may be stored on the hard disk of the programming station, on the Internet, on an operator terminal or other medium.

Operating modes

In the variable editor and folder, you can access the hyperlink in online and offline modes using the PLC.

You can modify a link while connected to the controller.

In the operator screens, you can create or edit the hyperlink in offline mode and run it in online mode.

Display

When you open the target document, the application at the origin of the creation of the document or that is used to view the document becomes active. The Control Expert window is however not active. The Control Expert application continues to run as a background task.

Your actions on the keyboard will be directed to the document that you have just opened and not to the Control Expert application.

To make the Control Expert window active again, click on the window or on the icon in the taskbar.

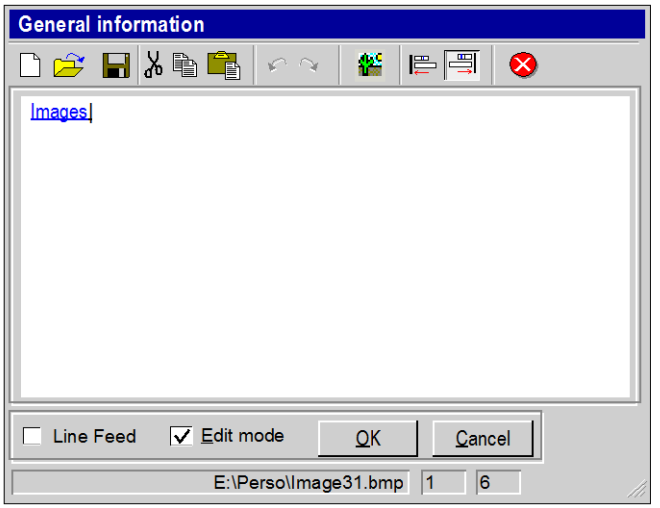
Drag / Drop

You can Drag & Drop, page 1819 icons between the file explorer and the hyperlink insertion windows. You can, for example, insert the full path of the target document.

List of barred characters

Principle

When you enter a comment for a hyperlink in the **General Information** or **DFB type description**,



window, the characters that are allowed are 8-bit characters of the ANSI alphabet.

You can write them in the language of your choice within the limits of the alphabet allowed.

Barred characters

16-bit UNICODE characters outside the ANSI range are not allowed.

Hyperlinks in the project browser

Object of this sub-section

This sub-section presents the operate-mode for creating, selecting and deleting a hyperlink in the project browser.

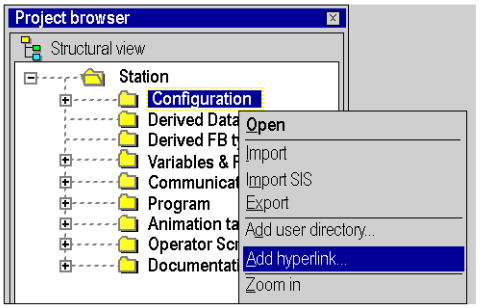
Creating a hyperlink in the project browser

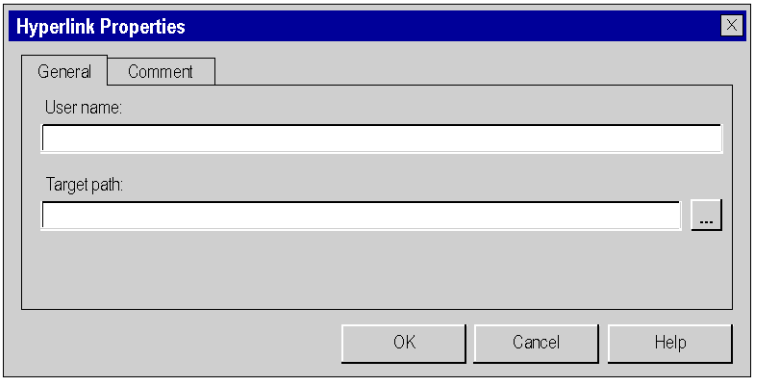
At a Glance

You can create hyperlinks in all the directories and sub-directories of the project browser. For example, you can create a hyperlink in the **Program** directory. You can also create another hyperlink in the **Sections** sub-directory of the **MAST task** directory.

Procedure

The table below describes the procedure for creating a hyperlink in the project browser using an existing directory.

Step	Action
1	Select the desired directory in the **Project browser**:
2	Right-click the selected directory. **Result**: The contextual menu appears. 
3	Click on **Add Hyperlink** in the contextual menu. **Result**: The **Hyperlink Properties** dialog box appears.

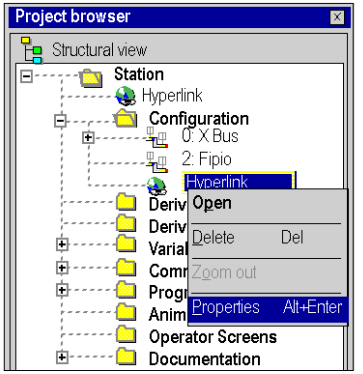
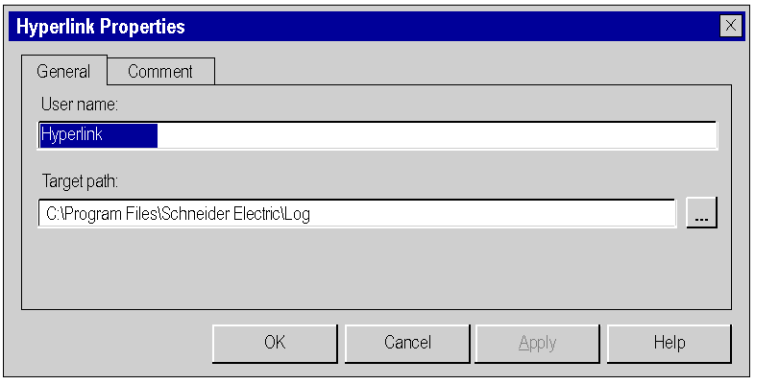
Step	Action
	
4	Enter the name of the hyperlink in the **User Name** field.
5	In the **Target Path** field: • use the ... button to choose the disk and/or the directory to select the target document (example: C:\Program Files\Office2000\Office\Example) or • enter the Internet address (example: https://www.se.com).
6	If you want to add a comment, select the **Comment** tab and enter your text in the entry field.
7	Click **OK** to confirm your choice.

Editing the properties of a hyperlink in the project browser

Procedure

The table below describes the procedure for editing the properties of a hyperlink in the project browser.

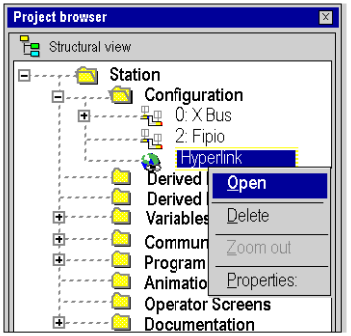
Step	Action
1	Select the hyperlink to be edited in the **Project browser**.
2	Right-click the selected hyperlink. **Result**: The contextual menu appears.

Step	Action
	
3	Click on **Properties** in the contextual menu. **Result**: The **Hyperlink Properties** dialog box appears. 
4	Modify the name of the hyperlink in the **User Name** field.
5	In the **Target Path** field: • use the  button to choose the disk and/or the directory to select the target document (example: c:\Program Files\Schneider Electric\log\example.doc) or • Change the Internet address (example: https://www.se.com).
6	If you want to add a comment, select the **Comment** tab and enter your text in the entry field.
7	Click **OK** to confirm the changes or **Cancel** to cancel.

Opening a hyperlink from the project browser

Procedure

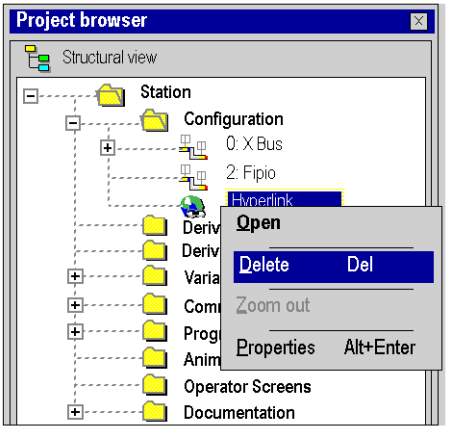
The table below describes the procedure for opening a hyperlink from the project browser.

Step	Action	
1	Select the hyperlink in the **Project browser**:	
2	Double click on the hyperlink represented by an icon with the name given to it when it was created. example: 	Right-click the selected hyperlink. **Result**: The contextual menu appears. 
3		Select **Open** in the contextual menu.
4	**Result**: The external document is launched.	

Deleting a hyperlink in the project browser

Procedure

The table below describes the procedure for deleting a hyperlink in the project browser.

Step	Action
1	Select the hyperlink to be deleted in the **Project browser**.
2	Right-click the selected hyperlink. **Result**: The contextual menu appears. 
3	Click **Delete** in the contextual menu. **Result**: The hyperlink disappears.

Hyperlink in the data editor

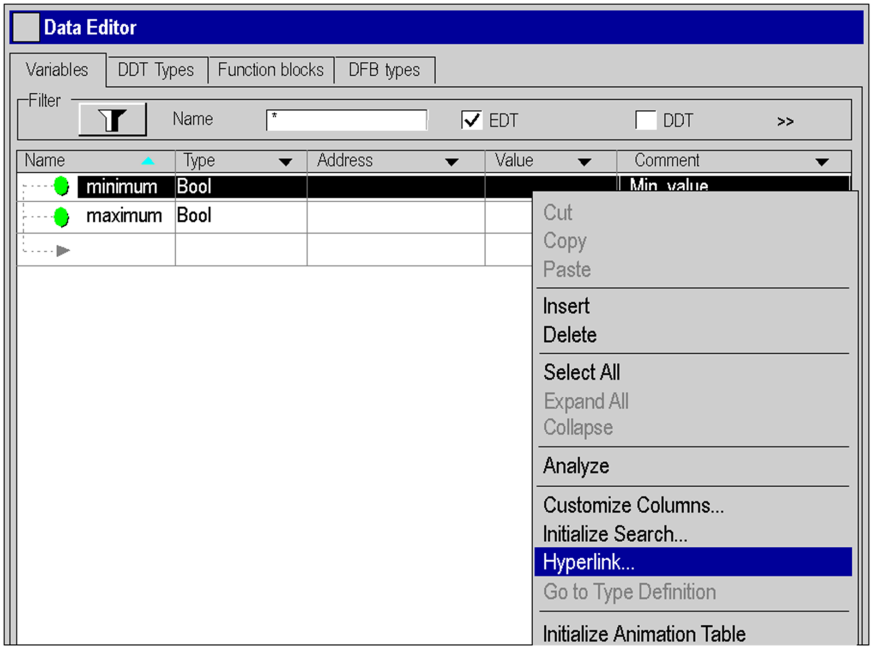
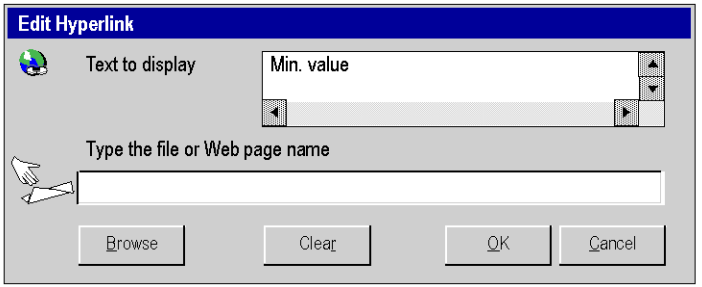
Object of this sub-section

This sub-section presents the operate-mode for creating, selecting, editing and deleting a hyperlink in the data editor.

Creating a hyperlink in the data editor

Procedure

The table below describes the procedure for creating a hyperlink in the data editor with a 1-step mode.

Step	Action
1	Open **the data editor**.
2	Select the comment where you wish to create a hyperlink in the **Comment** column of the **Variables** tab.
3	Right-click the selected comment. **Result**: The contextual menu appears. 
4	Click on **Hyperlink** in the contextual menu. **Result**: The **Edit Hyperlink** dialog box appears. 

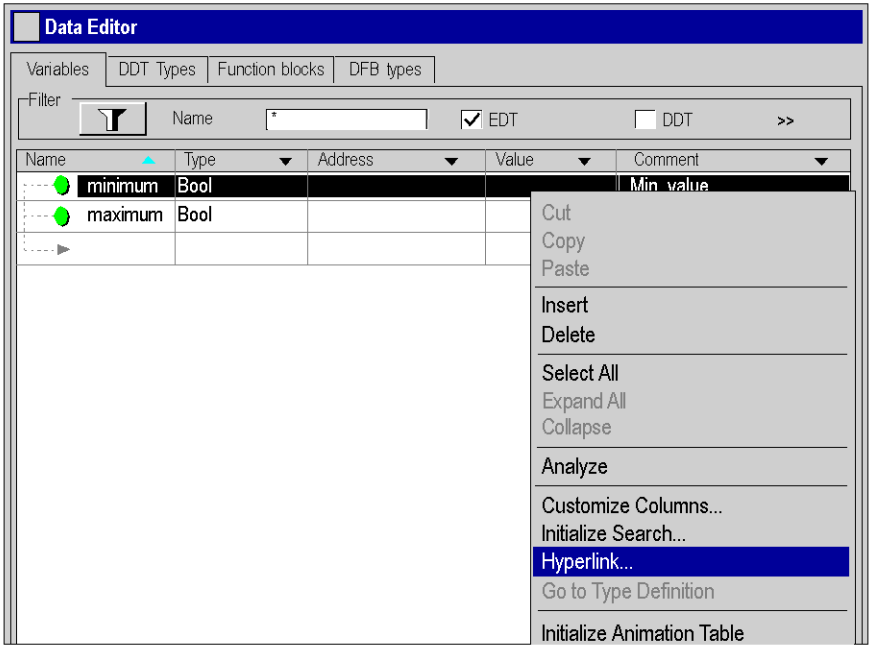
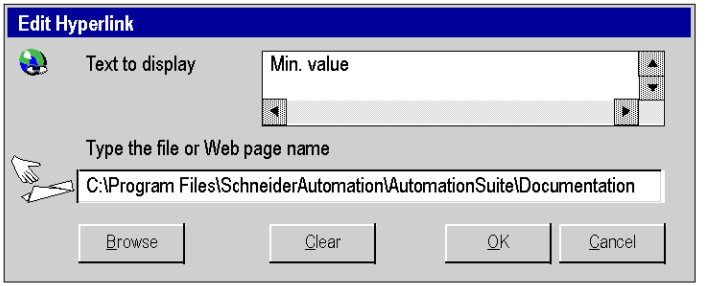
Step	Action
5	The selected comment appears in the **Text to display** field. In the **Enter name of file or Web page** field: • click  to select the target document (example: C:\Program Files\Office2000\Office \Example) or • enter the Internet address (example: https://www.se.com).
6	Click **OK** to confirm your choice.

Editing a hyperlink in the data editor

Procedure

The table below describes the procedure for modifying the properties of a hyperlink in the data editor.

Step	Action
1	Open the **data editor**.
2	Select the hyperlink comment to be edited in the **Comment** column of the **Variables** tab.
3	Right-click the selected comment. **Result**: The contextual menu appears.

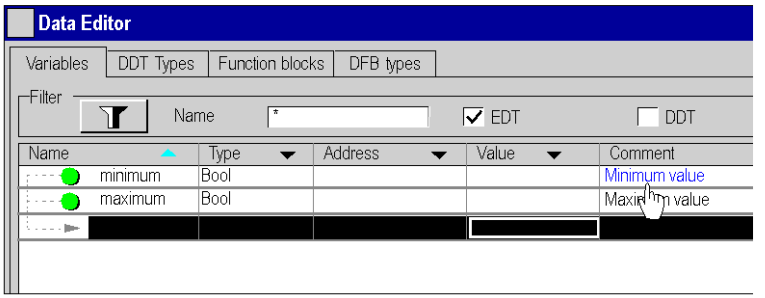
Step	Action
	
4	Click on **Hyperlink** in the contextual menu. **Result**: The **Edit Hyperlink** dialog box appears. 

Step	Action
5	The selected comment appears in the **Text to display** field. In the **Type the file or Web page name** field: • click  to select the target document (example: C:\Program Files\Office2000\Office \Example) or • change the Internet address (example: https://www.se.com).
6	Click **OK** to confirm the changes or **Cancel** to cancel.

Selecting a hyperlink in the data editor

Procedure

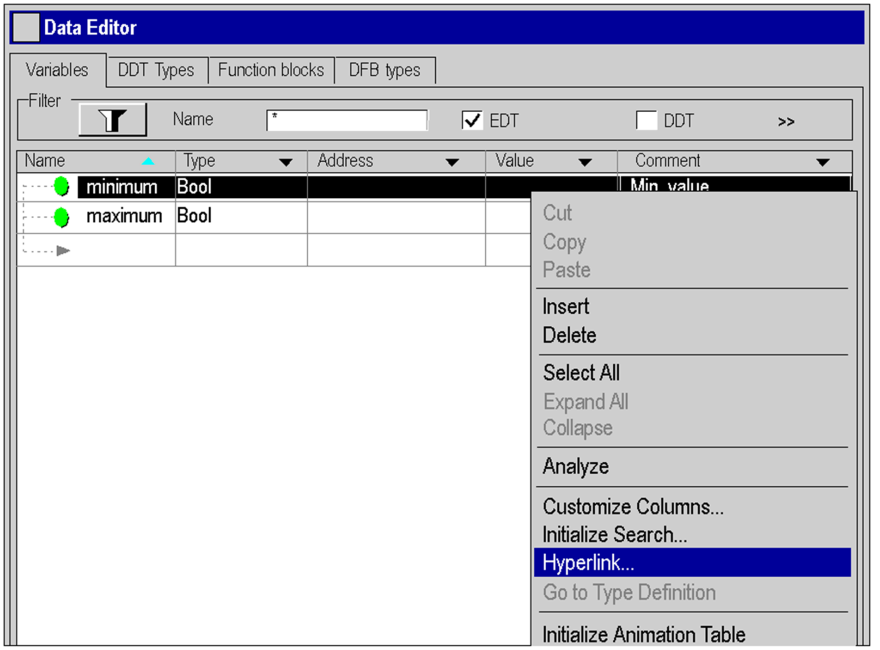
The table below describes the procedure for opening a hyperlink in the data editor.

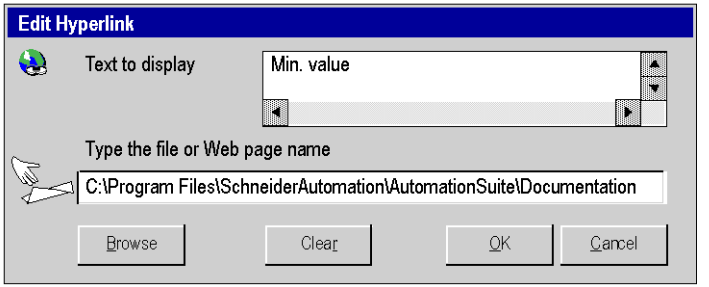
Step	Action
1	Open **the data editor**.
2	In the **Comment** column of the **Variables** tab, click on the required hyperlink (text in blue/mouse cursor changes to a hand). 

Deleting a hyperlink in the data editor

Procedure

The table below describes the procedure for deleting a hyperlink in the data editor.

Step	Action
1	Open the **data editor**.
2	Select the comment to be deleted in the **Comment** column of the **Variables** tab.
3	Right-click the selected comment. **Result**: The contextual menu appears. 
4	Click on **Hyperlink** in the contextual menu. **Result**: The **Edit Hyperlink** dialog box appears.

Step	Action
	
5	Click **Delete**: the text in the **Type the file or Web page name** field is deleted.
6	Click **OK** to confirm your choice. The hyperlink is deleted.

Hyperlinks in the DFB type attributes

Object of this sub-section

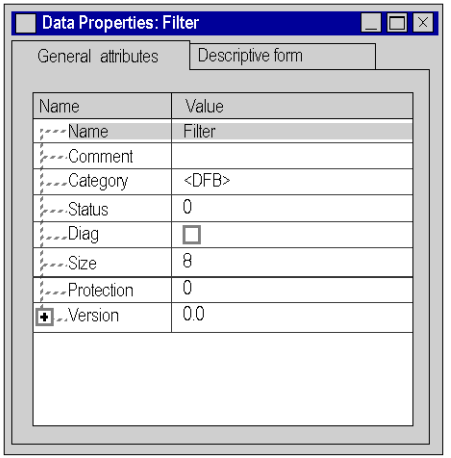
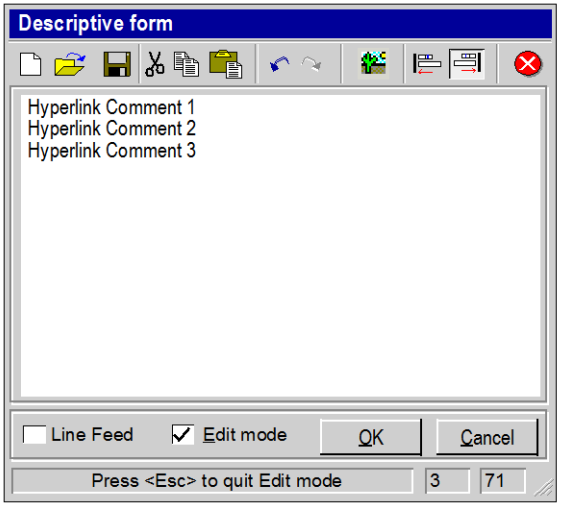
This sub-section presents the operate-mode for creating, editing and deleting a hyperlink in the DFB type attributes.

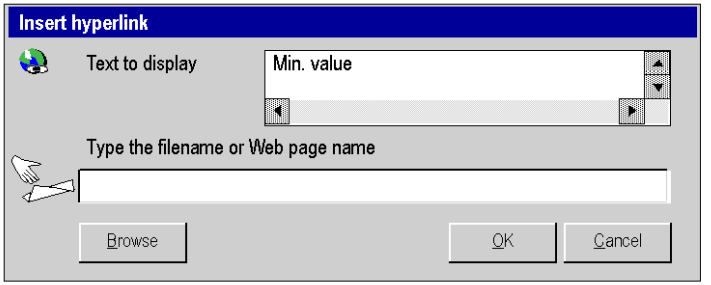
Creating a Hyperlink from DFB Type Attributes

Procedure

The table below describes the procedure for creating a hyperlink from the attributes, page 1269 of a DFB type with a 2-step mode. For example, you can associate an explanatory text with the definition of the DFB type.

Step	Action
1	Go to the **DFB Types**, page 1254 tab of the data editor.
2	Right-click the DFB type that you want to use to create a hyperlink. **Result**: the contextual menu appears.
3	Choose the **Data Properties** command. **Result**: the corresponding dialog box appears on the screen.

Step	Action
	
4	Click on the **Descriptive form** tab, then on the **Edit** button. **Result**: the **Descriptive form** window appears.  Details of window, page 1781
5	Enter a comment in the entry field.
6	Select one or more words (contiguous) from the comment you have entered and click on the  icon.

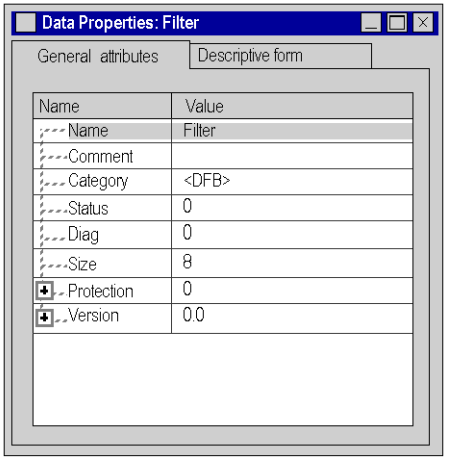
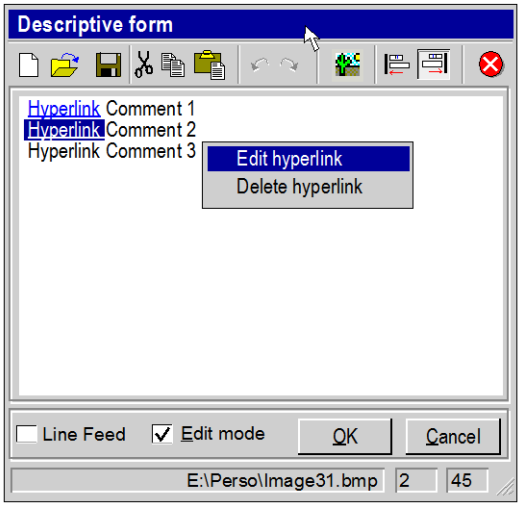
Step	Action
	Result: the **Insert hyperlink** window appears.  **Note**: You can create several hyperlinks in this way by selecting different words.
7	The text selected in step 6 appears in the **Text to display** field. In the **Type the filename or Web page name** field:  • click to select the target document (example: C:\Program Files \Office2000\Office\Example) or • enter the Internet address (example: https://www.se.com).
8	Click **OK** to confirm your choice.

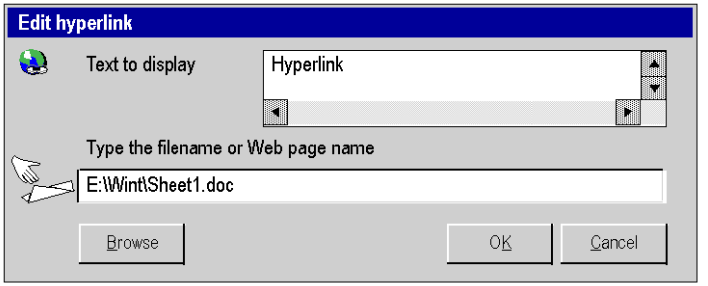
Modifying a Hyperlink from DFB Type Attributes

Procedure

The table below describes the procedure for editing the properties of a hyperlink in the attributes of a DFB type.

Step	Action
1	Go to the **DFB Types**, page 1254 tab of the data editor.
2	Right-click the DFB type whose hyperlink you wish to edit. **Result**: the contextual menu appears.
3	Choose the **Data Properties** command. **Result**: the corresponding dialog box appears on the screen.

Step	Action
	
4	Select the **Descriptive form** tab, and click on the **Edit** button. **Result**: the **Descriptive form** window appears.
5	Right click in the entry field, positioning the mouse on the hyperlink to edit. **Result**: the contextual menu appears. 
6	Click on **Edit hyperlink** in the contextual menu. **Result**: the **Edit hyperlink** window appears.

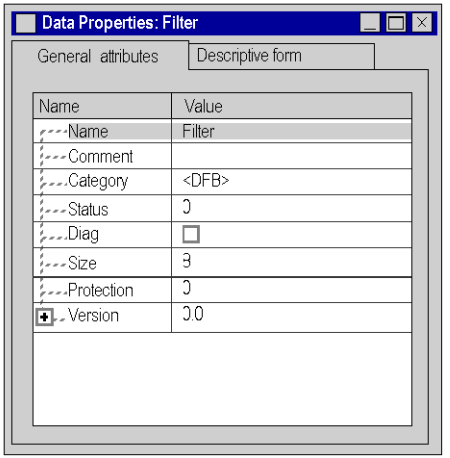
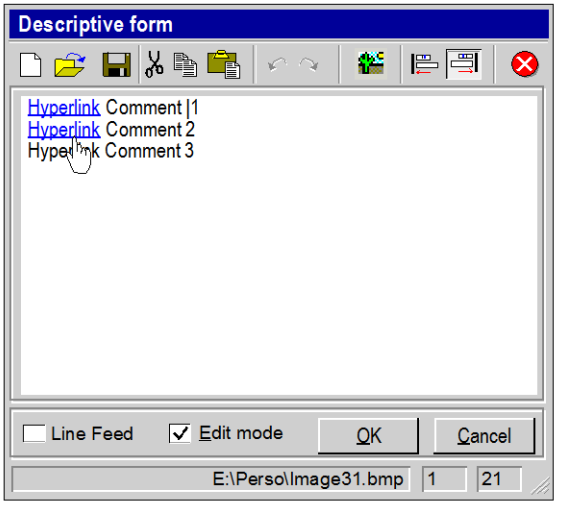
Step	Action
	
7	The name of the hyperlink appears in the **Text to display** field. In the **Type the filename or Web page name** field: • click **Browse** to select the target document (example: C:\Program Files\Office2000\Office \Example) or • change the Internet address (example: https://www.se.com).
8	Click **OK** to confirm your choice.

Selecting a Hyperlink from DFB Type Attributes

Procedure

The table below describes the procedure for opening a hyperlink from DFB type attributes.

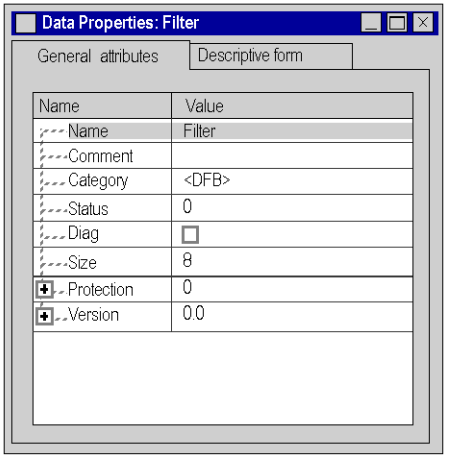
Step	Action
1	Go to the **DFB Types**, page 1254 tab of the data editor.
2	Right-click the DFB type whose hyperlink you wish to select. **Result**: the contextual menu appears.
3	Choose the **Data Properties** command. **Result**: the corresponding dialog box appears on the screen.

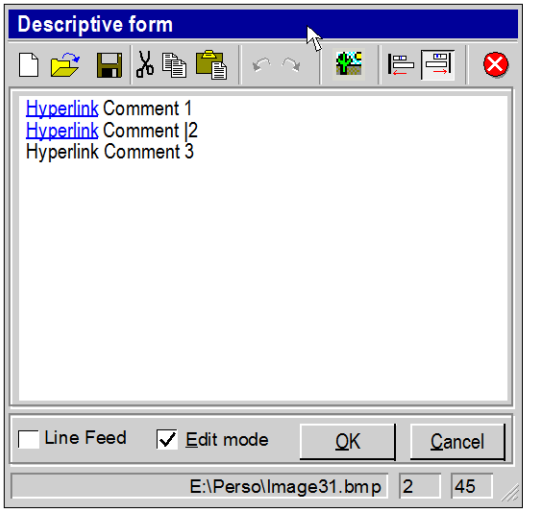
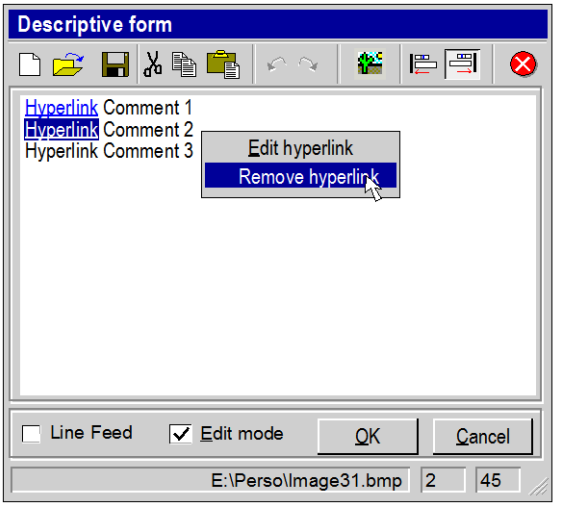
Step	Action
	
4	From the **Descriptive form** tab, click on the **Edit** button. **Result**: the **Descriptive form** window appears. 
5	In the **Descriptive form** window, click on the required hyperlink (text underlined in blue/mouse cursor changes to a hand).

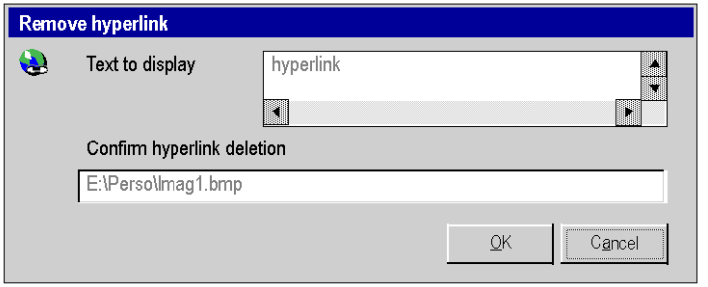
Deleting a Hyperlink from DFB Type Attributes

Procedure

The table below describes the procedure for deleting a hyperlink in the attributes of a DFB type.

Step	Action
1	Go to the **DFB Types**, page 1254 tab of the data editor.
2	Right-click the DFB type whose hyperlink you wish to delete. **Result**: the contextual menu appears.
3	Choose the **Data Properties** command. **Result**: the corresponding dialog box appears on the screen. 
4	From the **Descriptive form** tab, click on the **Edit** button. **Result**: the **Descriptive form** window appears.

Step	Action
	 Details of window, page 1781
5	Right click in the entry field, positioning the mouse on the hyperlink to be deleted. **Result**: the contextual menu appears. 
6	Click on **Remove hyperlink** in the contextual menu. **Result**: the **Remove hyperlink** window appears.

Step	Action
	
7	You do not have access to the **Text to display** or **Confirm the deletion of this hyperlink** entry fields.
8	Click **OK** to confirm your choice or click **Cancel** to cancel.

Hyperlinks in animation tables

Object of this sub-section

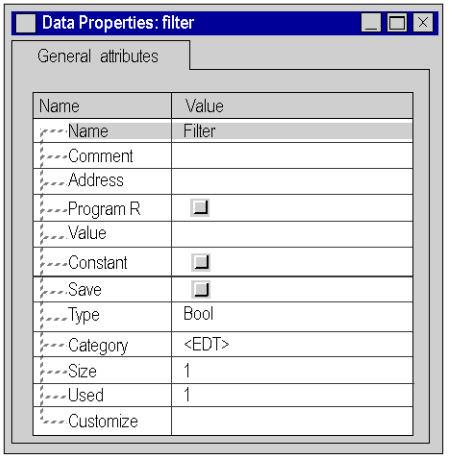
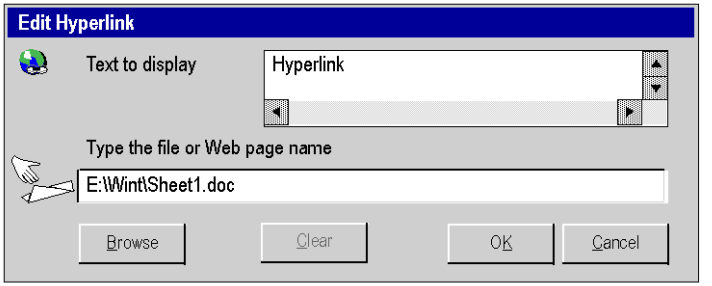
This sub-section presents the operate-modes for accessing a hyperlink in the animation tables.

Creating a hyperlink in an animation table

Procedure

The following table describes the procedure for creating a hyperlink in a comment for a variable in an animation table.

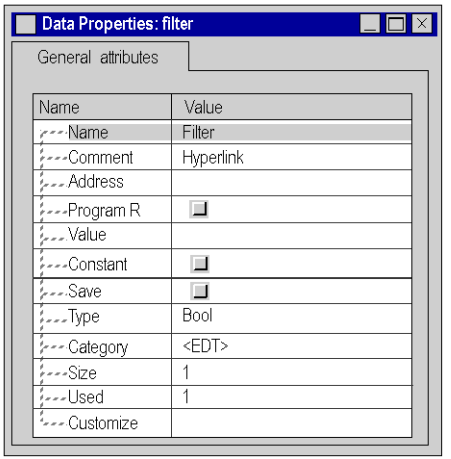
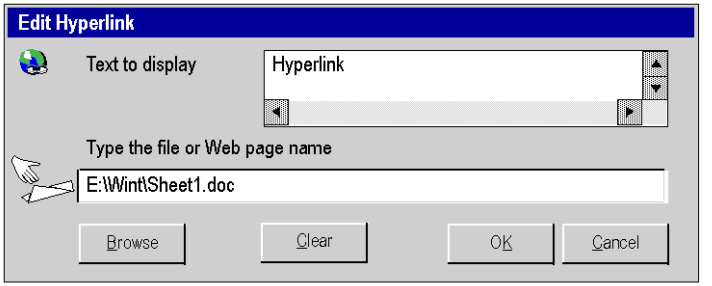
Step	Action
1	Opening an animation table.
2	Select the variable where the hyperlink is to be created.
3	Select **Data Properties** from the contextual menu. **Result**: The **Data Properties** dialog box appears.

Step	Action
	
4	Enter your comment in the comment line in the value column.
5	Select the comment.
6	Select **Hyperlink** in the contextual menu. **Result**: The **Edit Hyperlink** window appears. 
7	The selected comment appears in the **Text to display** field. In the **Enter name of file or Web page** field:  • click to select the target document (example: C:\Program Files \Office2000\Office\Example) or • enter the Internet address (example: https://www.se.com).
8	Click **OK** to confirm your choice.

Editing a hyperlink from an animation table

Procedure

The table below describes the procedure for editing the properties of a hyperlink in an animation table.

Step	Action
1	Opening an animation table.
2	Select the variable where the hyperlink is located in the comment.
3	Select **Data Properties** from the contextual menu. **Result**: The corresponding dialog box appears on the screen. 
4	Select the comment.
5	Select **Edit Hyperlink** in the contextual menu. **Result**: The **Edit Hyperlink** window appears. 

Step	Action
6	The name of the hyperlink appears in the **Text to display** field. In the **Type the file or Web page name** field: • click **Browse** to select the target document (example: C:\Program Files\Office2000\Office \Example) or • change the Internet address (example: https://www.se.com).
7	Click **OK** to confirm your choice.

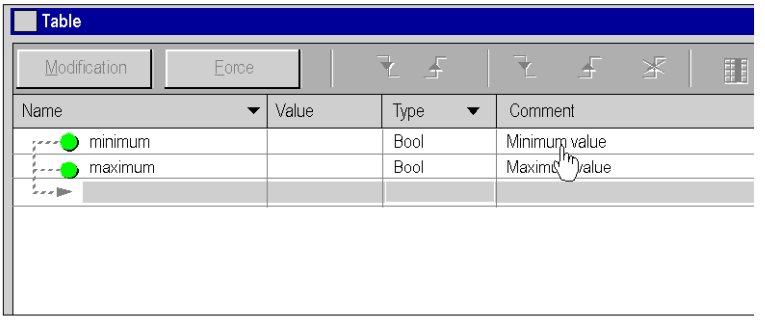
Selecting a hyperlink from an animation table

Principle

In an animation table, you can open a hyperlink associated with a variable comment.

Procedure

The table below describes the procedure for opening a hyperlink in animation tables.

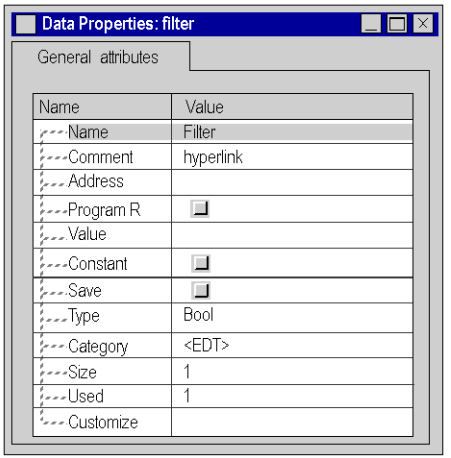
Step	Action
1	Open an **animation table**.
2	In the **Comment** column, double-click on the hyperlink for the selected variable. 

Deleting a hyperlink from an animation table

Procedure

The following table describes the procedure for deleting a hyperlink in a comment for a variable in an animation table.

NOTE: This procedure is only possible if the hyperlink was created, page 1804 from the Data Properties window.

Step	Action
1	After creating, page 1804 a hyperlink in a comment for a variable, the **Data Properties** window remains open. 
2	Select the comment from which the hyperlink is to be deleted.
3	Select **Delete Hyperlink** in the contextual menu.

Hyperlinks in runtime screens

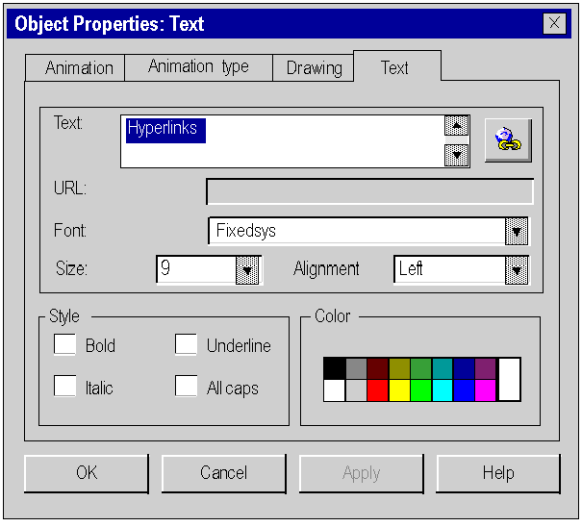
Object of this sub-section

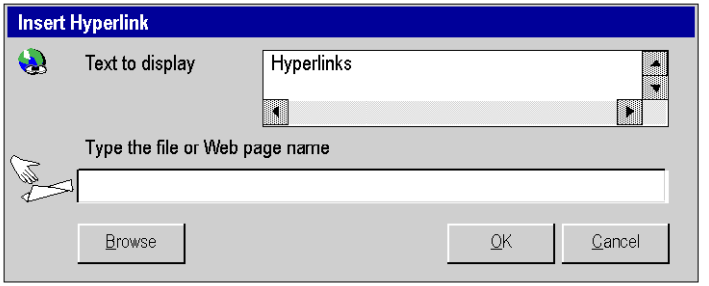
This sub-section presents the operate-mode for creating, editing and deleting a hyperlink in runtime screens.

Creating a hyperlink from operator screens

Procedure

The table below describes the procedure for creating a hyperlink in operator screens.

Step	Action
1	Open **an operator screen**.
2	Select a **Text** object or create one.
3	Select **Properties** in the contextual menu. **Result**: The following window appears. 
4	Click on the icon  of the **Text** tab. **Result**: The **Insert Hyperlink** window appears.

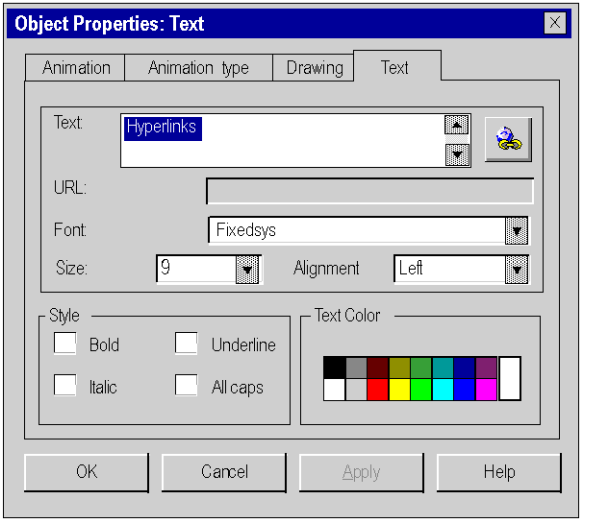
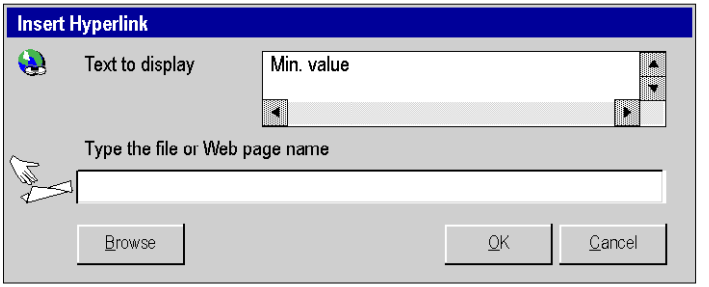
Step	Action
	
5	The text specified in the **Text** tab appears in the **Text to display** field. In the **Type the file or Web page name** field: • select the target document (C:\Program Files\Office2000\Office\Example) by using the  button or • enter the Internet address (example: https://www.se.com).
6	Click **OK** to confirm your choice.

Editing a hyperlink in the operator screens

Procedure

The table below describes the procedure for editing the properties of a hyperlink in an operator screen object.

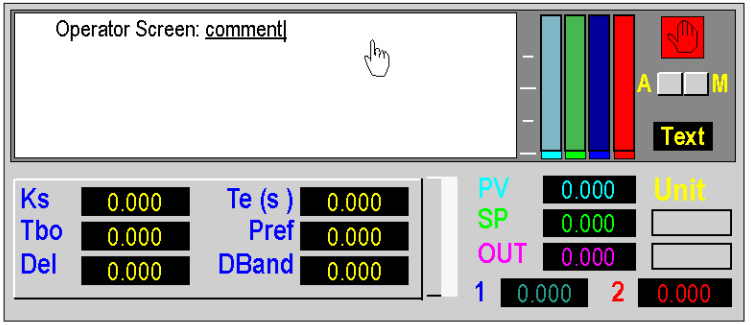
Step	Action
1	Open the **operator screen**.
2	Select the **Text** object to be edited.
3	Select **Properties** in the contextual menu. **Result**: The following window appears.

Step	Action
	
4	Click on the icon  of the **Text** tab. **Result**: The **Insert Hyperlink** window is displayed. 
5	The text of the **Text** tab appears in the **Text to display** field. In the **Type the filename or web page name** field:  • use the button to choose the disk and/or the directory to modify the target document (example: C:\Program Files\Office2000\Office\Example) or • change the Internet address (example: https://www.se.com).
6	Click **OK** to confirm the changes or **Cancel** to cancel.

Selecting a hyperlink in an operator screen

Procedure

The table below describes the procedure for opening a hyperlink in an operator screen.

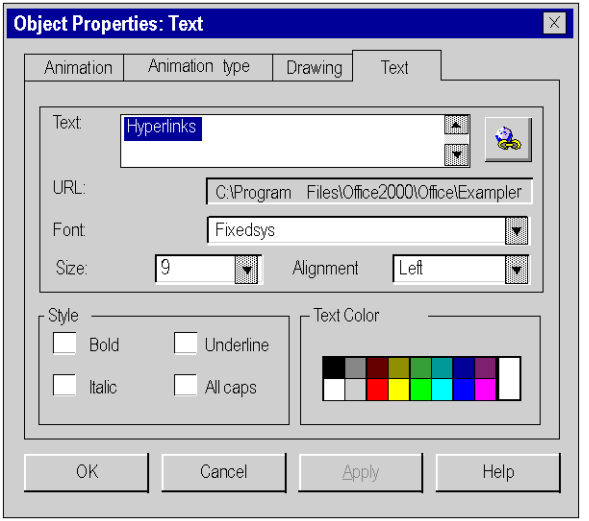
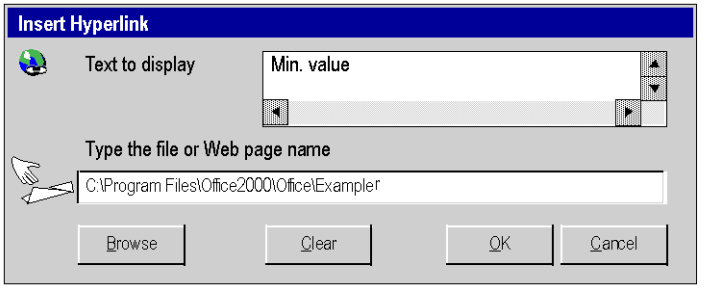
Step	Action
1	Open **an operator screen**.
2	Click on the underlined text and the mouse cursor is changed into a thumbnail.  **Result**: The document associated with the hyperlink is opened.

Deleting a hyperlink in an operator screen

Procedure

The table below describes the procedure for deleting a hyperlink in an operator screen.

Step	Action
1	Open **the operator screen**.
2	Select the **Text** object or hyperlink to be deleted.
3	Select **Properties** in the contextual menu. **Result**: The following window appears.

Step	Action
	
4	Click on the icon  of the **Text** tab. **Result**: The **Insert Hyperlink** dialog box appears. 
5	Click **Clear**: the text in the **Type the file or Web page name** field is cleared.
6	Click **OK** to confirm your choice. The hyperlink is deleted.

Hyperlinks in the documentation folder

Object of this sub-section

This sub-section presents the operate-mode for creating, editing and deleting a hyperlink in the documentation folder of a Control Expert project.

Creating a hyperlink in the general information of the project folder

Principle

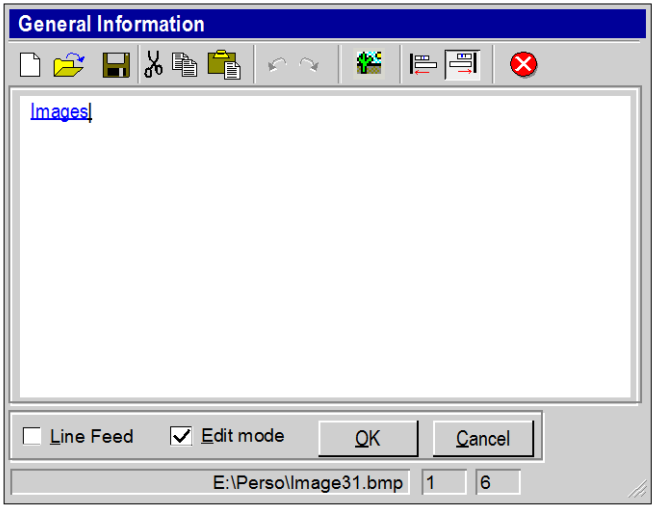
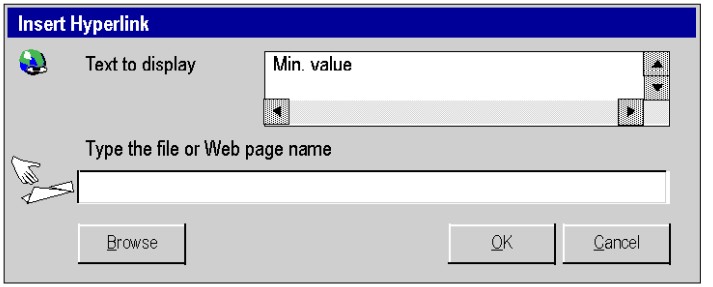
The **hyperlink** functionality is also used to make links to documents for printing in the documentation folder of a Control Expert project. For example, they can be used to present the architecture of a project.

You can in this way create a hyperlink from **General Information** sections of the documentation folder. These are image documents (.bmp) and you can print them in the **General Information** section of the documentation folder.

Procedure

The table below describes the procedure for creating a hyperlink in the documentation folder with a 2-step mode.

Step	Action
1	Use the project browser to select the **General Information** sub-directory of the **Documentation** directory.
2	Select **Open** in the contextual menu. **Result**: The **General Information** window appears.

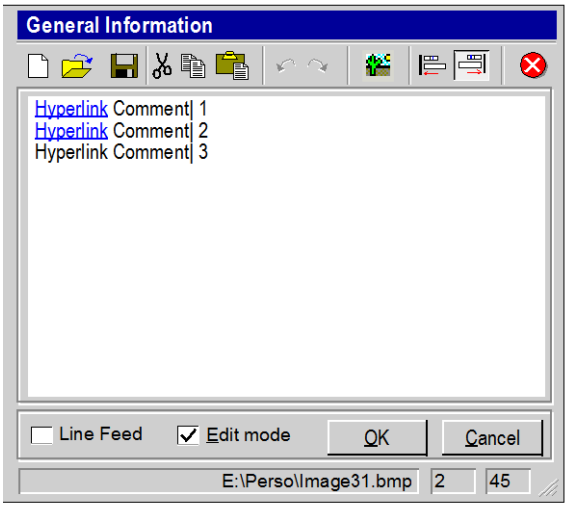
Step	Action
	 Details of window, page 1781
3	Enter a comment in the entry field.
4	Select one or more words (contiguous) from the comment you have entered and click on the icon  . **Result**: The **Insert Hyperlink** window is displayed.  **Note**: You can create several hyperlinks in this way by selecting different words.

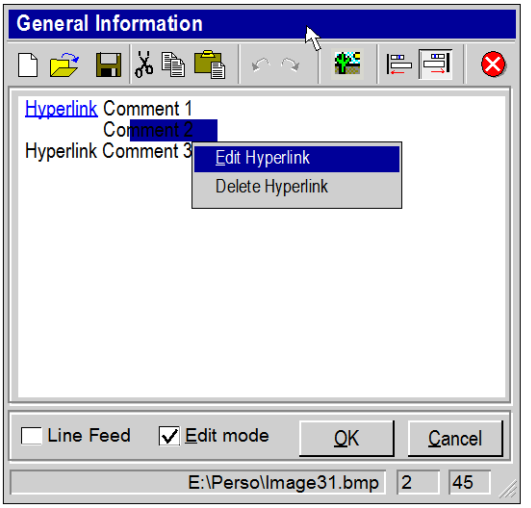
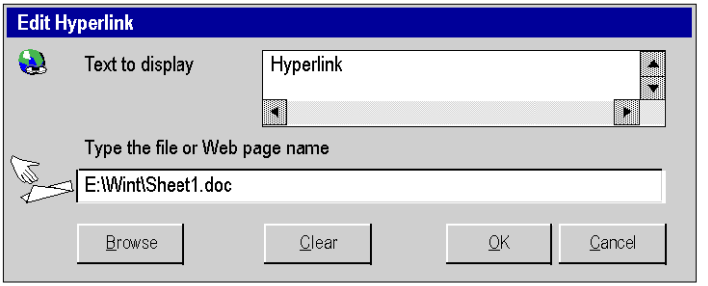
Step	Action
5	The text selected in step 4 appears in the **Text to display** field. In the **Type the filename or web page name** field, use the  button to choose the disk and/or the directory to select the target document (example: C:\Images\structure1.bmp).
6	Click **OK** to confirm your choice.

Editing a hyperlink in the documentation folder

Procedure

The table below describes the procedure for editing the properties of a hyperlink in the general information section of the documentation folder.

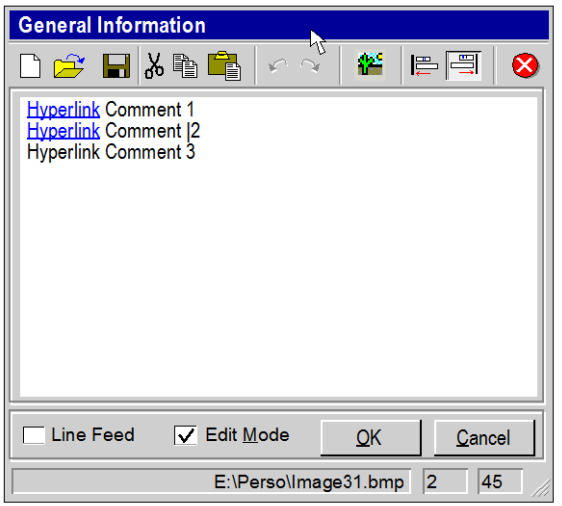
Step	Action
1	Use the project browser to select the **General Information** sub-directory of the **Documentation** directory.
2	Select **Open** in the contextual menu. **Result**: The **General Information** window appears.  Details of window, page 1781

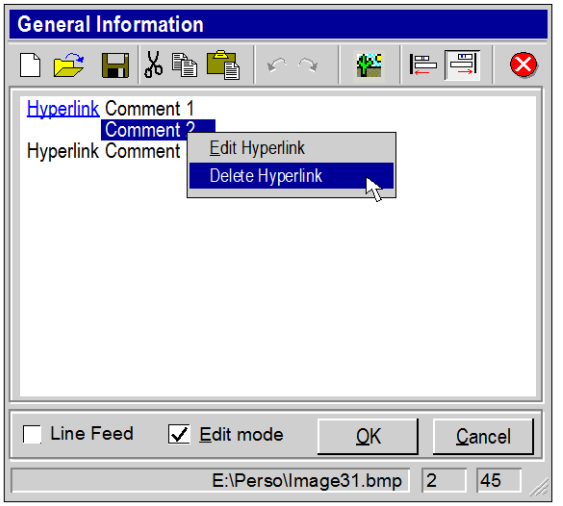
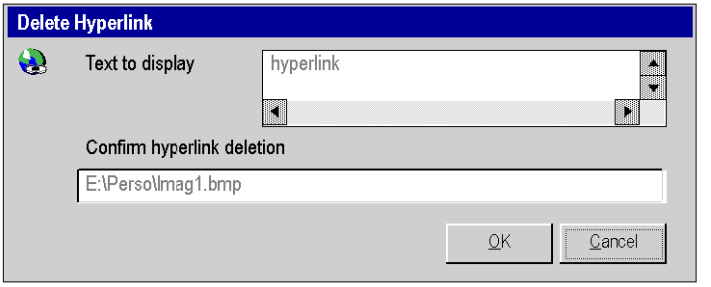
Step	Action
3	Right click in the entry field, positioning the mouse on the hyperlink to edit **Result**: The contextual menu appears. 
4	Click on **Edit Hyperlink** in the contextual menu. **Result**: The **Edit Hyperlink** window appears. 
5	The name of the hyperlink appears in the **Text to display** field. In the **Type the file or Web page name** field, use the  button to select the target document (example: C:\Images\structure1.bmp).
6	Click **OK** to confirm your choice.

Deleting a hyperlink in the documentation folder

Procedure

The table below describes the procedure for deleting a hyperlink in the general information section of the documentation folder.

Step	Action
1	Use the project browser to select the **General Information** sub-directory of the **Documentation** directory.
2	Select **Open** in the contextual menu. **Result**: The **General Information** window appears.  Details of window, page 1781
3	Right click in the entry field, positioning the mouse on the hyperlink to be deleted **Result**: The contextual menu appears.

Step	Action
	
4	Click on **Delete Hyperlink** in the contextual menu. **Result**: The **Delete Hyperlink** window appears. 
5	You do not have access to the **Text to display** or **Confirm hyperlink deletion** entry fields.
6	Click **OK** to confirm your choice or click **Cancel** to cancel.

Hyperlinks and drag-and-drop

Object of this sub-section

This sub-section presents the drag-and-drop function with hyperlinks.

Drag/drop

Principle

You can use the drag/drop function to create a hyperlink. This function can be used to:

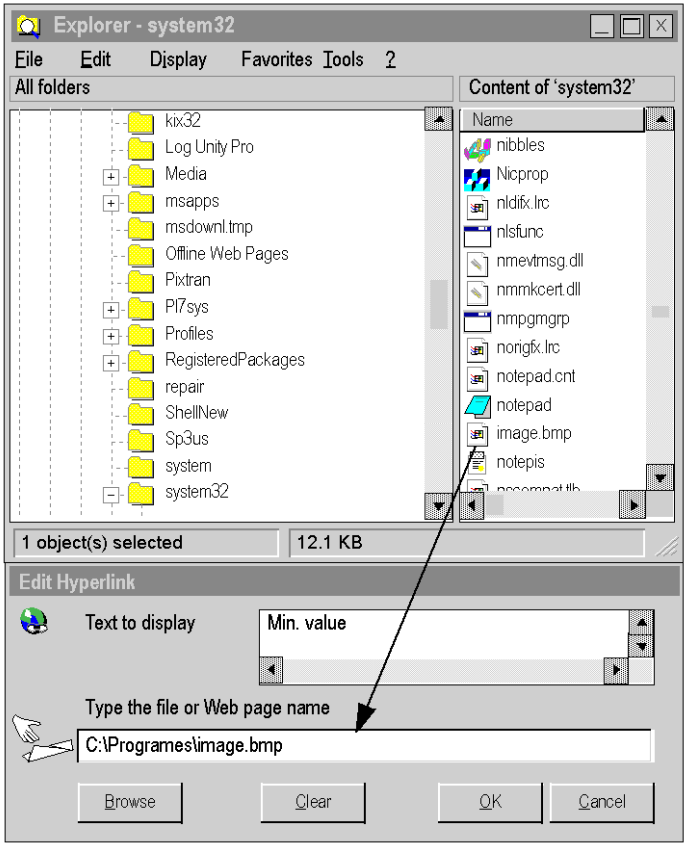
- drag and drop the path name of the target document between the explorer and the **Hyperlink Properties**, **Edit Hyperlink** or **Insert Hyperlink** dialog boxes,
- drag and drop a word or expression between a text document and the comment in the **General Information** window.

NOTE: only paths for **.bmp** fillies can be used for drag and drop.

Procedure

The table below shows the procedure for dragging/dropping between the explorer and the **Hyperlink Properties**, **Insert Hyperlink** or **Edit Hyperlink** dialog boxes.

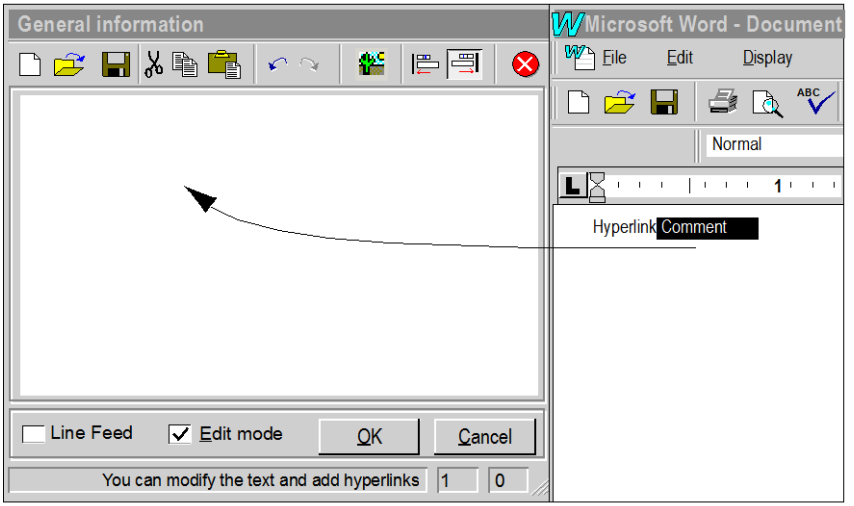
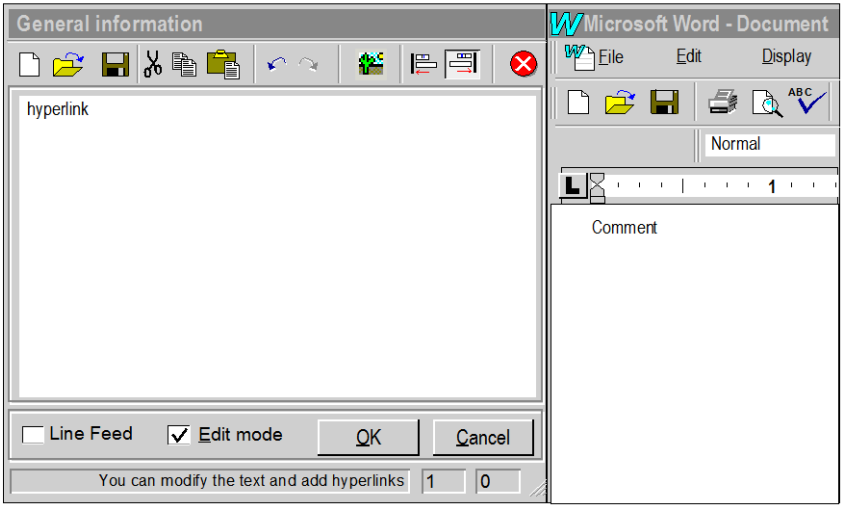
Step	Action
1	Locate the target document in the explorer.
2	In Control Expert, open either the **Hyperlink Properties**, **Edit Hyperlink** or **Insert Hyperlink** dialog box.

Step	Action
3	Size and move the Control Expert and explorer windows so they are displayed side by side on the screen without overlapping.
4	Drag and drop the file representing the target document between the explorer and the **Type the file or Web page name** field of the dialog box (image.bmp in our example). **Result**: The entire path of the target document appears in the **Type the file or Web page name** field. 

Procedure

The table below shows the procedure for dragging and dropping between a text file and the **General Information** dialog box.

Step	Action
1	In Control Expert, open the **General Information** window from the **Documentation** directory.
2	Open the text file.
3	Size the Control Expert and explorer windows to display them side by side on the screen.

Step	Action
4	Select the text to be moved in the text file. 
5	Drag and drop a word or expression between the **General Information** window and a word-processing application. **Result:** The text selected appears in the Hyperlink field.  **Note**: If you press and hold down the Ctrl key, a "+" sign will appear on the mouse cursor. When you drop the expression or word ("Hyperlink" in this case), it will be copied and not moved from the word processor to the comment editor.

User Interface

What's in This Chapter

Window 1825
Anchored window 1827
Menus 1830
Menu commands 1832
Toolbar 1835
Output Windows 1841
Status bar 1843
Directory Structures 1846
Using the Mouse 1847
Using the Keyboard 1848
Data selection boxes 1850
Dialog boxes 1851

Overview

This chapter describes the Control Expert user interface. It provides an overview of the elements used and how they are used.

Window

Introduction

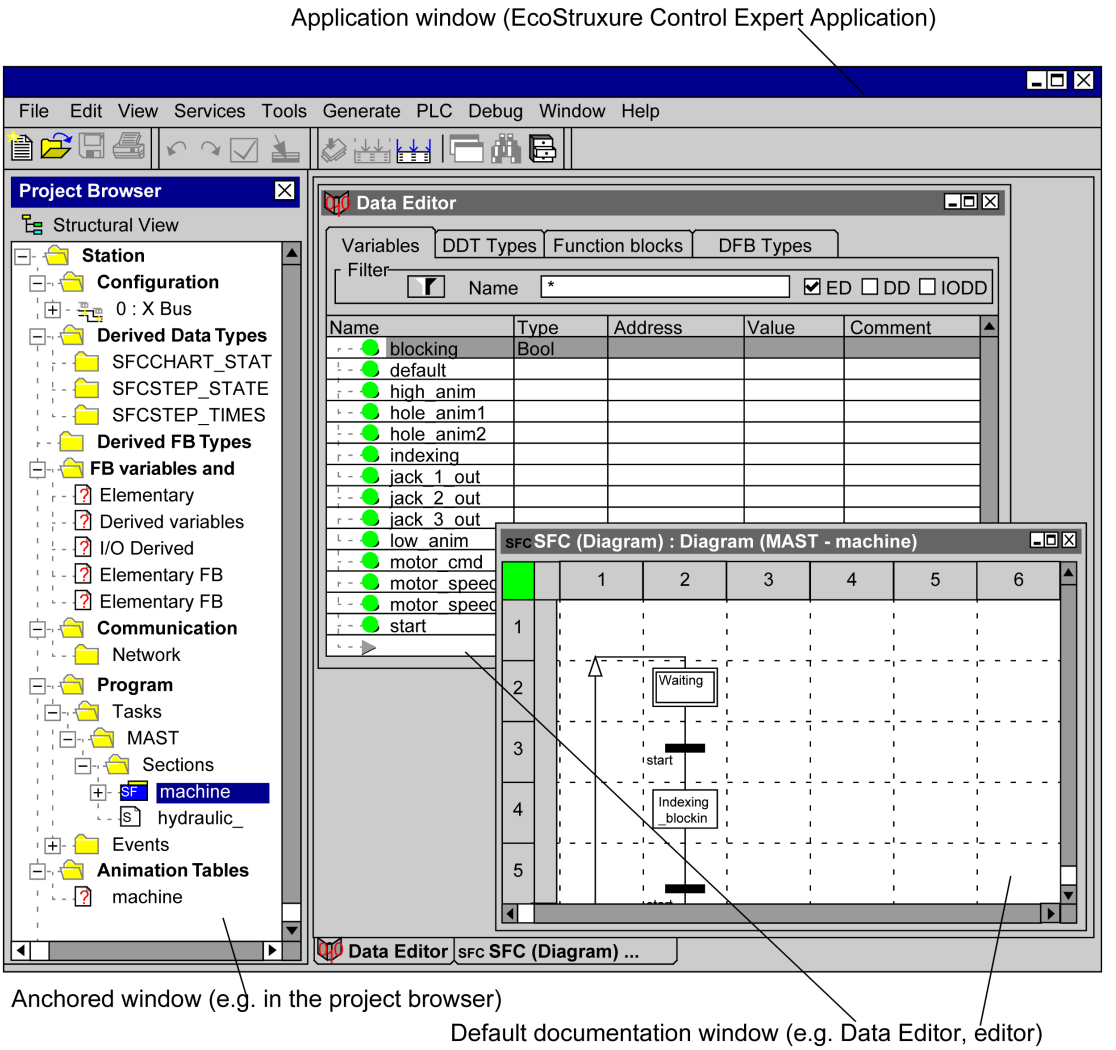
In Windows there are two types of windows:

- Application windows
- Documentation windows

Windows consist of:

- a title bar to move the window and to change its size
- Buttons to display the window as a symbol, to minimize, maximize and close.
- the actual contents of the window

Types of windows:



Application windows

When you start Control Expert, an application window will open on your desktop. You can place the application window at any location on your desktop. Or you can minimize it so that it appears in your taskbar.

In this application window you can open or create a project. In the title bar of the application window the name of the project appears.

Documentation windows

When you open or create a project you can open various documentation windows. For example, documentation windows are sections, where you create your application or the documentation window for the Data Editor.

The information contained in documentation windows is separated into different areas. The different areas in a dialog box can be selected using the **F6** key.

If the amount of information contained in a documentation window is too complex to be displayed in one documentation window, the documentation window is split into several tabs. The different tabs can be selected using the **Ctrl**+**Tab** key combination.

Several documentation windows can be open at the same time, but only one documentation window can be active. An active documentation window is marked by the color of the title bar. Depending on the documentation window the menu commands in the pulldown menu, page 1830 and the toolbar, page 1835 change.

If several windows are open at the same time you can switch between them using the key combination **Ctrl**+**F6**.

An exception to this are anchored windows, page 1827

Anchored window

Introduction

Anchored windows have basically the same function as standard windows with the difference that they can be anchored in certain positions. Anchored windows can also be freely positioned on the screen (also outside of the application window).

Build

Anchored windows consist of:

- a title bar to move the window and to change its size
- a button to prevent anchoring ()

- a button to close the window (**x**)
- the actual contents of the window

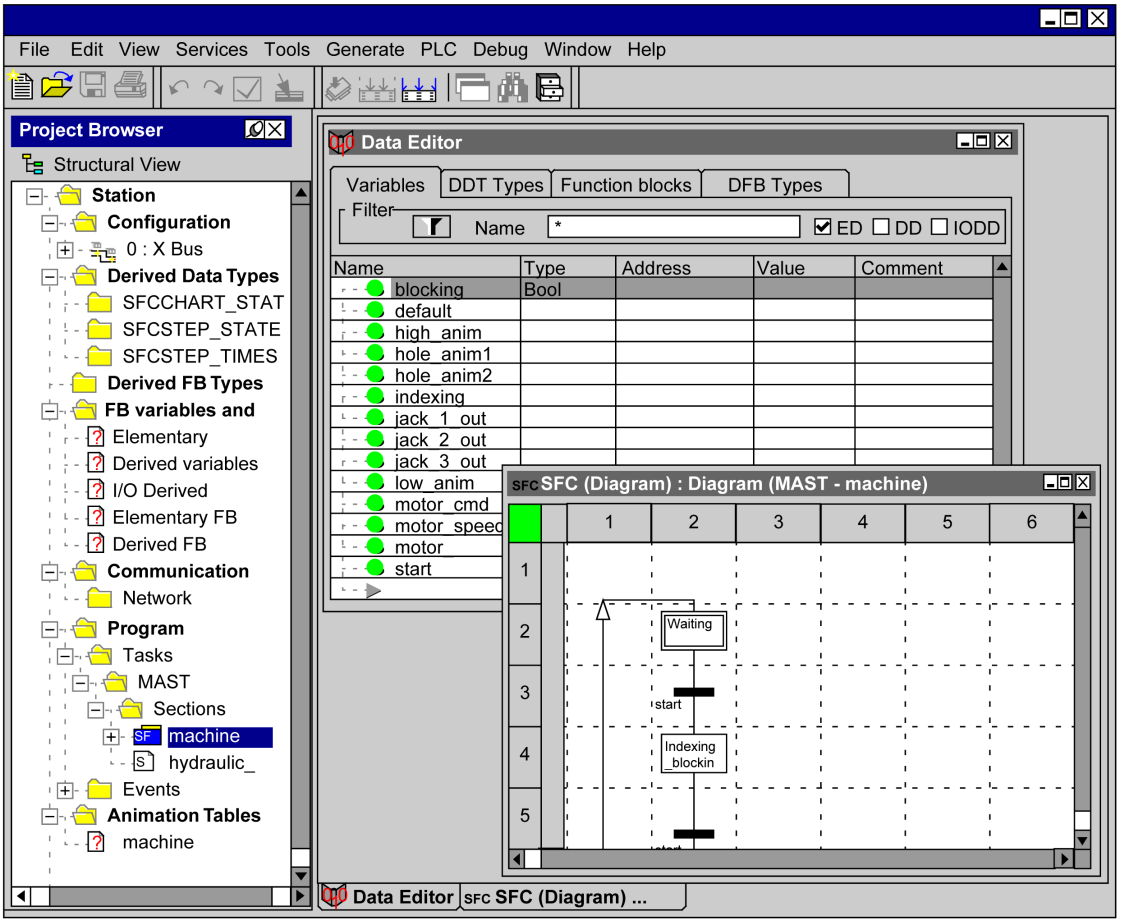
Function

Anchored windows have the following extra functions in comparison to standard windows:

- They can be anchored to the edge of the application window.
- They can be positioned outside of the application window.
- The menu commands in the **Window** (like e.g. **Arrange**) do not affect the anchored window.

Example of use

An example of an anchored window is the project browser:



Anchoring windows

Process the following steps to anchor a window:

Step	Action
1	Click the title bar of the anchored window and keep the mouse button pressed.
2	Drag the window close to an application window edge. **Result:** The window frame displays the future position of the window.
3	Release the mouse button. **Result:** The window is anchored to the selected edge. On the new position, the window is • anchored when the new position of the window is on the border or outside of the application window. • not anchored when the new position of the window is inside the application window or when the new position of the window is on the border or outside of the application window, but the **Ctrl** key was held when releasing the mouse button.

Menus

Introduction

There are 3 types of menus:

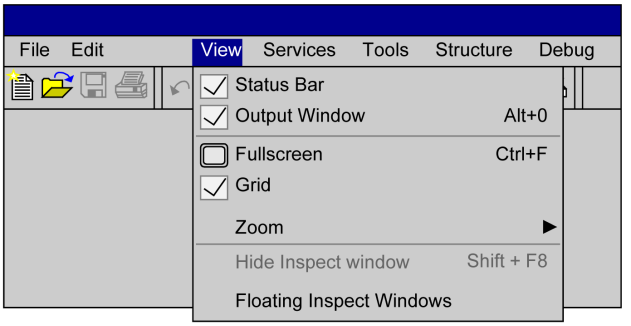
1. Main or pulldown menus
2. Submenus
3. Shortcut or popup menus

Main or pulldown menus

The titles of the individual menus are displayed in the menu bar. The individual menu commands, page 1832 are listed in the pulldown menus.

A pulldown menu is opened by left-clicking on the title of the menu or by pressing **Alt +selected letter** (underlined letter, page 1833). To go directly to a menu command, drag the mouse pointer down the menu and then release the mouse button. The menu can be closed by clicking on the title of the menu or anywhere outside of the menu or by pressing **Esc**.

Menu bar with pulldown menu:

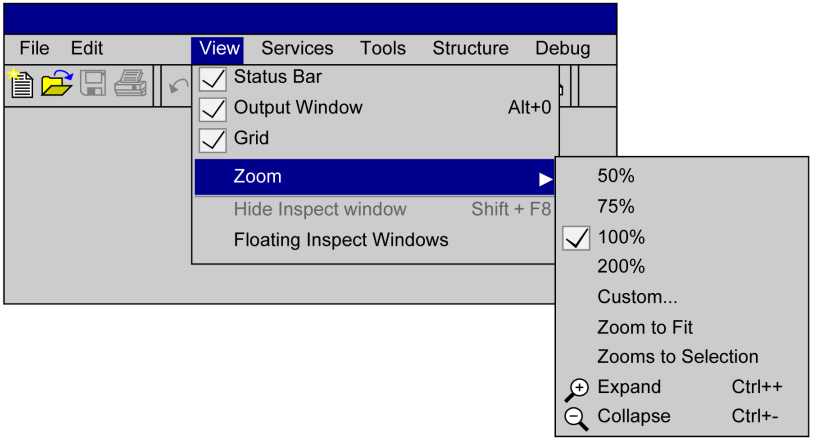


Submenus

The title of a submenu is a menu command, page 1832 of the menu above it. The individual menu commands for the submenu are listed in a pulldown menu. Menu commands which contain a submenu can be recognized by an arrow icon.

You can open a submenu by clicking on the menu title (left mouse button) or by placing the mouse pointer on the menu title for a short time. The menu can be closed by clicking on the title of the menu or anywhere outside of the menu or by pressing **Esc**.

Menu with submenu:



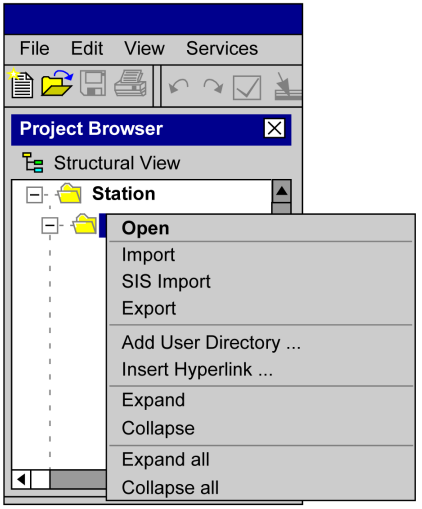
Shortcut or popup menus

Shortcut menus are menus which contain menu commands, page 1832 specially for the selected object.

You can open a shortcut menu by clicking on the object (right mouse button) or selecting the object and confirming with **Shift** + **F10** or press the key. The menu can be closed by clicking anywhere outside of the menu or by pressing **Esc**.

Shortcut menus can also be called if several objects are selected. In this case, the menu only contains the menu commands which are valid for all objects.

Object with shortcut menu:

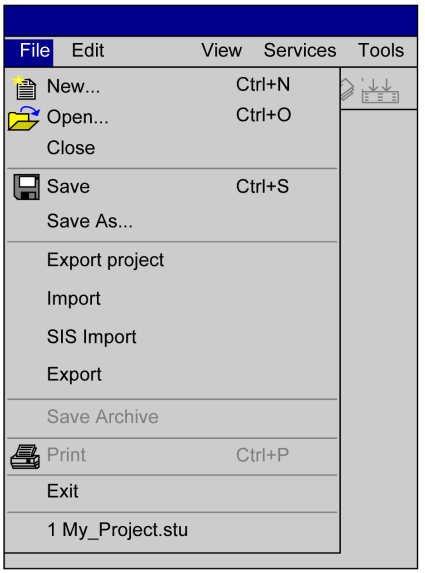


Menu commands

Introduction

Menu commands are used to execute commands or to call dialog boxes.

Example of a menu with menu commands:



Keyboard shortcuts or mnemonics

Keyboard shortcuts (underlined letters) in menu commands allow you to select menu commands using the keyboard. A main menu (menu title) and subsequently a menu command can be selected by holding down **Alt** and simultaneously entering the underlined letter in the menu title and then that of the menu command.

For example, in the **File** you want to use the menu command **Save...** you must press **Alt**+**D** to open the menu and then **Alt**+**S** to execute the menu command.

Grayed out menu command

If a menu command is not available it is grayed out. One or more other commands must be executed before the desired menu command can be executed.

Periods () after the menu command

On execution of this menu command a dialog box is opened with options, which must be selected before execution.

Checkmark (√) in front of the menu command

The menu command is enabled. If the menu command is selected, the checkmark disappears and the menu command is disabled. The checkmark is mostly used to identify enabled modes (e.g. normal display, dial in mode etc.).

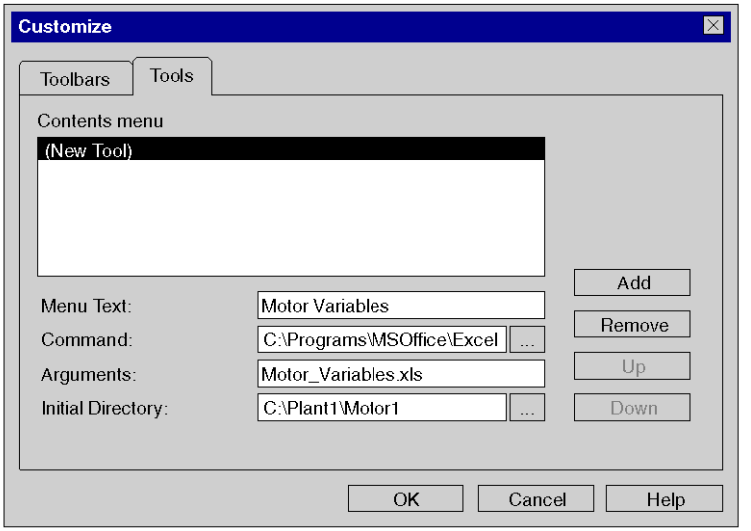
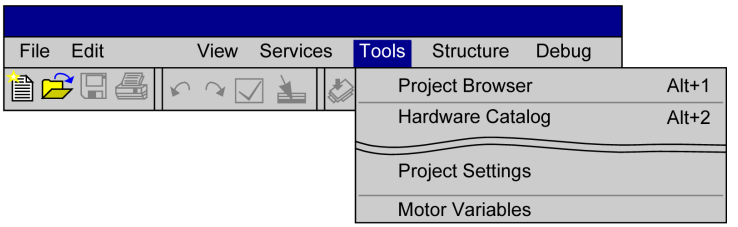
Shortcut keys

Shortcut keys (e.g. **F8**) or key combinations (e.g. **Ctrl**+**R**) after the menu command is a shortcut way for executing the menu command. You can select the menu command using this shortcut key or key combination without having to open the menu. For example, **Ctrl**+**S** to execute the menu command **Save**.

Starting external applications

Process the following steps to launch external applications from Control Expert:

Step	Action
1	Open, page 610**Tools** > **Adjust** the **Tools** tab.
2	Click the **Add** button.
3	In the **Menu Text** field, enter the name of the menu item to be created (e.g. `Motor Variables`).
4	In the **Command** field, enter the path to the program to be launched (e.g. Microsoft Excel).
5	In the **Arguments** field, enter the name of the file to be opened (e.g. Motor_Variables.xls).

Step	Action
6	In the **Output Directory** field, enter the path to the file to be opened. Display of the **Tools** tab: 
7	Confirm the entries with **OK**. **Result:** In the main menu **Tools** a menu command to launch the selected application will be generated (in this example **Motor Variables**). **Example:** 

Toolbar

Introduction

Toolbars allow you to quickly locate and execute frequently used functions.

Instead of opening a menu and then selecting the menu command, toolbars allow you to simply click the respective symbol to execute the desired command.

The software provides you with various standard toolbars. In addition, you can also create your own toolbar.

Build

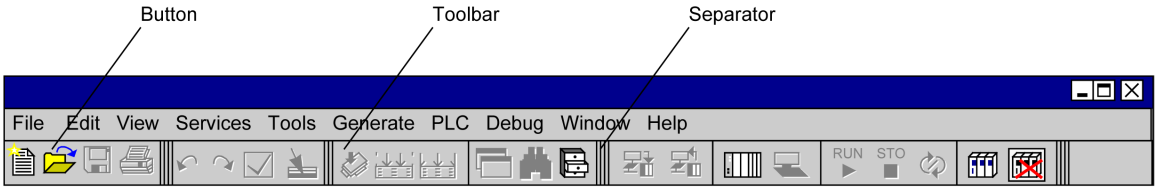
Toolbars consist of a row of buttons and combo boxes that call the corresponding functions.

There are several toolbars and they are displayed in a common toolbar.

The individual toolbars are separated using a separator.

For all items in a toolbar a tooltip can be displayed.

Example of a common toolbar:



Available toolbars

Standard toolbars are divided into two groups:

- Main menu toolbars

  These toolbars contain the general, i.e. non-context sensitive symbols.

  All available main menu toolbars are displayed when starting the software.
- Context sensitive toolbars

  These toolbars contain the context sensitive symbols.

  The context sensitive toolbars are automatically shown in the active editor.

Depending on the screen resolution it is possible that not all toolbars are visible in the application window.

In this case you have the option to hide the non-required toolbars, or to reposition the toolbars.

Main menu toolbars

The following main menu toolbars are available by default:

Name	Contained functions
File	**New project**, **Open**, **Save**, **Print**
Edit	**Copy**, **Delete**, **Paste**, **Undo**, **Redo**, **Validate**, **Minimize**, **Maximize**, **Full screen**, **Go to**
Services	**Analyse project**, **Generate project**, **Re generate entire project**, **Project browser**, **Start search**, **Type library manager**
Api	**Transfer Project to PLC**, **Transfer Project from PLC**, **Connect**, **Disconnect**, **Start**, **Stop**, **Start/Stop Animation**, **Standard mode**, **Simulation mode**
Window	**Cascade**, **Tile Horizontally**, **Tile Vertically**
Help	**Help**, **What's This**
Debug	**Set Breakpoint**, **Clear Breakpoint**, **Start**, **Step Over**, **Step Into**, **Step Out**, **Show Current Step**, **Display call stack**
Watchpoint	**Set Watchpoint**, **Clear Watchpoint**, **Show Watchpoint**, **Synchronize Animation Table**, Refresh counter **Note:** This toolbar is only displayed when a programming language editor is active.
Project Browser	**Structural view**, **Functional view**, **Vertical views**, **Horizontal views**, **Zoom Out**

Showing and hiding

You have three options to hide a toolbar (to close it):

1. Right-click in the common toolbar and deselect the 'undesired' toolbar from the shortcut menu.
2. Open the **Customize** dialog box using **Tools > Customize** and deselect the 'undesired' toolbar.
3. For non-anchored toolbars with the close symbol (**x**).

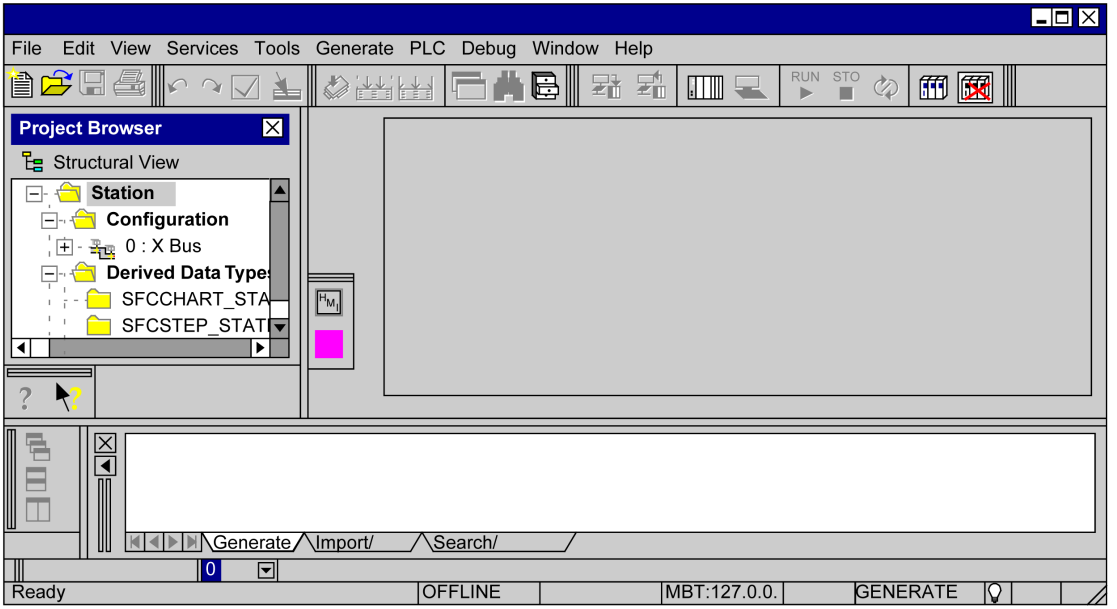
You have two options to show a toolbar (to open it):

1. Right-click in the common toolbar and select the 'desired' toolbar from the shortcut menu.
2. Open the **Customize** dialog box using **Tools > Customize** and select the 'desired' toolbar.

Possible positions

Toolbars can be anchored in the common toolbar (standard) or in different positions inside the application window. Toolbars can also be freely positioned on the screen (also outside of the application window).

Example of different toolbar positions:





Changing the position of a toolbar

Process the following steps to change the position of a toolbar:

Step	Action
1	Click the toolbar separator and hold the mouse button down.
2	Drag the toolbar to the target position inside or outside the common toolbar. **Result:** The window frame displays the intended position of the toolbar.
3	Release the mouse button. **Result:** The toolbar is moved to the target position. On the new position, the toolbar is • anchored, page 1827, when the new position of the toolbar is on the border or outside of the application window • not anchored, page 1827, when the new position of the toolbar is inside the application window or when the new position of the toolbar is on the border or outside of the application window, but the **Ctrl** key was held when releasing the mouse button

Modifying the contents of a toolbar

Process the following steps to modify the contents of a toolbar:

Step	Action
1	Open the **Customize** dialog box, see also Call the Dialog Box:, page 610.
2	Click the symbol to be moved in the source toolbar and hold the mouse button down.
3	Drag the symbol to the target position within the destination toolbar. **Result:** A frame displays the intended position in the toolbar.
4	Release the mouse button. **Result:** The symbol is removed from the source toolbar and inserted into the destination toolbar.
5	Click **OK** to confirm the modification(s). **Note:** If symbols are moved from context sensitive toolbars into the main menu toolbar, the symbols are inactive as long as the respective editor is inactive.

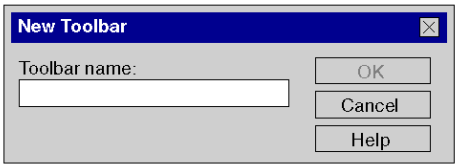
Restoring toolbars

Process the following steps to restore the original state of a toolbar:

Step	Action
1	Open the **Customize** dialog box, see also Call the Dialog Box:, page 610.
2	In the **Toolbars** list box, select the toolbar to be restored.
3	Click the **Reset** command button. **Result:** The original state of the toolbar is restored.

Creating your own toolbars

Process the following steps to create your own toolbars:

Step	Action
1	**Note:** New symbols cannot be created. Only existing symbols can be removed from toolbars and inserted into new toolbars. Open the **Customize** dialog box, see also Call the Dialog Box:, page 610.
2	Click **New...**. **Result:** A dialog for defining the toolbar name is opened. 
3	Enter a name and confirm with **OK**. **Result:** An empty toolbar is created and the name of the toolbar appears in the toolbar list.
4	Click the symbol to be moved in the source toolbar and hold the mouse button down.
5	Drag the symbol to the target position within the new toolbar. **Result:** A frame displays the intended position in the toolbar.
6	Release the mouse button. **Result:** The symbol is removed from the source toolbar and inserted into the destination toolbar.
7	Click **OK** to confirm the modification(s). **Note:** If symbols are moved from context sensitive toolbars into the main menu toolbar, the symbols are inactive as long as the respective editor is inactive.
8	Drag the new toolbar to the target position, see also Changing the position of a toolbar, page 1838.

Output Windows

Introduction

The output window displays information about the different processes (generate, import/export, user error, search/replace).

The output window is an anchored window, page 1827.

The output window is displayed at the bottom of the EcoStruxure Control Expert window by default.

If the output window is closed, it can be opened using:

- the **View > Output Window** menu command

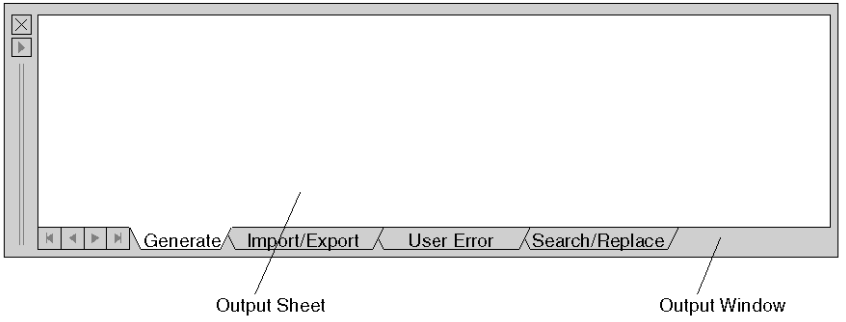
  or

- the **Alt**+**0** key combination.

Build

The output window consists of various output sheets. Each sheet corresponds to a tab.

Representation of an output window:



The output sheets display the results of an executed action or error messages.

Double-clicking an error message will open the respective editor and mark the erroneous entry.

New error entries are displayed in red, already checked errors are displayed in blue.

Available output sheets:

Output sheet	Function
Analyze and generate	Displays analyze and generate errors. Double-clicking an error entry directly accesses the erroneous programming language section, the erroneous configuration or the erroneous data.
Import/Export	Displays import and export errors Double-clicking an error entry directly accesses the error in the source file.
User error	Displays the user error.
Search/Replace	Displays the result of the search and/or replace operation. Double-clicking an entry directly accesses the searched for entry in the programming language section, in the configuration or in the data.
Comparison	Displays the comparison information.

Every output sheet has a shortcut menu (right-click).

Copy

Copies the selected text to the clipboard.

Delete

Deletes the entire contents of the output sheet.

Go to Previous Error

Jumps to the previous line and calls the erroneous programming language section, the erroneous configuration or the erroneous data.

This function can also be accessed pressing the **Ctrl**+**F10** key combination.

Go to Next Error

Jumps to the next line and calls the erroneous programming language section, the erroneous configuration or the erroneous data.

This function can also be accessed pressing the **F10** key.

Go to Error

Calls the erroneous programming language section, the erroneous configuration or the erroneous data as indicated in the current line.

Print

Prints the contents of the output sheet.

Status bar

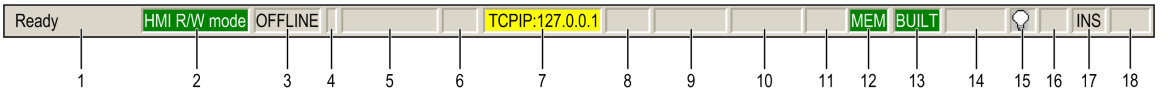
Introduction

The status bar displays information about the current project on the PC, about the PLC and about the software status.

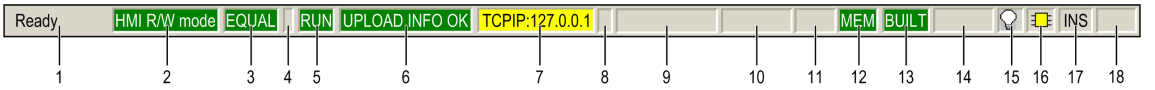
Status Bar Description with Examples

Examples of status bar in various configurations.

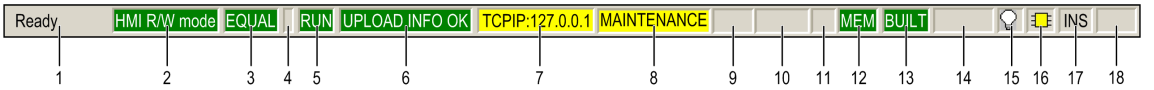
Offline status bar:



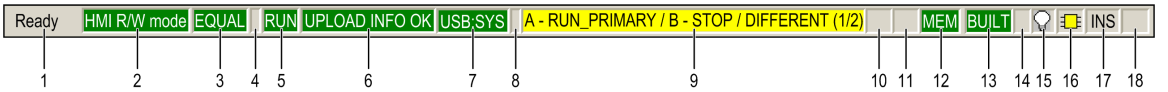
Online status bar:



Online M580 Safety status bar:



Online M580 Hot Standby status bar:



Status bar fields description:

Label	Description	Values
1	Information area for displaying menu information, FFB instances information (type and name), variables information (type, name and if available: address, comment and Alias of).	Example: **Ready**
2	Displays the current HMI access rights. An application can have one of the following modes:	
	• Human Machine Interface Read/Write mode This is the default mode when opening an application.	**HMI R/W mode**
	• Human Machine Interface Read only mode. This mode is active in the following cases: ◦ Applications are opened in Read Only mode by selecting the option **Open the project in read-only mode** in the open file window. ◦ Opening an application previously opened in write mode with a third-party software (using Pserver). A pop-up window alerts the user at the opening. This mode does not allow to save the application. When using third-party application with Pserver, Control Expert can be launched without the Control Expert graphical interface (called Human Machine Interface). In this case, a dialog box prompts the user if it should switch on the current PServer Instance.	**HMI R/O mode**
3	Link status.	**OFFLINE**, **DIFFERENT**, **EQUAL**
4	An ***** is displayed when the initial value in the project is different/modified from the initial value on the CPU. To remove the ***** it is advisable to **Update Local Init Values with PLC Init Values**, or to **Transfer Project from PLC** and save (if parameters have been modified in the adjust screens of specific modules).	–, *****
5	CPU state.	**RUN**, **STOP**, **NO CONF**, ...
6	ETSI information (Empty terminal Support)	Example: **UPLOAD INFO OK**
7	Type of connection and address of the connected CPU.	Example: **TCPIP:127.0.0.1**

Label	Description	Values
8	For M580 Safety platform only: operating mode status. More details on M580 Safety platform operating modes are provided in *Modicon M580, Safety Manual*.	**MAINTENANCE**, **SAFETY**
9	For M580 platform only: Hot Standby status and number of changes.	Example: **A - RUN_PRIMARY / B - STOP / DIFFERENT (1/2)**
10	Line and column information (only available in the programming language editor).	Example: **In 13, Col 15**
11	Syslog server availability (when configured):	
	• Syslog server available.	–
	• Syslog server cannot be reached.	**SYSLOG**
12	Memory status:	
	• **Function Pack for Memory** is not needed.	**MEM** (green)
	• **Function Pack for Memory** command is recommended.	**MEM** (red)
13	Generate status.	**BUILT**, **NOT BUILT**
14	In the connected equal mode, a red F displayed indicates that the bits or words have been forced in the CPU. Click this field to display the animation table containing the list of variables (bits/words) forced.	–, **F**
15	Event information.	
	• If a CPU detects an error, click this field for details about the detected error.	
	• Icon displayed when the CPU detects a new error.	
16	For M580 and M340 platform only, memory backup status:	
	• CPU RAM application equals flash memory and/or SD card content.	–
	• M580 platform only: flash memory backup in progress. **NOTE:** Do not click this icon when displayed. If clicked, a popup proposes to back up the application: do not accept the backup proposal because a backup is already in progress.	
	• SD card memory backup in progress.	
	• No SD card in the CPU or invalid SD card.	
17	Indicates whether the insert or overwrite mode is active.	**INS**, **OVR**

Label	Description	Values
18	Indicates whether the caps lock button is active or not.	–, **CAPS**
– Empty field.		

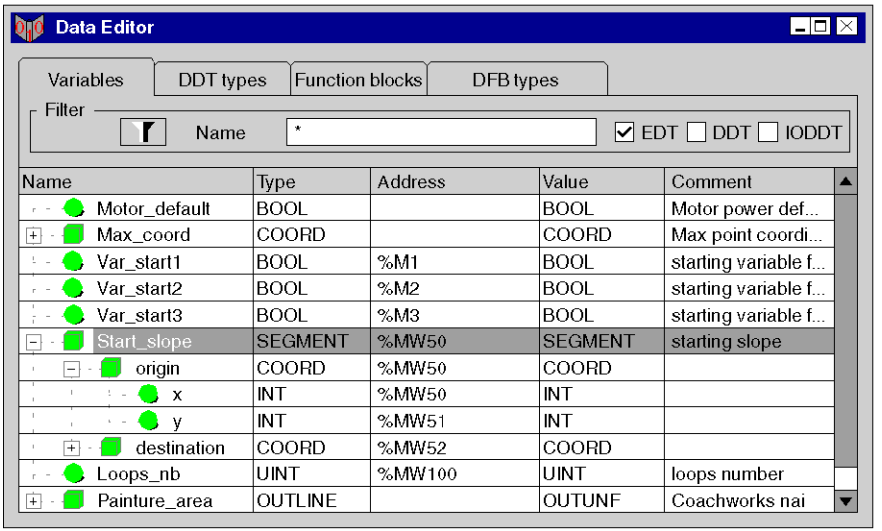
Directory Structures

Introduction

Directory structures are used to display and edit hierarchical lists with several columns. Directory structures are combinations of directory trees and lists.

Example of use

In the Data Editor, directory structures are used to display the hierarchy between variable instances, variable types, structure elements, etc.



Meaning of the text colors

The text colors have the following meaning:

Text color	Meaning
black	Information is correct and can be edited
gray	Information is correct and write protected
red	Information is incorrect

Modes

There are two different modes for directory structures:

- **Select mode**

  One or more lines are selected or a cell is selected.

  In this mode, you can move between all lines and cells.

- **Edit mode**

  The current cell is opened for editing.

  Texts can be edited.

Switching from the selection mode to the edit mode is done by pressing the **Enter** key or by entering a character, see the **Open editing session by holding down the character key** check box in the Data and Languages, page 598 dialog box.

Switching from edit mode to selection mode is done by pressing the **Enter** key or by pressing an error key.

Using the Mouse

Mouse Functions

The following mouse functions are available:

Mouse function	In select mode	In edit mode
Click on links	• Click +/- button: Open/close the substructure. • Click on cell: Selects the lines in the cell clicked and the cell becomes the current cell. • Click on check box: Activate/deactivate the check box • Click on icon: -	• Click +/- button: Saves the information entered (if valid) and opens/closes the substructure. • Click on cell: Saves the information entered (if valid) and selects the lines in the cell clicked. • Click on check box: Saves the information entered (if valid) and activates/deactivates the check box • Click on icon: Saves the information entered (if valid).
Double-click on links	• Double-click on names: Selects the lines in the cell clicked and opens the cell in edit mode. • Double-click on the vertical line in the column header: Column width is adjusted to the current contents. • Double-click on the column header cell: Lists are sorted according to column contents	• Double-click on names: Saves the information entered (if valid) and selects the lines in the cell clicked und opens the cell in edit mode. • Double-click on the vertical line in the column header: Saves the information entered (if valid) and the column width is adjusted to the current contents. • Double-click on the column header cell: Saves the information entered (if valid) and lists will be sorted according to the column contents.
SHIFT+Click on links	Selects consecutive elements. The first and last element selected and all elements in between are marked.	Saves the information entered (if valid) and selects the lines in the cell clicked.
Right click	Opens the shortcut menu (if available) for the object currently being clicked.	Saves the information entered (if valid) and selects the lines in the cell clicked.

Using the Keyboard

Key functions

The following key functions are available:

Key function	In select mode	In edit mode
Enter	Opens the current cell in edit mode	Saves the information entered (if valid).
Esc	-	Ends edit mode without saving.
Up arrow	Deselects the current cell and selects the previous cell (upward).	Saves the information entered (if valid), deselects the current cells and selects the previous cell (upwards).
Cursor down	Deselects the current cell and selects the next cell (downwards).	Saves the information entered (if valid), deselects the current cells and selects the previous cell (downwards).
Left arrow	Deselects the current cell and selects the previous cell (left).	Saves the information entered (if valid), deselects the current cells and selects the previous cell (to the left).
Right arrow	Deselects the current cell and selects the next cell (right).	Saves the information entered (if valid), deselects the current cells and selects the next cell (to the right).
Shift+Up arrow	Selects multiple lines in upward direction.	-
Shift+Down arrow	Selects multiple lines in downward direction.	-
+	Open the substructure.	-
-	Close the substructure.	-
Character	Goes to the next lines in which the contents begin with that character. Exceptions: **+**, **-** and **space**.	-
F2	Opens the current cell in edit mode	-
Insert	Adds an empty line before the selected line.	-
Home	The first cell in the selected line becomes the current cell.	Sets the insertion point at the beginning of the current line.
Ctrl+Home	Selects the first line in the table and the first cell becomes the current cell.	Sets the insertion point at the beginning of the first line.
End	The last cell in the selected line becomes the current cell.	Sets the insertion point at the end of the current line.
Ctrl+End	Selects the last line in the table and the first cell becomes the current cell.	Sets the insertion point at the end of the last line.

Data selection boxes

Introduction

Data selection boxes are used to select compatible, already defined data from a list or to select from a list of previous entries.

In this case, "data" means instances and types of variables and instances and types of functions and function blocks.

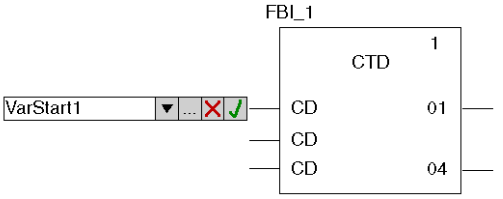
Build

Data selection boxes consist of:

- a combined text selection and text entry field
- a button to open the selection list (▼)
- a button to open the data selection dialog box (**...**)
- a button to close the data selection dialog box and delete the (✕) data
- a button to close the data selection dialog box and transfer the (✓) data

Examples of use

In the language editors, data selection boxes are used to select variables with the correct data type.



Data entry

There are three options for data entry:

1. Enter the name if known or enter the first letter(s) in the text box.

   The system works with a context dependent background list which automatically completes the characters entered with the first compatible name.

2. Selecting the name in the list box.

   When entering the first letters of the name being searched in the text box, the list box receives an alphabetic list of names that correspond to this entry.

   Without entering the first letters, the list box contains a list of previous entries.

3. Open the data selection dialog box using the (**...**).

Accepting data

With the **Enter** key or the ✓ button, the entry is acknowledged and accepted after having been confirmed successfully.

Cancel

The entry can be canceled using the **Esc** key or the ✕ button.

Dialog boxes

Introduction

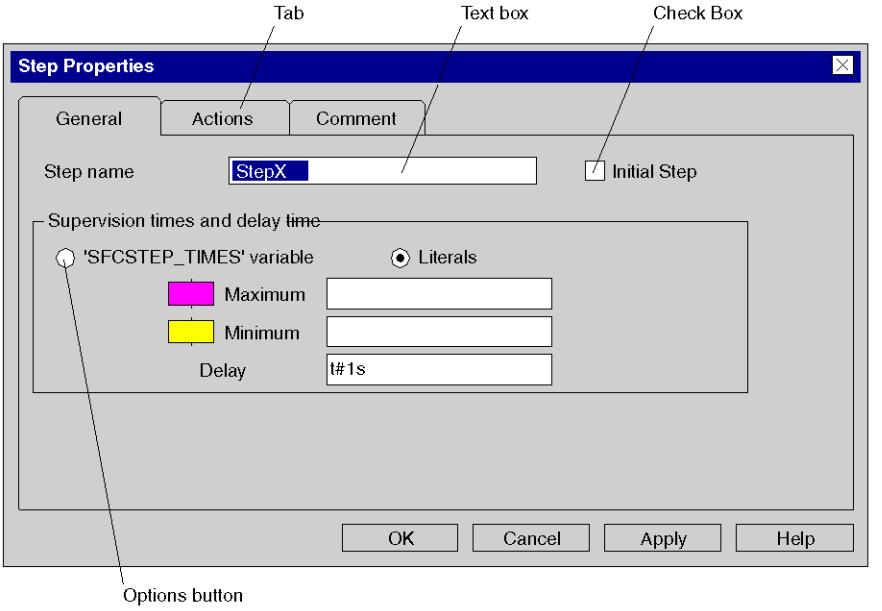
Dialog boxes are displayed if additional information is required from you in order to perform a particular task. Information that you may require is also communicated in this way.

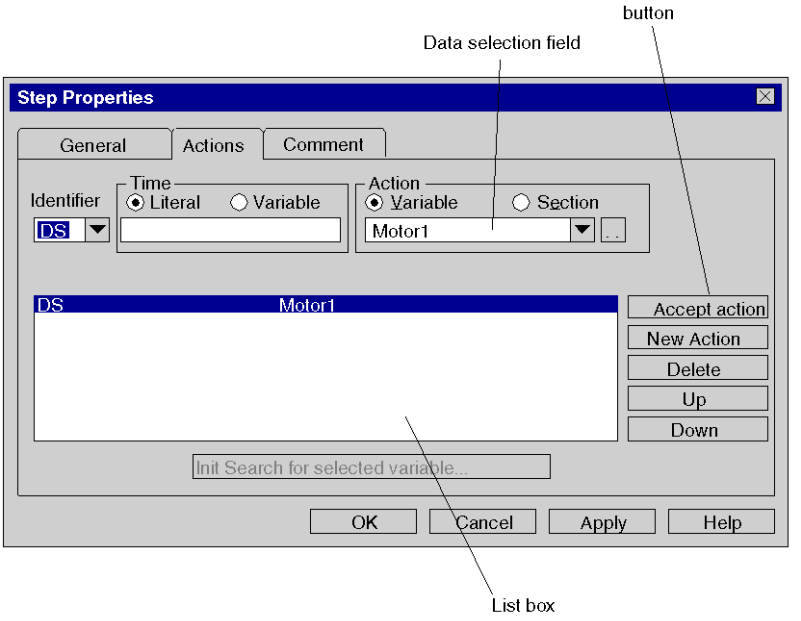
Most dialog boxes contain options which can be selected, text boxes in which text can be entered, and buttons which can be pressed.

Grayed out options are currently not available. One or more other commands must be executed, or options selected or deselected, before the desired option can be activated.

The information contained in dialog boxes is separated into different areas. The different area in a dialog box can be selected using the **F6** key.

Basic elements of a window:





Tab

If the amount of information contained in a dialog box is too complex to be displayed in one dialog box, the dialog box is split into several tabs.

The different tabs in a dialog box can be selected using the **Ctrl**+**Tab** key combination.

button

buttons are used to initiate actions immediately, e.g. executing or aborting a command. buttons are e.g. **OK**, **Cancel** and **Apply**.

buttons followed by three dots (**...**), open another dialog box. A button with a "greater than" sign (**>>**) extends the active dialog box.

The default setting is identified by a darker border. This default button can also be selected by pressing **Enter**.

If the name of the button contains an underlined letter, the button can be activated from any position in the dialog box by holding down the **Alt** key and entering the underlined letter.

To close a dialog box without executing a command, select the **Cancel** button.

Text box

Information (text) is entered in a text box.

If you go to an empty text box the cursor appears in the far left of the box. The entered text begins at the cursor position. If text is already present within the respective text box, the text will be selected and replaced by the new text automatically. The text can, however, also be deleted by pressing the **Del** or **backspace** key.

If the name of the text box contains an underlined letter, the text box can be activated from any position in the dialog box by holding down the **Alt** key and entering the underlined letter.

List box

The available options to select are given in list form in a list box. If more options are available than fit into the list, then the scrollbar or arrow keys can be used to move around the list.

Normally only a single entry can be chosen form the list. There are, however, some cases in which several entries can be chosen, e.g. when opening sections.

If the name of the list box contains an underlined letter, the list box can be activated from any position in the dialog box by holding down the **Alt** key and entering the underlined letter.

Single line list boxes

A single line list box initially appears as a rectangular box, in which the current selection (the default value) is selected. If the arrow in the right of the box is selected, a list of the available options to select opens. If more options are available than fit into the list, then the scrollbar or arrow keys can be used to move around the list.

If the name of the list box contains an underlined letter, the list box can be activated from any position in the dialog box by holding down the **Alt** key and entering the underlined letter.

Data selection boxes

See section Data selection boxes, page 1850

Directory structures

See section Directory Structures, page 1846

Option buttons

Option buttons represent mutually exclusive options. In each case only one option can be chosen.

The selected option button is identified by a black dot.

If the option name contains an underlined letter, the option button can be activated from any position in the dialog box by holding down the **Alt** key and entering the underlined letter.

Check boxes

A checkbox next to an option means that the option can be activated or deactivated. Any number of checkbox options can be activated.

Activated options are identified by an **X** or a check mark (√).

If the name of the option contains an underlined letter, the checkbox can be activated from any position in the dialog box by holding down the **Alt** key and entering the underlined letter.

Keyboard Shortcuts

What's in This Chapter

General Keyboard Shortcuts 1857
Configuring the Project 1860
Creating the Project 1861
Operating the Project 1870
Documenting the Project 1874

Overview

This chapter describes the keyboard shortcuts of the Control Expert program.

General Keyboard Shortcuts

Object of this section

This section provides general information about commonly used keyboard shortcuts. It also provides information about a widely-used browser, namely the Project Browser.

Bus Editor Keyboard Shortcuts

Introduction

Numerous actions or commands can be performed with the mouse as well as the keyboard. The shortcuts that you can use in the Bus Editor are described in the table below. Furthermore, most of the general keyboard shortcuts are available in the General Keyboard shortcuts, page 1858 map.

Keyboard Shortcuts

This table describes the Bus Editor keyboard shortcuts.

Action	Menu	Keyboard Shortcut
Open the New Device windows	Edit	Ctrl+Enter
Open the PLC Configuration windows	Edit	Ctrl+Alt+Enter

Control Expert Main Screen and Project Browser Keyboard Shortcuts

Introduction

Numerous actions or commands can be performed with the mouse as well as the keyboard. The shortcuts that you can use in the Control Expert main screen and in the Project Browser when a project is opened are described in the table below.

Keyboard Shortcuts

This table describes the Control Expert main screen and the Project Browser keyboard shortcuts when a project is opened:

Action	Menu	Keyboard Shortcut
Create a project	File	Ctrl+N
Open an existing project	File	Ctrl+O
Save an existing project	File	Ctrl+S
Print part of an existing project	File	Ctrl+P
Display properties	Edit	Alt+Enter
Display a full screen	View	Ctrl+F8
Undo the preceeding action	Edit	Ctrl+Z
Redo the preceeding action	Edit	Ctrl+Y
Cut an item	Edit	Ctrl+X
Copy an item	Edit	Ctrl+C
Select all an item	Edit	Ctrl+A
Paste an item	Edit	Ctrl+V
Delete an item	Edit	Del
Show or hide the ouput windows	View	Alt+0

Action	Menu	Keyboard Shortcut
Animate the project	Services	Ctrl+J
Initialize Search	Services	Ctrl+U
Create an animation table	Services	Ctrl+T
Display the Project Browser	Tools	Alt+1
Display the Hardware Catalog	Tools	Alt+2
Display the Types Library Browser	Tools	Alt+3
Display the Operator Screen Library	Tools	Alt+4
Search for Cross-References	Tools	Alt+5
Display the Diagnostic Viewer	Tools	Alt+6
Display the PLC Screen	Tools	Alt+7
Display the Variable Window	Tools	Alt+8
Open the Data Editor	Tools	Alt+9
Insert	Edit	Ins
Display the Data Properties	Edit	Ctrl+Enter
Analyze a Type	Build	Ctrl+Shift+B
Control the Changes	Build	Ctrl+B
Connect or disconnect a PLC	PLC	Ctrl+K
Transfer a project to PLC	PLC	Ctrl+L
Transfer a project from a PLC	PLC	Ctrl+Shift+L
Put a Quantum Safety PLC in maintenance or safety mode	PLC	Ctrl+Shift+M
Run or stop a PLC	PLC	Ctrl+R
What's this	Help	Shift+F1

NOTE: When a project is closed, the Tools and PLC tab shortcuts in the PLC menu are not available but it is still possible to use the Ctrl+K shortcut. The shortcuts from the File, View and Help tab shortcuts are available.

NOTE: Commands without shortcuts are not mentionned in this table.

Configuring the Project

Object of this section

This section provides general information about the keyboard shortcuts used to configure a project.

PLC Configuration Screen Keyboard Shortcuts

Introduction

Numerous actions or commands can be performed with the mouse as well as the keyboard. The shortcuts that you can use in the PLC Configuration Screen are described in the table below. Furthermore, most of the general keyboard shortcuts are available in the General Keyboard shortcuts, page 1858 map.

Keyboard Shortcuts

This table describes the PLC Configuration Screen keyboard shortcuts.

Action	Menu	Keyboard Shortcut
Validate the action	Edit	Ctrl+W
Copy an input from the list	Edit	Ctrl+C
Paste an input in the list	Edit	Ctrl+V
Delete an AS_I Slave	Edit	Del
Cut an AS_I Slave	Edit	Ctrl+X
Copy an AS_I Slave	Edit	Ctrl+C
Paste an AS_I Slave	Edit	Ctrl+V
Delete a Series 7 module	Edit	Del
Cut a Series 7 module	Edit	Ctrl+X
Copy a Series 7 module	Edit	Ctrl+C
Paste a Series 7 module	Edit	Ctrl+V

NOTE: The Delete, cut, copy paste AS_I Slave shortcuts are only available with Premium Communication TSX SAY 100x modules.

NOTE: The Delete, cut, copy paste Series 7 modules shortcuts are only available with Premium Communication TSX ESY 007 modules. Furthermore, these shortcuts are available if the user has specific access rights. Otherwise, the access is denied.

Creating the Project

Object of this section

This section provides information about the keyboard shortcuts used to create a project.

ST/IL Editors Keyboard Shortcuts

Introduction

Numerous actions or commands can be performed with the mouse as well as the keyboard. The shortcuts that you can use in the ST/IL Editors are described in the table below. Furthermore, most of the general keyboard shortcuts are available in the General Keyboard shortcuts, page 1858 map.

Keyboard Shortcuts

This table describes the ST/IL Editors keyboard shortcuts.

Action	Menu	Keyboard Shortcut
Select data from a list	Edit	Ctrl+D
Select the Function Input Assistant	Edit	Ctrl+I
Create an IF Statement	Edit/New	F4
Create a FOR Statement	Edit/New	F5
Create a WHILE Statement	Edit/New	F6
Create a REPEAT Statement	Edit/New	Shift+F6
Create a CASE Statement	Edit/New	Shift+F4
Create a variable	Edit/New	Shift+Enter
Edit the selected text in Upper case	Edit	Shift+Alt+U
Edit the selected text in lowercase	Edit	ALt+U

Action	Menu	Keyboard Shortcut
Find the selected text	Edit	Ctrl+F
Find the next selected occurrence	Edit	F3
Find the previous selected occurrence	Edit	Shift+F3
Replace the selected occurrence	Edit	Ctrl+H
Go to part of the selected project	Edit	Ctrl+G
Add a bookmark	Edit/Bookmark	Ctrl+F2
Go to next bookmark	Edit/Bookmark	F2
Go to previous bookmark	Edit/Bookmark	Shift+F2
Delete all bookmarks	Edit/Bookmark	Ctrl+Shift+F2
Display the general attributes of the selected item	Edit	Ctrl+Enter
Modify the properties of the selected item	Edit	Alt+Enter
Add a comment	Edit/New	F8
Open the Inspect window	Edit/New	F9
Zoom in	View/Zoom	• Ctrl+Plus (+) • Ctrl+Rotate the mouse wheel
Zoom out	View/Zoom	• Ctrl+Minus (-) • Ctrl+Rotate the mouse wheel
Show/Hide the Inspect window	View	Shift+F8
Go to the previous section	View	Ctrl+Alt+Pg Up
Go to the next section	View	Ctrl+Alt+Pg Dn
Refine an object	Services	Ctrl+Q
Set breakpoint	Debug	Ctrl+F11
Clear breakpoint	Debug	Ctrl+Shift+F11
Go to the selected breakpoint	Debug	Ctrl+F5
Step over the next action	Debug	F11
Step into the action	Debug	F12
Step out the action	Debug	Shift+F12
Set watchpoint	Debug	Alt+F11
Clear watchpoint	Debug	Alt+Shift+F11

NOTE: The IF, FOR, WHILE, REPEAT, CASE statement shortcuts are only available with the ST Editor.

FBD Editor Keyboard Shortcuts

Introduction

Numerous actions or commands can be performed with the mouse as well as the keyboard. The shortcuts that you can use in the FBD Editor are described in the table below. Furthermore, most of the general keyboard shortcuts are available in the General Keyboard shortcuts, page 1858 map.

Keyboard Shortcuts

This table describes the FBD Editor keyboard shortcuts.

Action	Menu	Keyboard Shortcut
Select the Function Input Assistant	Edit	Ctrl+I
Create a variable	Edit	Shift+Enter
Select data from a list	Edit	Ctrl+D
Link an item	Edit/New	F6
Add a comment	Edit/New	F8
Open the Inspect window	Edit/New	F9
Select Mode	Edit	Esc
Add a bookmark	Edit/Bookmark	Ctrl+F2
Go to next bookmark	Edit/Bookmark	F2
Go to previous bookmark	Edit/Bookmark	Shift+F2
Delete all bookmarks	Edit/Bookmark	Ctrl+Shift+F2
Zoom to fit	View/Zoom	Ctrl +* (* = multiplication sign of numeric pad)
Zoom in	View/Zoom	• Ctrl+Plus (+) • Ctrl+Rotate the mouse wheel
Zoom out	View/Zoom	• Ctrl+Minus (-)

Action	Menu	Keyboard Shortcut
		• Ctrl+Rotate the mouse wheel
Show/Hide the Inspect window	View	Shift+F8
Go to the previous section	View	Ctrl+Alt+Pg Up
Go to the next section	View	Ctrl+Alt+Pg Dn
Refine an object	Services	Ctrl+Q
Set breakpoint	Debug	Ctrl+F11
Clear breakpoint	Debug	Ctrl+Shift+F11
Go to the selected breakpoint	Debug	Ctrl+F5
Step over the next action	Debug	F11
Step into the action	Debug	F12
Step out the action	Debug	Shift+F12
Set watchpoint	Debug	Alt+F11
Clear watchpoint	Debug	Alt+Shift+F11
Properties	Edit	Alt+Enter
Data Properties	Edit/New	Ctrl+Enter
Data Selection	Edit/New	Ctrl+D
FFB Input Assistant	Edit/New	Ctrl+I

NOTE: Most of the Edit tab shortcuts are available by selecting Shift+F10.

SFC Editor Keyboard Shortcuts

Introduction

Numerous actions or commands can be performed with the mouse as well as the keyboard. The shortcuts that you can use in the SFC Editor are described in the table below. Furthermore, most of the general keyboard shortcuts are available in the General Keyboard shortcuts, page 1858 map.

Keyboard Shortcuts

This table describes the STB Editor keyboard shortcuts.

Action	Menu	Keyboard Shortcut
Add a step	Edit/New	F3
Add a macro step	Edit/New	Ctrl+F3
Add a transition	Edit/New	Shift+F3
Add an alternative branch	Edit/New	F4
Add an alternative joint	Edit/New	Shift+F4
Add a parallel branch	Edit/New	F5
Add a parallel joint	Edit/New	Shift+F5
Add a link	Edit/New	F6
Add a comment	Edit/New	F8
Select Mode	Edit	Esc
Add a bookmark	Edit/Bookmark	Ctrl+F2
Go to next bookmark	Edit/Bookmark	F2
Go to previous bookmark	Edit/Bookmark	Shift+F2
Delete all bookmarks	Edit/Bookmark	Ctrl+Shift+F2
Zoom to fit	View/Zoom	Ctrl +* (* = multiplication sign of numeric pad)
Zoom in	View/Zoom	• Ctrl+Plus (+) • Ctrl+Rotate the mouse wheel
Zoom out	View/Zoom	• Ctrl+Minus (-) • Ctrl+Rotate the mouse wheel
Expand/come back to the project display	View	Ctrl+E
Go to the previous section	View	Ctrl+Alt+Pg Up
Go to the next section	View	Ctrl+Alt+Pg Dn
Initialize Animation Table	Services	Ctrl+T
Initialize Search	Services	Ctrl+U

Action	Menu	Keyboard Shortcut
Data Properties	Edit/New	Ctrl+Enter
Properties	Edit	Alt+Enter

NOTE: Most of the Edit table shortcuts are available by selecting Shift+F10.

LD Editor Keyboard Shortcuts

Introduction

Numerous actions or commands can be performed with the mouse as well as the keyboard. The shortcuts that you can use in the LD Editor are described in the table below. Furthermore, most of the general keyboard shortcuts are available in the General Keyboard shortcuts, page 1858 map.

Keyboard Shortcuts

This table describes the LD Editor keyboard shortcuts.

Action	Menus	Keyboard Shortcut
Add a normally open contact	• **Edit > New** • **Objects**	F3
Add a closed contact	• **Edit > New** • **Objects**	Shift+F3
Add a positive transition-sensing contact	• **Edit > New** • **Objects**	Ctrl+F3
Add a negative transition-sensing contact	• **Edit > New** • **Objects**	Ctrl+Shift+F3
Add a coil	• **Edit > New** • **Objects**	F5
Add a negated coil	• **Edit > New** • **Objects**	Shift+F5
Add a set coil	• **Edit > New** • **Objects**	Alt+F5
Add a reset coil	• **Edit > New**	Shift+Alt+F5

Action	Menus	Keyboard Shortcut
	• **Objects**	
Add a call coil	• **Edit > New** • **Objects**	F4
Add a boolean connection	• **Edit > New** • **Objects**	F7
Add a vertical connection	• **Edit > New** • **Objects**	Shift+F7
Add a boolean link	• **Edit > New** • **Objects**	Alt+F6
Add a link	• **Edit > New** • **Objects**	F6
Add an operate block	• **Edit > New** • **Objects**	Alt+F7
Add a compare block	• **Edit > New** • **Objects**	Ctrl+F7
Add a comment	• **Edit > New** • **Objects**	F8
Select mode	• **Edit** • **Objects**	Esc
Add a bookmark	**Edit > Bookmark**	Ctrl+F2
Go to next bookmark	**Edit > Bookmark**	F2
Go to previous bookmark	**Edit > Bookmark**	Shift+F2
Delete all bookmarks	**Edit > Bookmark**	Ctrl+Shift+F2
Modify the properties of the selected item	**Edit**	Alt+Enter
Show or hide the status bar	**View**	Alt+0
Zoom to fit	**View > Zoom**	Ctrl +* (* = multiplication sign of numeric pad)
Zoom in	**View > Zoom**	• Ctrl+Plus (+) • Ctrl+Rotate the mouse wheel
Zoom out	**View > Zoom**	• Ctrl+Minus (-)

Action	Menus	Keyboard Shortcut
		• Ctrl+Rotate the mouse wheel
Show/Hide the Inspect window	**View**	Shift+F8
Display the mixed mode	**View**	Ctrl+E
Go to the previous section	**View**	Ctrl+Alt+Pg Up
Go to the next section	**View**	Ctrl+Alt+Pg Dn
Set breakpoint	**Debug**	Ctrl+F11
Clear breakpoint	**Debug**	Ctrl+Shift+F11
Go to the selected breakpoint	**Debug**	Ctrl+F5
Step over the next action	**Debug**	F11
Step into the action	**Debug**	F12
Step out the action	**Debug**	Shift+F12
Set watchpoint	**Debug**	Alt+F11
Clear watchpoint	**Debug**	Alt+Shift+F11
Inspect Window	• **Edit** • **Objects**	F9
Data Selection	**Edit**	Ctrl+D
FFB Input Assistant	**Edit**	Ctrl+I
Properties	**Edit**	Alt+Enter
Data Properties	**Edit > New**	Ctrl+Enter
Refine	**Services**	Ctrl+Q

NOTE: Most of the Edit tab shortcuts are available by selecting Shift+F10.

Data Editor Keyboard Shortcuts

Introduction

Numerous actions or commands can be performed with the mouse as well as the keyboard. The shortcuts that you can use in the Data Editor are described in the table below. Furthermore, most of the general keyboard shortcuts are available in the General Keyboard shortcuts, page 1858 map.

Keyboard Shortcuts

This table describes the Data Editor keyboard shortcuts.

Action	Menu	Keyboard Shortcut
Hide the Inspect window	View	Shift+F8

NOTE: Some shortcuts are not available from the menu bar. Please note the following shortcuts.

- Select F2 to edit data
- Select Ctrl+ Home to access the first cell in row
- Select Ctrl+ End to access the last cell in row
- Select + on a node to expand it
- Select - on a node to contract it

Type Library Browser Keyboard Shortcuts

Introduction

Numerous actions or commands can be performed with the mouse as well as the keyboard. The shortcuts that you can use in the Type Library Browser are described in the table below. Furthermore, most of the general keyboard shortcuts are available in the General Keyboard shortcuts, page 1858 map.

Keyboard Shortcuts

This table describes the Type Library Browser keyboard shortcuts.

Action	Menu	Keyboard Shortcut
Refine an object	Services	Ctrl+Q
Initialize search	Services	Ctrl+U

Type Library Manager Keyboard Shortcuts

Introduction

Some actions or commands can be performed with the keyboard. The shortcuts that you can use in the Type Library Manager are described in the table below. Furthermore, most of the general keyboard shortcuts are available in the General Keyboard shortcuts, page 1858 map.

Keyboard Shortcuts

This table describes the Type Library Manager keyboard shortcuts.

Action	Keyboard Shortcut
Access the first cell in row	Home
Access the last cell in row	End
Access the next cell up	Up
Access the next cell down	Down
Access next cell left	Left
Access next cell right	Right
Expand a node	+
Contract a node	-
Exit the project	Alt+F4/Esc
Delete data	Delete
Selected Several EFB	Shift+Left

Operating the Project

Object of this section

This section provides information about the keyboard shortcuts used to operate a project.

Search Replace Tool Screen Keyboard Shortcuts

Introduction

Numerous actions or commands can be performed with the mouse as well as the keyboard. The shortcuts that you can use in the Search Replace Tool Screen are described in the table below. Furthermore, most of the general keyboard shortcuts are available in the General Keyboard shortcuts, page 1858 map.

Keyboard Shortcuts

This table describes the Search Replace Tool Screen keyboard shortcuts.

Action	Menu	Keyboard Shortcut
Find Next item	Edit	F3
Find Previous item	Edit	Shift+F3
Replace an item	Edit	Ctrl+H

NOTE: Most of the Edit tab shortcuts are available by selecting Shift+F10.

Diagnostic Viewer Screen Keyboard Shortcuts

Introduction

Numerous actions or commands can be performed with the mouse as well as the keyboard. All the shortcuts that you can use in the Diagnostic Viewer Screen are described in the General Keyboard Shortcut General Keyboard shortcuts, page 1858 map.

PLC Screen Keyboard Shortcuts

Introduction

Numerous actions or commands can be performed with the mouse as well as the keyboard. All the the shortcuts that you can use in the PLC Screen are described in theGeneral Keyboard Shortcuts General Keyboard shortcuts, page 1858 map.

Animation Screen Keyboard Shortcuts

Introduction

Numerous actions or commands can be performed with the mouse as well as the keyboard. The shortcuts that you can use in the Animation Screen are described in the table below. Furthermore, most of the general keyboard shortcuts are available in the General Keyboard shortcuts, page 1858 map.

Keyboard Shortcuts

This table describes the Animation Screen keyboard shortcuts.

Action	Menu	Keyboard Shortcut
Display format in decimal	View/Display Format	F3
Display format in hexadecimal	View/Display Format	Ctrl+F3
Display format in binary	View/Display Format	Shift+F3
Display format in ASCII	View/Display Format	Alt+F3
Enable Variable Modification	Services	F7
Enable Variable Forcing	Services	F6
Set value to 0	Services	Shift+F5
Set value to 1	Services	F5
Force to 0	Services	Shift+F4
Force to 1	Services	F4
Unforce	Services	Shift+F6
Multiple Mode	Services	Ctrl+F7
Execute Multiple Values	Services	Alt+F7
Reset Multiple Values	Services	Shift+F7
Safety/Maintenance	PLC	Ctrl+Shift+M

NOTE: Most of the Edit tab shortcuts are available by selecting Shift+F10.

NOTE: This shortcut "Ctrl+Shift+M" is only available with XLS

Operator Screen Keyboard Shortcuts

Introduction

Numerous actions or commands can be performed with the mouse as well as the keyboard. The shortcuts that you can use in the Operator Screen are described in the table below. Furthermore, most of the general keyboard shortcuts are available in the General Keyboard shortcuts, page 1858 map.

Keyboard Shortcuts

This table describes the Operator Screen keyboard shortcuts.

Action	Menu	Keyboard Shortcut
Validate	Edit	Ctrl +W
Select all	Edit	Ctrl+A
Enable variation modification	Services	F7
Go to the previous animated object	Services	F4
Go to the next animated object	Services	F5

NOTE: Some shortcuts are not available from the menu bar. Please note the following shortcuts.

- Select Space to select next item
- Select Shift+ Space to select the previous item
- Select Alt+ Backspace to cancel thelast action
- Select Tab to focus on the next control object in online mode
- Select Shift+ Tab to focus on the previous control object in online mode
- Select Enter to display the properties of a selected object
- Select an Arrow to make a one pixel move in the arrow direction
- Select Shift+ Arrow to make a 10 pixel move in the arrow direction
- Select Page Up/Down to move the vertical scrollbar up or down
- Select Ctrl + Left/Right to move the horizontal scrollbar left or right

Documenting the Project

Object of this section

This section provides information about the keyboard shortcuts used to document a project.

Documentation Screen Keyboard Shortcuts

Introduction

Numerous actions or commands can be performed with the mouse as well as the keyboard. The shortcuts that you can use in the Documentation Screen are described in the table below. Furthermore, most of the general keyboard shortcuts are available in the General Keyboard shortcuts, page 1858 map.

Keyboard Shortcuts

This table describes the Documentation Screen keyboard shortcuts.

Action	Menu	Keyboard Shortcut
Validate the action	Edit	**Ctrl**+**W**

NOTE: Some shortcuts are not available from the menu bar. Please note the following shortcuts.

- Select F5 to refresh a project
- Select **Ctrl**+**Plus (+)** to zoom in a project
- Select **Ctrl**+**Plus (-)** to zoom out of a project
- Select **Ctrl**+**E** to zoom to fit

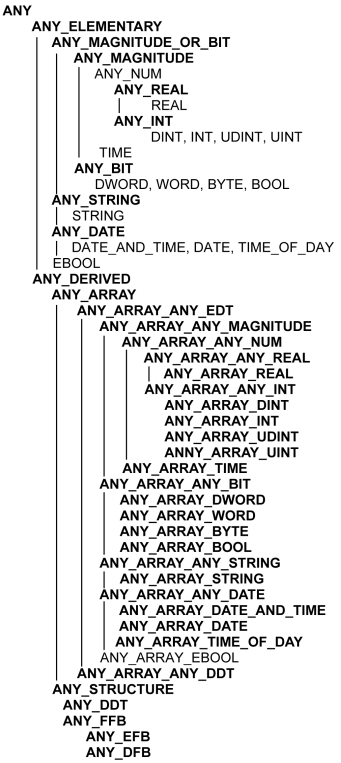
Glossary

A

ANY:

There is a hierarchy between the different types of data. In the DFB, it is sometimes possible to declare which variables can contain several types of values. Here, we use `ANY_xxx` types.

The following diagram shows the hierarchically-ordered structure:



B

BCD:

The Binary Coded Decimal (BCD) format is used to represent decimal numbers between 0 and 9 using a group of four bits (half-byte).

In this format, the four bits used to code the decimal numbers have a range of unused combinations.

Example of BCD coding:

- the number `2450`
- is coded: `0010 0100 0101 0000`

BOOL:

`BOOL` is the abbreviation for Boolean type. This is the elementary data item in computing. A `BOOL` type variable has a value of either: 0 (`FALSE`) or 1 (`TRUE`).

A `BOOL` type word extract bit, for example: `%MW10.4`.

C

CPU:

Is the abbreviation for Control Processing Unit.

This is the microprocessor. It is made up of the control unit combined with the arithmetic unit. The aim of the control unit is to extract the instruction to be executed and the data needed to execute this instruction from the central memory, to establish electrical connections in the arithmetic unit and logic, and to run the processing of this data in this unit. We can sometimes find ROM or RAM memories included in the same chip, or even I/O interfaces or buffers.

D

DDT:

DDT is the abbreviation for Derived Data Type.

A derived data type is a set of elements of the same type (`ARRAY`) or of various types (structure)

DFB:

DFB is the abbreviation for Derived Function Block.

DFB types are function blocks that can be programmed by the user ST, IL, LD or FBD.

By using DFB types in an application, it is possible to:

- simplify the design and input of the program,
- increase the legibility of the program,
- facilitate the debugging of the program,
- reduce the volume of the generated code.

DINT:

`DINT` is the abbreviation for Double Integer format (coded on 32 bits).

The lower and upper limits are as follows: -(2 to the power of 31) to (2 to the power of 31) - 1.

Example:

`-2147483648, 2147483647, 16#FFFFFFFF.`

E

EBOOL:

`EBOOL` is the abbreviation for Extended Boolean type. A `EBOOL` type variable brings a value (0 (`FALSE`) or 1 (`TRUE`) but also rising or falling edges and forcing capabilities.

An `EBOOL` type variable takes up one byte of memory.

The byte split up into:

- one bit for the value,
- one bit for the history bit (each time the state's object changes, the value is copied inside the history bit),
- one bit for the forcing bit (equals to 0 if the object isn't forced, equal to 1 if the bit is forced.

The default type value of each bit is 0 (`FALSE`).

EDT:

EDT is the abbreviation for Elementary Data Type.

These types are as follows:

- `BOOL`,
- `EBOOL`,
- `WORD`,
- `DWORD`,
- `INT`,
- `DINT`,
- `UINT`,
- `UDINT`,
- `REAL`,
- `DATE`,
- `TOD`,
- `DT`.

EFB:

Is the abbreviation for Elementary Function Block.

This is a block which is used in a program, and which performs a predefined software function.

EFBs have internal statuses and parameters. Even where the inputs are identical, the output values may be different. For example, a counter has an output which indicates that the preselection value has been reached. This output is set to 1 when the active value is equal to the preselection value.

F

FTP:

FTP is the abbreviation for File Transfer Protocol. FTP is the web's file transfer protocol.

H

HTTP:

HTTP is the abbreviation for HyperText Transfer Protocol. HTTP is the protocol for the formatting and transmission of files on the web. HTTP runs on top of TCP/IP (internet) protocols.

I

I/O object:

An I/O object is an implicit or explicit language object for an expert function module or an I/O device on a fieldbus. They are of the following types: %Ch, %I, %IW, %ID, %IF, %Q, % QW, % QD, QF, %KW, %KD, %KF, %MW, %MD, and %MF.

The object's topological address depends on the module's position on the rack or the device's position on the bus.

For Premium/Atrium controllers, double-type instances of located data (`%MD<i>`, `%KD<i>`) or floating (`%MF<i>`, `%KF<i>`) are located by an integer type (`%MW<i>`, `%KW<i>`). Only I/O objects make it possible to locate type instances (`%MD<i>`, `%KD<i>`, `%QD`, `%ID`, `%MF<i>`, `%KF<i>`, `%QF`, `%IF`) by using their topological address (for example `%MD0.6.0.11`, `%MF0.6.0.31`).

For Modicon M340 controllers, double-type instances of located data (`%MD<i>`, `%KD<i>`) or floating (`%MF<i>`, `%KF<i>`) are not available.

INT:

`INT` is the abbreviation for single integer format (coded on 16 bits).

The lower and upper limits are as follows: -(2 to the power of 31) to (2 to the power of 31) - 1.

Example:

`-32768, 32767, 2#1111110001001001, 16#9FA4.`

IODDT:

IODDT is the abbreviation for Input/Output Derived Data Type.

The term IODDT designates a structured data type representing a module or a channel of a controller module. Each application expert module possesses its own IODDTs.

M

multi task:

Application comprising several tasks (MAST, FAST, Auxiliary, event processing).

The order of priority for the execution of tasks is defined by the operating system of the controller.

multiple token:

Operating mode of an SFC. In multitoken mode, the SFC may possess several active steps at the same time.

N

naming convention (identifier):

An identifier is a sequence of letters, numbers and underlines beginning with a letter or underline (for example, the name of a function block type, an instance, a variable, or a section). If you select the **Extended** option in the **Tools > Project Settings... > Variables** dialog, letters from national character sets (for example, ö, ü, é, õ) can be used. Underscores are significant in identifiers. That is, `A_BCD` and `AB_CD` are interpreted as different identifiers. An underscore at the end of an identifier is not valid.

Identifiers cannot contain spaces. They are not case-sensitive.

According to IEC 61131-3 leading digits are not allowed in identifiers. Nevertheless, you can use them if you select (check) the **Allow leading digits** box in the **Tools > Project Settings... > Variables** dialog box.

According to IEC 61131-3 multiple leading underlines and consecutives underlines are not allowed in identifiers. Nevertheless, you can use them if you select the **Extended** option in the **Tools > Project Settings... > Variables > Character set** dialog.

Identifiers cannot be keywords.

P

procedure:

Procedures are functions view technically. The only difference to elementary functions is that procedures can take up more than one output and they support data type `VAR_IN_OUT`. To the eye, procedures are no different than elementary functions.

Procedures are a supplement to IEC 61131-3.

Program Unit:

A Program Unit is a part of program with it's own set of local and public variables. Program Units allow easy duplication and clear organization of program with local and public variables. Program Units are compliant with Program Organization Units (POUs) program as defined in IEC1131-3 standard.

S

section:

Program module belonging to a task which can be written in the language chosen by the programmer (FBD, LD, ST, IL, or SFC).

A task can be composed of several sections, the order of execution of the sections corresponding to the order in which they are created, and being modifiable.

single token:

Operating mode of an SFC chart for which only a single step can be active at any one time.

subroutine:

Program module belonging to a task (MAST, FAST, AUX) which can be written in the language chosen by the programmer (FBD, LD, ST, or IL).

A subroutine may only be called by a section or by another subroutine belonging to the task in which it is declared.

T

TIME:

The type `TIME` expresses a duration in milliseconds. Coded in 32 bits, this type makes it possible to obtain periods from 0 to (2 to the power of 32)-1 milliseconds.

U

UDINT:

`UDINT` is the abbreviation for Unsigned Double Integer format (coded on 32 bits) unsigned. The lower and upper limits are as follows: 0 to (2 to the power of 32) - 1.

Example:

`0, 4294967295, 2#11111111111111111111111111111111, 8#37777777777, 16#FFFFFFFF.`

UINT:

`UINT` is the abbreviation for Unsigned integer format (coded on 16 bits). The lower and upper limits are as follows: 0 to (2 to the power of 16) - 1.

Example:

`0, 65535, 2#1111111111111111, 8#177777, 16#FFFF.`

Index

A

adding
user directory 176
adjusting
variables 1466
animation
animation tables 1473
inspect window 1471
PLC screen 1453
program 1397
variables window 1468
animation tables
hyperlinks 1804
application 197
protecting 179
assistant
calling FFB functions 1199
import/export 1727
type library manager 327

B

backup project
Modicon M340 122
Premium 125
bookmarks
graphical languages 615
browser
DTM 283
Browser Contextual Menus 293
Field Bus Discovery 301
browser directories 203
budget
IO 1341

C

coils
LL984 editor 1120
comparing
project 104
configuration editor 1288
configuring
project 1288
configuring networks 448
configuring X-way 456
connecting to controller 91
setting address 93
connecting to controllers
communication parameters 102
types of connection 98
contacts
LL984 editor 1120, 1139
Container
FDT 267
conversion wizard
access 1744
conversion 1747
CPU configuration tab 1770
general procedure 1742
general representation 1746
introduction 1742
I/O remapping tab 1752
launch 1744
results 1747
structure tab 1748
creating
derived data types 356
EDT/DDT/IODDT/Device DDT instances 390
EFB/DFB instances 380
creating a program
LL984 editor 1120
current library 320

D

data editor 348
hyperlinks 1790
management 425
data storage 197
protecting 195
DDT (derived data type) 356
instances 390
debugging
program 1390
derived data types (DDT) 356
instances 390
derived function blocks (DFB) 1254

DFB
hyperlinks ... 1796
DFB (derived function block) ... 1254
DFB Type ... 1254
diagnostics ... 1515
diagnostics viewer ... 1522
documentation ... 1659
hyperlinks ... 1814
DTM
add ... 275
Browser ... 283
DTM hardware catalog ... 271
DTM Hardware Catalog ... 270
DTM properties ... 309
DTM services ... 295

E

EDS file
add ... 276
remove ... 278
EDT (elementary data type) ... 390
EFB (elementary function block)
instances ... 380
elementary data types (EDT) ... 390
elementary function blocks (EFB)
instances ... 380
encryption
file ... 179
equation network blocks
LL984 editor ... 1120, 1163
Excel
export ... 1708
import ... 1711
export ... 1686
Excel ... 1708
exporting
exporting with file encryption active ... 1687

F

FBD
reference data type ... 739
FBD editor ... 651
creating a program ... 651
editing FFBs ... 669
other functions ... 615
search/replace function ... 1239
selecting data ... 1181
using type library browser ... 1231
FDT container
introduction ... 267
user rights ... 268
FDT Container ... 267
FFB library version
properties ... 177
Field Bus Discovery ... 301
file
encryption ... 179
file encryption
partial program export ... 1687
filtering
data ... 433
firmware ... 197
protecting ... 190
forget
password ... 197
FTP ... 155
function blocks
LL984 editor ... 1120, 1154
functional modules ... 237

G

go to
project browser ... 173
graphical languages
bookmarks ... 615

H

Hardware Catalog
DTM ... 270
hyperlinks ... 1774

I

IL
reference data type ... 1079
IL editor ... 1051
other functions ... 1016
search/replace function ... 1239
selecting data ... 1181

using type library browser 1231
import 1686
Excel 1711
import XEF file
Libset version 1723
import ZEF file
Libset version 1723
information
Libset 316, 320
INIT 122
instruction editor
LL984 editor 1501
I/O objects management 1207
IO
budget 1341
IODDT 390

K

keyboard
shortcut 1848

L

LD
reference data type 882
LD editor 745
creating a program 745
other functions 615
search/replace function 1239
using type library browser 1231
libraries
Libset 314
library content 320
library management
variables and FFB types 314
library manager 318
areas 319
Libset
installation tool 314
Libset (set of libraries) 314
Libset information 316, 320
Libset version
import XEF file 1723
import ZEF file 1723
modifying 315
options 76
properties 177
upgrade/downgrade 316
links
LL984 editor 1120, 1158
LL984 editor
assigning variables 1120, 1170
coils 1120, 1146
contacts 1120, 1139
creating a program 1120
equation network blocks 1120, 1163
function blocks 1120, 1154
instruction editor 1501
links 1120, 1158
network display 1120, 1129
search/replace function 1239
subroutines 1120, 1173
loss
password 197

M

maintain output links on disabled EF
project settings 581
memory
pack function 151
memory card 155
memory card access
Modicon M340 153
memory consumption 139
modifying the Libset version 315

N

network display
LL984 editor 1120, 1129

O

offline mode 74
online mode 83
online modifications 85
open project dialog 76
operator screen 1554
hyperlinks 1808
options 549
Libset version 76

output windows 1841

P

password
forget 197
loss 197
section 186
Program Unit 475
creation 475
protecting 192
project browser 161
hyperlinks 1786
project settings
maintain output links on disabled EF 581
properties
FFB library version 177
Libset version 177
protecting
application 179
data storage 195
firmware 190
Program Unit 192
section 192
purge of data 439

R

Ready 289
RUN 122

S

safety-related area
password 186
search/replace function 1239
section
protecting 192
sections 502
security editor 71
selecting
data 1181
settings
Control Expert 595
project 549
toolbar/menu 610
SFC editor 890
creating a program 890
other functions 615
search/replace function 1239
selecting data 1181
using type library browser 1231
ST
reference data type 1118
ST editor 1082
other functions 1016
search/replace function 1239
selecting data 1181
using type library browser 1231
status bar 1843
STOP 122
structural view 203
structure (DDT) 356
instances 390
subroutines
LL984 editor 1120, 1173
summary
configuration 270
connections 270

T

tasks 472
time out
FTP 155
transferring
data 127
projects 117
type library
update 336
type library browser 1231
type library manager 314, 318
areas 319
description 321

U

updating
initial values 139
upgrade/downgrade the Libset version 316
upload information 120

V

variables (EDT) 390
variables assignment
  LL984 editor 1120, 1170

Z

zoom 621, 1023
  project browser 166

Schneider Electric
35 rue Joseph Monier
92500 Rueil Malmaison
France

+ 33 (0) 1 41 29 70 00

www.se.com

As standards, specifications, and design change from time to time, please ask for confirmation of the information given in this publication.

© 2023 Schneider Electric. All rights reserved.

33003101.27