

System 800xA Engineering Engineering Studio Function Designer

System Version 5.1

Power and productivity
for a better world™



System 800xA Engineering

Engineering Studio Function Designer

System Version 5.1

NOTICE

This document contains information about one or more ABB products and may include a description of or a reference to one or more standards that may be generally relevant to the ABB products. The presence of any such description of a standard or reference to a standard is not a representation that all of the ABB products referenced in this document support all of the features of the described or referenced standard. In order to determine the specific features supported by a particular ABB product, the reader should consult the product specifications for the particular ABB product.

ABB may have one or more patents or pending patent applications protecting the intellectual property in the ABB products described in this document.

The information in this document is subject to change without notice and should not be construed as a commitment by ABB. ABB assumes no responsibility for any errors that may appear in this document.

In no event shall ABB be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without written permission from ABB, and the contents thereof must not be imparted to a third party nor used for any unauthorized purpose.

The software or hardware described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license. This product meets the requirements specified in EMC Directive 2004/108/EC and in Low Voltage Directive 2006/95/EC.

TRADEMARKS

All rights to copyrights, registered trademarks, and trademarks reside with their respective owners.

Copyright © 2004-2015 by ABB.
All rights reserved.

Release: July 2015
Document number: 3BDS011224-510 G

Table of Contents

About this User Manual

General	13
User Manual Conventions	14
Feature Pack	14
Warning, Caution, Information, and Tip Icons	14
Terminology	15
Released User Manuals and Release Notes	15
.....	16

Section 1 - Introduction

Product Scope	20
Prerequisites and Requirements	21
Product Verification	22
Getting Started	23
Reporting Problems	23

Section 2 - Configuration

Function Components and Diagrams	28
Preference Settings	29
Object Type Definition and Usage	34
Aspect Properties/Parameters	36
Defining Function Components (Component View)	50
Function Components with XML Type Description (Component View)	52
Customizing a Function Component Type	59
Aspect Objects and Symbol Objects	67
Object Type Definition Create Info	69

Inherited Function Components.....	72
Overridden Components	75
Default Aspect	78
Instantiating Function Components in Diagram View.....	80
Change Type	80
Connections/Networks.....	84
Vertical Navigation	95
Horizontal Navigation.....	95
System Functions	108
AC 800M Integration	112
Introduction and Requirements.....	112
Workflow Overview.....	113
Building up Functional Structure and Function Diagrams	114
Defining Data Flow Order	116
Explicitly and Implicitly Defined Variables	118
Conversion of Global Variables (Diagram Variables) into Communication Variables	
121	
Building up Control Structure.....	124
Allocating Function Diagrams.....	126
Allocating I/O Signals	127
Generate Configuration Data	132
Variable Creation	133
Connection and Inversion Checks.....	137
Splitter and Joiner	141
Connection Mapping.....	143
Instance Specific Initial Values.....	144
Consistency	146
Enhanced Naming.....	153
Paste Rename Functionality.....	172
Communication Between Applications using Cross Communication	178
Cross Communication Procedure	180
On-line Display.....	184

Watch Window	192
Force Dialog	199
Field Device Integration	200
Introduction	200
HART Devices	201
PROFIBUS Devices	206
FOUNDATION Fieldbus	211
Error Messages	220
PROFINET Devices	221
IEC61850 Devices	226
Bulk Data Manager Support and Bulk Operations	232
Import / Export (AFW-Files)	263
Export / Import Function Diagram	263
Export / Import Application	265
Export / Import IO Allocation	267
Export / Import Project	268
Segmentation of AFW-Files	269
Engineering Repository	270

Section 3 - Graphic Editor Reference

Exploring the Windows User Interface	273
Main Window Handling	274
Diagram/Component View	274
Diagram/Component Document Windows	276
Dockable Windows/Toolbars	278
Changing the Location of Docked Windows/Toolbars	279
Working with Menus	281
File Menu	282
Edit Menu	293
View Menu	299
Insert Menu	308
Layout Menu	311
Allocation Menu	315

Online Menu	317
Window Menu.....	323
Help Menu	324
Context Menus	325
Working with Toolbars.....	343
Customize	344
Toolbar Categories	349
Showing/Hiding Toolbars	364
Creating a User-Defined Toolbar	366
Adding Tool Buttons to Toolbars.....	366
Removing Tool Buttons from a Toolbar	368
Resetting the Composition of a Toolbar	368
Renaming a User-Defined Toolbar	368
Deleting a User-Defined Toolbar	369
Changing the Look of Toolbars	369
Showing Tooltips	369
General Accelerator Keys	370
Open Aspect.....	373
Inserting/Creating Components	373
Inserting Graphic Primitives	377
Inserting Text Components	378
Inserting Label Components	378
Inserting Ports	380
Inserting Pictures/Images.....	380
Inserting ActiveX Controls	381
Inserting Graphic Elements build with Graphics Builder.....	383
Inserting Designer Components.....	384
Viewing Components	389
Zoom	390
Pan	392
Selecting Components.....	393
Selecting Symbols.....	396

Selecting Ports.....	400
Selecting Connections.....	404
Layout of Components	407
Grid	407
Move	409
Size	411
Align	411
Rotate	413
Flip	414
Order	414
Group	417
Cut/Copy/Paste/Delete Components	417
Cut and Paste.....	418
Copy and Paste.....	419
Paste Options.....	423
Delete	427
Defining Component Properties	429
Ambient Properties.....	429
Component Properties	437
Default Properties.....	465
Layers	470
Layer Properties	470
Pages	473
Measurements and Size.....	474
Page Setup.....	479
Select Master Page Template	479
Templates and Skeletons	481
Insert/Delete Page	482
Page Comments.....	483
Page Navigation	484
Printing	486
Contents	487

Print	488
Print Preview	492
Options	493
Print	495
Paste	496
Autorouting	497
Naming	497
Import/Export	499
File Export	500
File Import	501
Export Diagram Data / Import Diagram Data	502
Undo/Redo	503
Undo	504
Redo	504

Section 4 - Error Messages

Appendix A - Restored Functionalities

Single Control Modules	507
Creation of an SCM based Function Diagram	507
Implicit Casts	509
Connection/Networks	512
Connection Mapping	514
MMS Cross Communication	537
MMS Communication Overview	538
Export / Import of MMS Generated Function Diagrams from Engineering System to Production System	575
Multiple Allocatable Groups and Multiple Code Blocks	576
Mapping Aspect Objects	576
Grouping Function Components into Allocatable Groups	577
Vertical Navigation	594
Variable Creation	598
Explicitly and Implicitly Defined Variables	598

Labelling Split Connections	599
Building up Control Structure	601
Allocating Allocatable Groups	602
Referencing Devices	608
Synchronization between Control Builder Name aspect and Name aspect	609

Index

Revision History

Updates in Revision Index A	628
Updates in Revision Index B	629
Updates in Revision Index C	629
Updates in Revision Index D	629
Updates in Revision Index E	630
Updates in Revision Index F	630
Updates in Revision Index G	631

About this User Manual

General



Any security measures described in this User Manual, for example, for user access, password security, network security, firewalls, virus protection, etc., represent possible steps that a user of an 800xA System may want to consider based on a risk assessment for a particular application and installation. This risk assessment, as well as the proper implementation, configuration, installation, operation, administration, and maintenance of all relevant security related equipment, software, and procedures, are the responsibility of the user of the 800xA System.

This User Manual describes Function Designer, the configuration tool for Function Diagrams. It serves as an application and reference manual for Function Designer provided with Engineering Workplace of System 800xA.

The User Manual also describes the use of Function Designer as a diagnostic and commissioning tool.

[Section 1, Introduction](#) describes briefly about Function Designer.

[Section 2, Configuration](#) describes configuration and application procedures with Function Designer.

[Section 3, Graphic Editor Reference](#) describes the graphic editor of Function Designer in detail.

[Appendix A, Restored Functionalities](#) provides information for specific Engineering Studio functionalities or features available in Function Diagrams restored to Engineering Studio 800xA 5.1.

Basic Function Designer functionalities and workflows are explained in *System 800xA Engineering, Engineering Studio Function Designer Getting Started (3BDS100968*)*.

For details on Bulk Data Manager and IO allocation, refer to *System 800xA Engineering, Engineering Studio (3BDS011223*)*.

User Manual Conventions

Microsoft Windows conventions are normally used for the standard presentation of material when entering text, key sequences, prompts, messages, menu items, screen elements, etc.

Feature Pack

The Feature Pack content (including text, tables, and figures) included in this User Manual is distinguished from the existing content using the following two separators:

Feature Pack Functionality

<Feature Pack Content>

Feature Pack functionality included in an existing table is indicated using a table footnote (*):

*Feature Pack Functionality

Feature Pack functionality in an existing figure is indicated using callouts.

Unless noted, all other information in this User Manual applies to 800xA Systems with or without a Feature Pack installed.

Warning, Caution, Information, and Tip Icons

This User Manual includes Warning, Caution, and Information where appropriate to point out safety related or other important information. It also includes Tip to point

out useful hints to the reader. The corresponding symbols should be interpreted as follows:



Electrical warning icon indicates the presence of a hazard that could result in *electrical shock*.



Warning icon indicates the presence of a hazard that could result in *personal injury*.



Caution icon indicates important information or warning related to the concept discussed in the text. It might indicate the presence of a hazard that could result in *corruption of software or damage to equipment/property*.



Information icon alerts the reader to pertinent facts and conditions.



Tip icon indicates advice on, for example, how to design your project or how to use a certain function

Although Warning hazards are related to personal injury, and Caution hazards are associated with equipment or property damage, it should be understood that operation of damaged equipment could, under certain operational conditions, result in degraded process performance leading to personal injury or death. Therefore, fully comply with all Warning and Caution notices.

Terminology

A complete and comprehensive list of terms is included in *System 800xA System Guide Functional Description (3BSE038018*)*. The listing includes terms and definitions that apply to the 800xA System where the usage is different from commonly accepted industry standard definitions and definitions given in standard dictionaries such as Webster's Dictionary of Computer Terms.

Released User Manuals and Release Notes

A complete list of all User Manuals and Release Notes applicable to System 800xA is provided in *System 800xA Released User Manuals and Release Notes (3BUA000263*)*.

System 800xA Released User Manuals and Release Notes (3BUA000263)* is updated each time a document is updated or a new document is released. It is in pdf format and is provided in the following ways:

- Included on the documentation media provided with the system and published to ABB SolutionsBank when released as part of a major or minor release, Service Pack, Feature Pack, or System Revision.
- Published to ABB SolutionsBank when a User Manual or Release Note is updated in between any of the release cycles listed in the first bullet.



A product bulletin is published each time *System 800xA Released User Manuals and Release Notes (3BUA000263*)* is updated and published to ABB SolutionsBank.

Section 1 Introduction

Function Designer is the engineering portal for plant oriented functional planning with Aspect Objects. It provides a graphical user interface to easily engineer, document, and maintain the complex AC 800M control strategies.

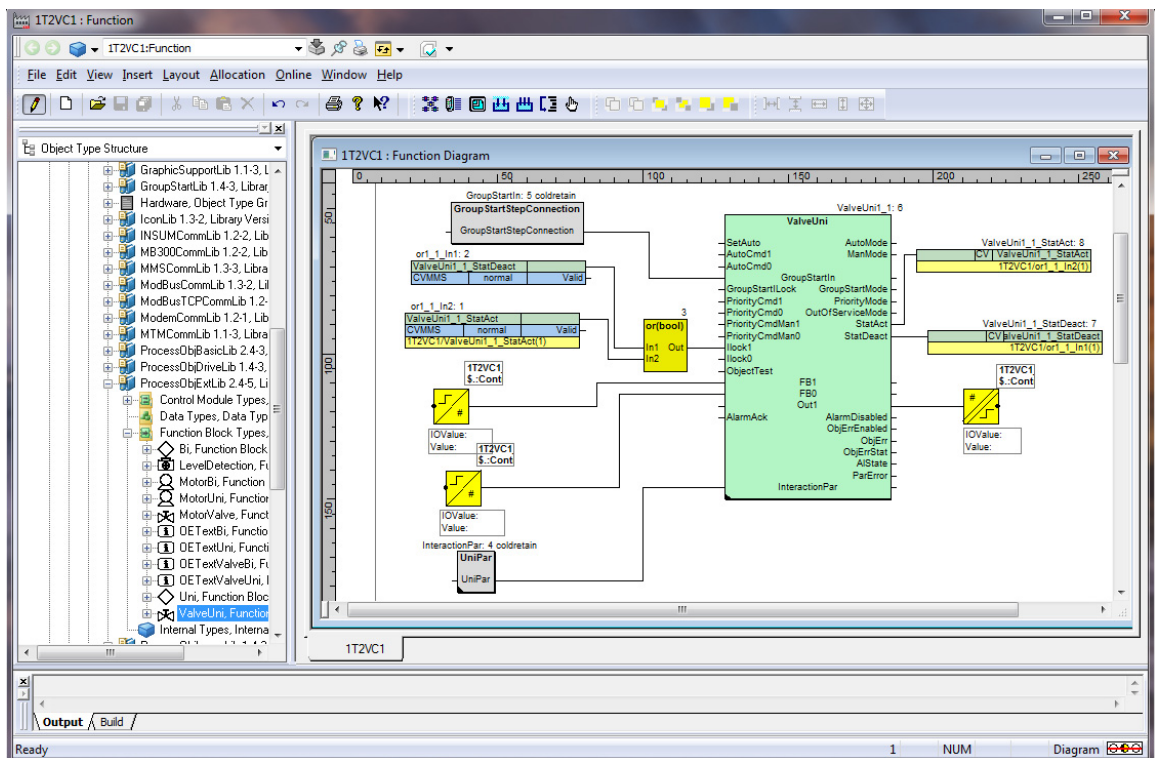


Figure 1. Example Function Diagram

Function Designer is used to create Function Diagrams as shown in [Figure 1](#), which are interlinked by Function aspects of the objects in the Functional Structure. These objects represent Process Functions or Process Sequences of the plant.

Function Diagram can be accessed in **Diagram view** from the Function aspect. User can insert and graphically connect block symbols on a Function Diagram. The block symbols represent Function Component Aspect Object types.



The standard library object types of AC 800M Connect and HART Device Integration are provided as ready-made Function Component Aspect Object types.

A Function Diagram can have a complete process control loop with all its Function Blocks / Control Modules, Field Devices, CBM_Signal objects (engineering signals) and their signal connections, see [Figure 1](#). It can be enriched by textual comments, schematic drawings, and ActiveX components.

Sequences can be included into Function Diagrams and their steps, transitions, and actions can be configured in additional Function Diagrams.

Configuration data can be generated for a Function Diagram after allocating it to an application in Control Builder M.

From 800xA 5.1 onwards, the newly created Function Diagrams are available within a **Diagram** folder in the Control Builder and Control Structure. Function Diagrams restored to 800xA 5.1 are available as Single Control Modules (SCM), as shown in [Figure 2](#).

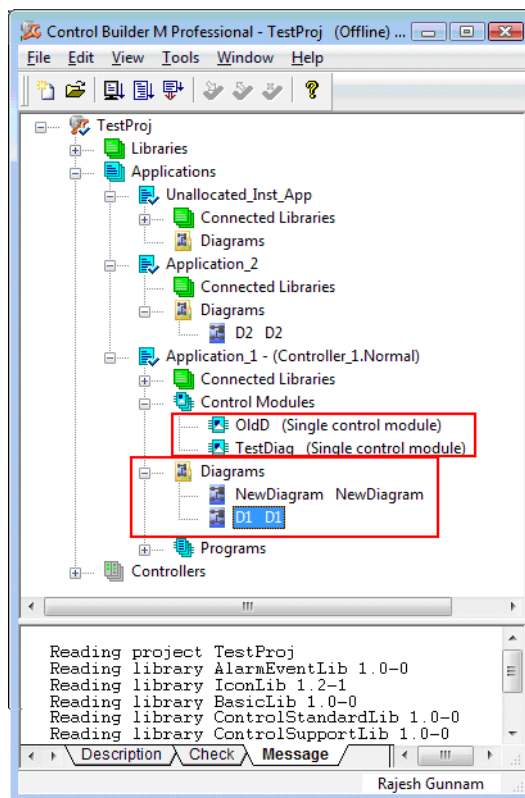


Figure 2. Function Diagrams Available as Diagram and Single Control Module

Diagrams can be connected between applications or across controllers using communication variables. User can also manually assign the I.P address for controllers to enable communication across aspect directories.

The diagrams can be downloaded using Control Builder Professional. They can be tested in Function Designer using Online Display and Watch Window functions and Control Builder Professional online functions.

Function Diagrams can be printed in bulk for plant documentation on paper or electronic media.

Copying, allocating, generating configuration data of Function Diagrams can be done in bulk using Bulk Data Manager.

User can design graphical (block) symbols representing Function Component Aspect Object types from the **Component view** of the Function aspect and perform the engineering from the Object Type Structure.

Product Scope

Function Designer is a component of Engineering Workplace of System 800xA.

By default Function Designer supports the functional planning approach for the engineering of AC 800M applications with local and remote S800/S900 I/O, HART devices, and PROFIBUS devices using the standard libraries of AC 800M Connect, HART Device Integration, and PROFIBUS Device Integration.

Scope of Function Designer is extendable due to the ability to configure user defined Function Components.

If environment support is enabled in the Configuration Wizard, Function Designer can be used in Engineering environment and in Production environment.

Function Designer can be used to configure applications in **Configure and Deploy** method. Applications configured with Function Designer can be downloaded through **Load Evaluate Go** method.

Configure and Deploy

Configure and Deploy allows modifications to be performed in the Engineering environment without affecting the running process until the modifications are deployed to the Production environment for immediate execution.



Separate license is required to enable the **Configure and Deploy** functionality.

For more information on **Configure and Deploy**, refer to *System 800xA Engineering, Engineering and Production Environments (3BSE045030*)*.



Function Designer is not certified for building SIL-certified applications. It can be used to configure non-SIL-certified applications for the AC 800M High Integrity controller. SIL stands for Safety Integrity Level, as specified in the standard IEC-61508.

The certified tool to configure SIL certified applications is Control Builder M Professional. To run SIL certified AC 800M applications, the SIL certified AC 800M High Integrity controller is required.

For information on guidelines and safety considerations related to all safety life-cycle phases of an AC 800M High Integrity controller, refer to *System 800xA Safety, AC 800M High Integrity Safety Manual (3BNP004865*)*. This manual also lists the available SIL certified library object types.

For more information on SIL-application related configuration procedures to perform with Control Builder M Professional, refer to *System 800xA Control, AC 800M Configuration (3BSE035980*)*.

Prerequisites and Requirements

Prerequisites, requirements, and the installation procedures are explained in the *System 800xA, Installation (3BSE034678*)* manual. Post installation procedures (including system creation) are explained in the *System 800xA, Post Installation (3BUA000156*)* manual.



The following system extensions related to Function Designer have to be loaded:

- AC800M Connect.
- ABB CI Extension for AC800M Connect.
- ABB Engineering Base.
- ABB DM & PM Application.
- ABB Signal Extension for AC800M Connect.
- ABB Function Designer for Fieldbus Builder PROFIBUS/HART.
- PROFIBUS & HART Device Integration Library - Basics.
- ABB Function Designer.
- ABB Function Designer for AC800M Connect.

Product Verification

User can verify the product version in Microsoft Windows 7 through **Control Panel > Programs > Programs and Features**. If the **Version** tab is not visible, then follow the steps to view the **Version** tab:

1. Right-click any available tab (such as **Name**, **Publisher**, etc).
2. Select **More...**, see [Figure 3](#).

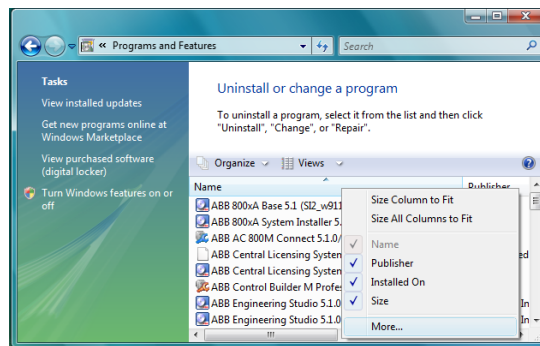


Figure 3. Context Menu to Access More Information

3. Select the **Version** check box.
4. Click **OK**.

The version details are displayed below the **Version** tab.

Follow the above procedure to view other tabs such as **Support Telephone**, **Help Link**, etc.

User can also verify the product version from the Plant Explorer through **About Industrial^{IT}**.

Further, the specific product's version can be verified through its **Help > About** menu command. For example, Function Designer's version details can be verified through its **Help > About** menu command from the Diagram or Component view. The version details are displayed as shown in [Figure 4](#).

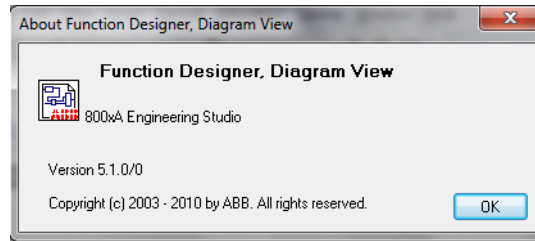


Figure 4. About Function Designer, Diagram View Window

Getting Started

For information on getting started and tutorial instructions, refer to *System 800xA Engineering, Engineering Studio Function Designer Getting Started (3BDS100968*)*.

Reporting Problems

User can report problems to the local ABB supplier. Ensure to report the problems with the relevant information, such as the product version designation and build number, the description of the problem scenario, and the detailed description of the error messages that appear.

Section 2 Configuration

The following definitions are essential while working with Function Designer, and during its integration with Control Builder M Professional/AC 800M, and with Fieldbus Builder PROFIBUS/HART:

- [Function Diagram](#)
- [Symbols](#)
- [Function Component](#)
- [Function Component Type](#)
- [Communication Variable](#)
- [Allocatable Group](#)
- [Mapping Aspect Objects](#)

Function Diagram

A Function Diagram includes one or more function components and symbols. A diagram is displayed in the preview area or in a new window.

Symbols

The Function Diagram contains symbols in its default logic layer, which are rectangular blocks with icons, labels, and $0 \dots n$ input, and $0 \dots n$ output ports.

Function Component

Function components are symbols which have function-related aspect data to other aspect systems.

For example, to:

- Controller configuration aspect system Control Builder Professional/AC 800M Connect:
A function component may represent one of the following instances:
 - System function.
 - System function block.
 - System control module.
 - Sequence (Either IEC 61131-3 Sequence or Sequence2D (based on library SeqStartLib).
 - User-defined function block.
 - User-defined control module.
 - I/O signal.
- PROFIBUS-, HART-configuration aspect systems:
A function component may represent a field device.
- HSI aspect system:
A function component may represent an HSI function.
- Function Designer aspect system:
A function component may represent:
 - A child Function Diagram on lower level. For more information, refer to [Vertical Navigation](#) on page 95.
 - An off-diagram reference, which is a connection or cross-link to other Function Diagrams. For more information, refer to [Horizontal Navigation](#) on page 95.
 - An on-page connector, which is a label representing a sink/source of connection on the same page to avoid too many crossings.
 - An off-page connector, which is a page cross-reference with label representing a sink/source of connection between different pages. For more information, refer to [Off Page Connector](#).
 - An explicitly declared local variable. For more information, refer to [Explicitly and Implicitly Defined Variables](#).

- A splitter or joiner block. For more information, refer to [Splitter and Joiner](#) on page 141 [Splitter and Joiner](#).

Function Component Type

Function components are instances of function component types defined in Object Type Structure. The object type definition also defines if a function component is instantiated as Aspect Object, or as symbol object in the Function Diagram only (see also [Object Type Definition Create Info](#)).

According to this, function components can be:

- Aspect Objects both visible in Functional Structure and in a Function Diagram.
- Symbol objects only visible in the Function Diagram.

[Figure 5](#) shows an example.:

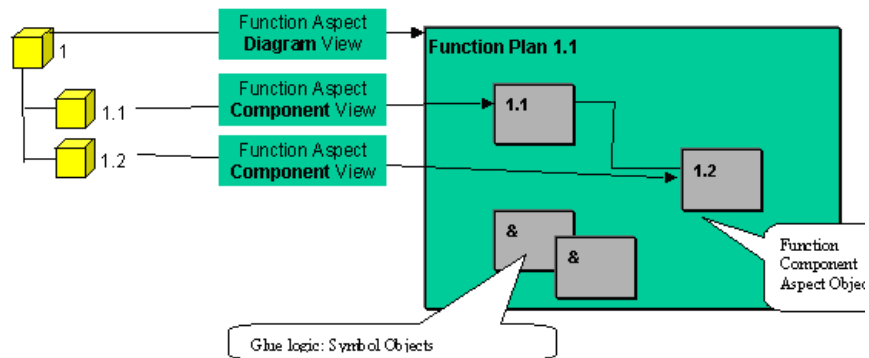


Figure 5. Function Diagram with Function Component Aspect Objects and Symbol Objects

Communication Variable

Communication Variables are used to connect various diagrams which are placed in the same or across applications / controllers.



Communication variables can not be used for SIL based applications.

Allocatable Group

Every bundle of function components, which will be assigned later on in the Control Structure, is regarded as an allocatable group. An allocatable group describes a group of components which are or will be allocated to the same Aspect Object. By default, each object having a function diagram also has an allocatable group aspect.

- **Assigning Complete Function Diagram Logic to One Allocatable Group:**

By default, an allocatable group aspect is automatically generated for each diagram object (aspect object with function aspect and non-empty Diagram view). This allows the user to allocate the whole function diagram to the Control Structure later on. Then, all logical components belonging to that function diagram can be allocated to one application / task on a controller.

Mapping Aspect Objects

Each allocatable group can belong to only one Function Diagram. It is not possible to assign logic of different Function Diagrams into the same allocatable group.

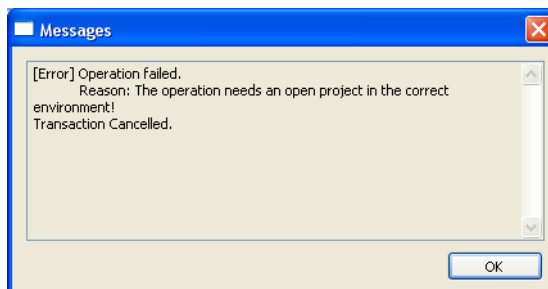
For AC 800M controllers, each allocatable group creates a Diagram or an I/O signal.

Function Components and Diagrams



If the user uses *Control Builder M* (CBM) function blocks, control modules, and data types in *Function Designer*, CBM must be running and a control project with the corresponding library must be open.

Otherwise, if the user tries to insert such objects in a Function Diagram, user will receive an error message as shown below:



Preference Settings

Function Designer can save the recent settings and preferences as described below:

- **Function Settings**

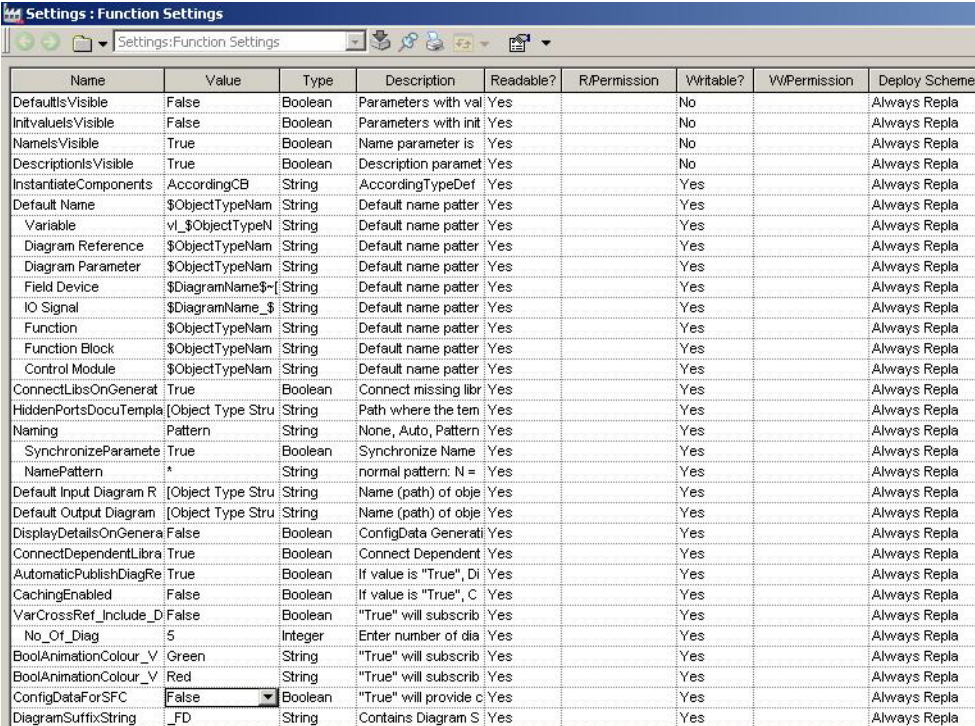
System wide settings like visibility of port parameters, instantiation kind, and rules are stored within the Function Settings aspect, see [Figure 6](#).

- The Function Settings **DefaultIsVisible**, **InitvalueIsVisible**, **NameIsVisible**, and **DescriptionIsVisible** hide corresponding port parameters on instantiation of Function Blocks and control modules. The default settings in [Figure 6](#) automatically hide port parameters of control modules with default or initial values (need not to be connected), ‘Name’ and ‘Description’ and show other parameters which need to be connected.
- The function setting **InstantiateComponents** sets the instantiation kind. According to that setting, new components in a function diagram are either created as function component aspect object, visible in Functional Structure, or as function component symbol object, visible on the diagram only.
- The function setting **DefaultName** determine the name pattern used on instantiation of variables, diagram references, diagram parameters, field devices, IO signals, functions, function blocks and control modules. \$ObjectTypeName_1 generates a default name that takes the object type name, and a unique running number 1,2...N. E.g. PidLoop1.



If symbols are instantiated in a Function Diagram with a naming convention different from the default naming pattern, then any further instantiations of the same component uses the new naming pattern as the default for that particular Function Diagram.

- By default, the function setting **CachingEnabled** is set to False. This setting enables diagram modifications to be saved to the aspect directory. If the function setting **CachingEnabled** is set to True, caching is enabled and diagram modifications are not saved to aspect directory until **File > Save** is performed.



Name	Value	Type	Description	Readable?	R/Permission	Writable?	W/Permission	Deploy Scheme
DefaultsVisible	False	Boolean	Parameters with val	Yes		No		Always Repla
InitvaluesVisible	False	Boolean	Parameters with init	Yes		No		Always Repla
NamesVisible	True	Boolean	Name parameter is	Yes		No		Always Repla
DescriptionsVisible	True	Boolean	Description paramet	Yes		No		Always Repla
InstantiateComponents	AccordingCB	String	AccordingTypeDef	Yes		Yes		Always Repla
Default Name	\$ObjectTypeNam	String	Default name patter	Yes		Yes		Always Repla
Variable	vl_\$ObjectTypeN	String	Default name patter	Yes		Yes		Always Repla
Diagram Reference	\$ObjectTypeNam	String	Default name patter	Yes		Yes		Always Repla
Diagram Parameter	\$ObjectTypeNam	String	Default name patter	Yes		Yes		Always Repla
Field Device	\$DiagramName\$-	String	Default name patter	Yes		Yes		Always Repla
IO Signal	\$DiagramName_\$	String	Default name patter	Yes		Yes		Always Repla
Function	\$ObjectTypeNam	String	Default name patter	Yes		Yes		Always Repla
Function Block	\$ObjectTypeNam	String	Default name patter	Yes		Yes		Always Repla
Control Module	\$ObjectTypeNam	String	Default name patter	Yes		Yes		Always Repla
ConnectLibsOnGenerat	True	Boolean	Connect missing libr	Yes		Yes		Always Repla
HiddenPortsDocuTempla	[Object Type Stru	String	Path where the tem	Yes		Yes		Always Repla
Naming	Pattern	String	None, Auto, Pattern	Yes		Yes		Always Repla
SynchronizeParamete	True	Boolean	Synchronize Name	Yes		Yes		Always Repla
NamePattern	*	String	normal pattern: N =	Yes		Yes		Always Repla
Default Input Diagram R	[Object Type Stru	String	Name (path) of obje	Yes		Yes		Always Repla
Default Output Diagram	[Object Type Stru	String	Name (path) of obje	Yes		Yes		Always Repla
DisplayDetailsOnGenera	False	Boolean	ConfigData Generati	Yes		Yes		Always Repla
ConnectDependentLibra	True	Boolean	Connect Dependent	Yes		Yes		Always Repla
AutomaticPublishDiagRe	True	Boolean	If value is "True", Di	Yes		Yes		Always Repla
CachingEnabled	False	Boolean	If value is "True", C	Yes		Yes		Always Repla
VarCrossRef_Include_D	False	Boolean	"True" will subscrib	Yes		Yes		Always Repla
No_Of_Diag	5	Integer	Enter number of dia	Yes		Yes		Always Repla
BoolAnimationColour_V	Green	String	"True" will subscrib	Yes		Yes		Always Repla
BoolAnimationColour_V	Red	String	"True" will subscrib	Yes		Yes		Always Repla
ConfigDataForSFC	False	Boolean	"True" will provide c	Yes		Yes		Always Repla
DiagramSuffixString	_FD	String	Contains Diagram S	Yes		Yes		Always Repla

Figure 6. Function Settings

In Function Settings aspect set **DisplayDetailsOnGenerateConfigData** property to 'true', in order to display a working dialog indicating the progress and a Cancel button. On cancel, the generation of the current diagram/group is finished before the build process is aborted. By default **DisplayDetailsOnGenerateConfigData** property is 'false'.

In Function Settings aspect set **ConnectDependentLibrariesOnGenerateConfigData** property to 'true', in order to avoid errors in Control Builder due to dependent libraries. By default **ConnectDependentLibrariesOnGenerateConfigData** property is 'false'.



Depending on the [Object Type Definition Create Info](#) and [Function Settings](#) an instantiated object will be placed only in the function diagram as symbol object or additionally in the Functional Structure as an Aspect Object.

Depending on the [Function Settings](#) default names for field devices, IO signals, function blocks and control modules are prefixed by the diagram name \$DiagramName - or not.

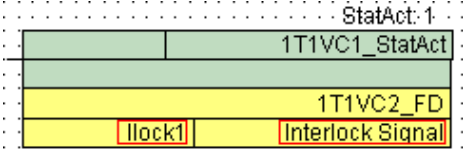
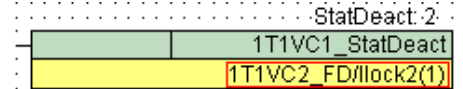
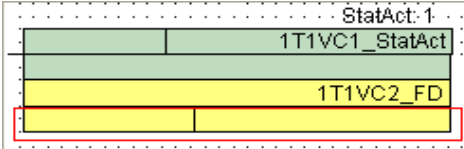
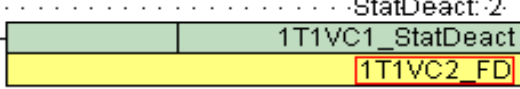
For example to change the given default settings (see [Figure 6](#)) you could set:

```
[Object Type Structure]Object Types/Functional Planning/Settings:  
Function Settings.InstantiateComponents = AccordingTypeDefCreateInfo  
Function Settings.DefaultName.FieldDevice =  
$DiagramName_$ObjectName  
Function Settings.DefaultName.IO Signal = $ObjectName  
Function Settings.DefaultName.Function Block = $ObjectName  
Function Settings.DefaultName.Control Module = $ObjectName.
```

Two new Function Setting properties are included for memory optimization. Set **VarCrossRef_Include_DiagRef** to **False** to use the memory optimally.

[Table 1](#) indicates the Function Designer behavior when the setting **VarCrossRef_Include_DiagRef** is set to True or False.

Table 1. Settings for VarCrossRef_Include_DiagRef

VarCrossRef_Include_DiagRef > True	VarCrossRef_Include_DiagRef > False
<p>If VarCrossRef_Include_DiagRef is set to True:</p> <p>1. The connected diagram reference details (name and description) appear in the yellow reference (refer screenshot below):</p>  <p>2. The connected reduced diagram reference detail (name) appears in the yellow reference (refer screenshot below):</p> 	<p>If VarCrossRef_Include_DiagRef is set to False:</p> <p>1. The connected diagram reference details (name and description) do not appear in the yellow reference (refer screenshot below):</p>  <p>2. The connected reduced diagram reference detail (name) does not appear in the yellow reference (refer screenshot below):</p> 

The value of the sub property **No_Of_Diag** indicates the number of connected diagrams loaded simultaneously per transaction for updating the modified properties of the diagram variable. It is recommended to use the default value (5).



No_Of_Diag sub property is applicable only if **VarCrossRef_Include_DiagRef** is set to **False**.

Two new Bool Animation color settings are added in the **Function Settings** to indicate various connection line colors based on the value transfer between the

ports. Color settings are explained in [Table 2](#):

Table 2. Bool Animation Color

Name of the Function Setting	Color
BoolAnimation_Value_1 (True)	Green (or) Red
BoolAnimation_Value_0 (False)	Red (or) Blue

Config Data For SFC

Default: False

If Config Data For SFC is set to True, then **SFCXRefDataMapper** is created while generating config data.

This setting is used by SFC viewer feature pack version to navigate from action window.

Diagram Suffix String

Default:Diagram Suffix String is _FD

Naming convention of the Diagram name is _(underscore) followed by any character.

This setting is used by SFC viewer feature pack version to navigate from action window.



The SFC Viewer Feature Pack release version needs to be installed to configure Config Data For SFC and Diagram Suffix String.

- **Workspace in Function aspect**

Recent function component / diagram settings like zoom factor, grid properties, default component properties are stored in a workspace within the Function aspect. These settings are individually maintained per function component / diagram.

- **Windows registry**

User interface preferences including maximized popup window, customized toolbar settings and dockable windows are stored per user and per view in a

workspace in the Windows registry. The first time *Function Designer* is run, default toolbars and dockable windows are visible. The windows and toolbars can be rearranged as desired. Next time *Function Designer* is run, the same view, and the window/toolbar workspace gets restored.

There are four separate workspaces:

- Plant Explorer preview window, Component view
- Plant Explorer preview window, Diagram view
- Plant Explorer popup window, Component view
- Plant Explorer popup window, Diagram view

If a previous version is installed, few newly added toolbar buttons may not be initially visible. To return to the default toolbars, and to ensure all toolbar buttons visible, use the menu command **View > Toolbar > Customize**. Select each toolbar and click **Reset**. Do this for the Component view and the Diagram view, both in preview and in popup window of Plant Explorer.



If some toolbar buttons are still not visible, open the Microsoft Windows 7 / Server 2008 registry (run regedit) and delete the following keys:

HKEY_CURRENT_USER\Software\ABB Automation Products
GmbH\EngineerIT\FuDiagramDesigner

HKEY_CURRENT_USER\Software\ABB Automation Products
GmbH\EngineerIT\FuComponentDesigner

Object Type Definition and Usage

To have Function aspects on the Object Types, for example to provide function component object types, there are three ways to use the Function aspects.

On creation of an Aspect Object from the Object Type which includes a Function aspect, it may be:

- used as a *template* - the Function aspect is not copied. If a Function aspect is created later, the one which was created on the Object Type will be copied.
- used as a *copy* - the Function aspect is copied.
- *inherited* - the Function aspect will not be copied but inherited.

For more information on the concepts of object types, refer to *System 800xA, System Planning (3BSE041389*)*. For more details on how to create object types, refer to *System 800xA, Configuration (3BDS011222*)*.

Function component instances typically comprise of both inherited and copied data. The symbol and interface definition might be inherited from an object type, but store component properties including connections in a separate, copied aspect. Therefore, Function Designer uses the following two aspects:

- The Function aspect that is typically inherited.
- The Function Parameter aspect that is typically copied.



Function Designer automatically creates, copies, and deletes the Function Parameter aspect and the Function aspect. Do not create, copy, or delete the Function Parameter aspect manually.

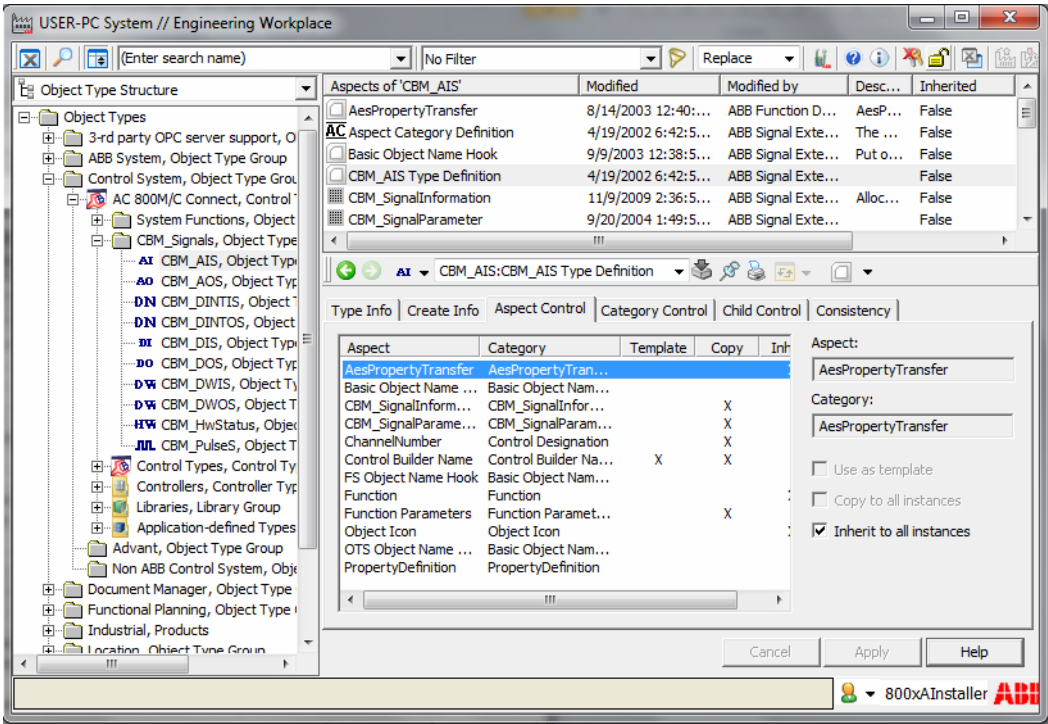


Figure 7. Example Object Type Definition with Function Aspect and Function Parameters Aspect

Aspect Properties/Parameters

Aspect properties/parameters can be defined in Component view and in Diagram view for the selected function component, function diagram or port. You can define properties that get inherited from the object type, or properties that get copied on object instantiation. Inherit ones are stored within the function aspect, and copied ones within the Function Parameter aspect.

Use the **Aspect Properties Dialog (Ports Tab)** dialog to create, edit, or delete aspect properties. Their scope depends on the selection:

Table 3. Scope of Aspect Properties / Parameters

View	Object	Scope
Component view	Selected port	Aspect properties for the selected port
	Any other selected object	Aspect properties for the whole component
Diagram view	Selected port	Aspect properties for the selected port
	Selected function component	Aspect properties for the selected component
	Any other selected object	Aspect properties for the whole diagram object

Aspect Properties/Parameters Dialog

Double-click an aspect object or variable to invoke the **Aspect Properties** dialog. Access the **Aspect Properties** dialog through the **Edit > Aspect Properties** menu command.

There are three different tabs based on the selection:

- **Component Tab**

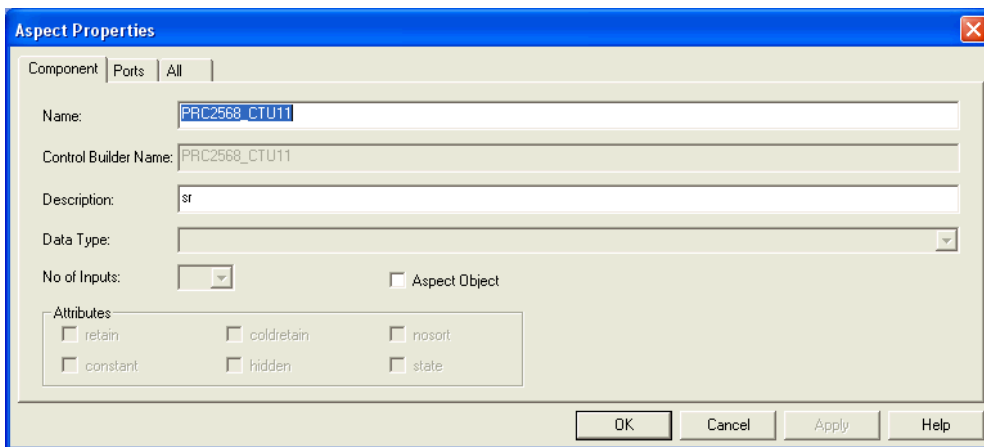


Figure 8. Aspect Properties Dialog (Component Tab)

The Component tab appears only if called on a function component only. Use it to edit or inspect the following fields of the component:

- **Name**
 - **Control Builder Name** (this field is dimmed and unavailable for editing)
 - **Description**
 - **Data Type** (for Variables, Diagram Parameters and Diagram References only)
 - **Number of Inputs** (for certain components e.g. AND, ADD, OR... which have got the possibility to control the number of single ports or port groups)
 - **Initial value**
 - **Aspect Object** property (for all function component to switch between Aspect Object and Symbol Object)
 - **Attributes** (for Variables, Diagram Parameters and Diagram References only)
- **Ports Tab**

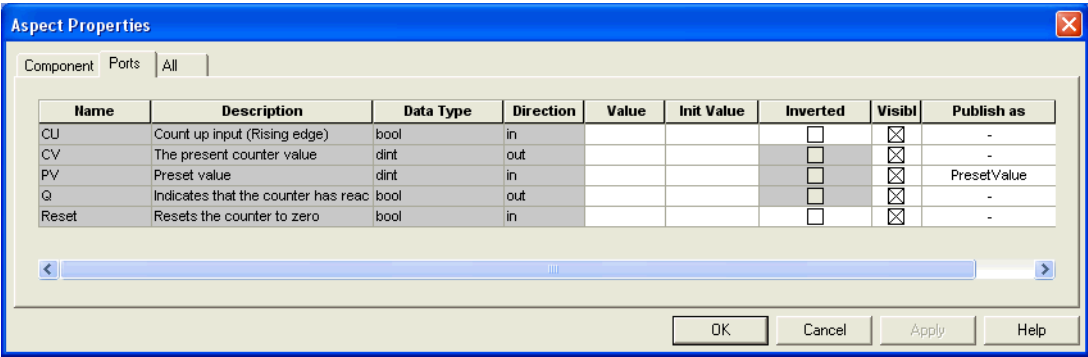


Figure 9. Aspect Properties Dialog (Ports Tab)

Ports tab appears only if called on a function component (for the ports of the component) or on the diagram background (for the diagram parameters).

The **Ports** tab can be used to edit or inspect the following properties:

- **Name** (this field is dimmed and unavailable for editing)
- **Description** (this field is dimmed and unavailable for editing)
- **Data Type** (this field is dimmed and unavailable for editing)
- **Direction** (this field is dimmed and unavailable for editing)
- **Value** (the connection at the port; for ports of direction **inout 2** **Value** fields show up (the left one for the input, the right one for the output))
- **Initial Value**
- **Inverted** (for boolean input ports only)
- **Visible** (checked for visible ports, unchecked for hidden ports)
- **Publish As** (a name that should be used to access the value property from the parent diagram; “-” or an empty string if the value property is not published (for more details see [Bulk Data Manager Support and Bulk Operations](#)); for ports of direction **inout 2** **Publish As** fields show up (the left one for the input, the right one for the output))

- **All Tab**

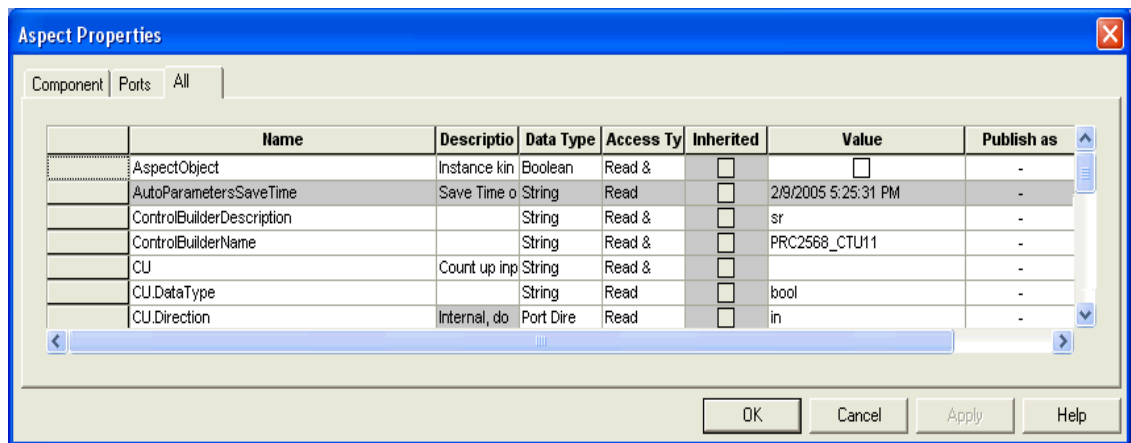


Figure 10. Aspect Properties Dialog (All Property Page)

All tab appears only if called on a function component (for all properties of the component) or on the diagram background (for all properties of the diagram). Use it to edit or inspect the following properties of the ports:

– **Name**

Defines the name of the aspect property/parameter. The name can be directly edited in the Name cell. Copied and inherited properties share the same name space.



Do not use a dot ‘.’ in the name, as a dot is used as a separator for structured properties.

– **Description**

Defines the description of an aspect property. Enumerated data types use this description internally “Internal, do not edit...”.

Port parameters selected in a diagram indicate their description in a tooltip.

– **Data Type**

Defines the data type of the aspect property. The data type can be selected from a drop-down list.



Do not take this data type as the data type of port parameters. Port parameters define a set of properties

- <Parameter>
- <Parameter>.DataType
- <Parameter>.Direction
- ...

see [Table 6](#). E.g. ‘Value = 3.14’, ‘Value.DataType = real’. The port parameter data type is ‘real’, whereas the data type of the properties ‘Value’ and ‘Value.DataType’ is ‘String’.

Besides the following default data types:

- Integer
- Real
- String

•Boolean

Two additional enumerations are listed:

1.**Port Direction** (noDirection, in, out, inout, plug, socket)

2.**Data Type** (only used for compatibility with old diagrams)

The enumerations are defined on the Function Aspect category as Function Enumeration aspect, see [Figure 11](#).



Function Designer does not support Re-Authentication or Double Authentication of Aspect Properties.

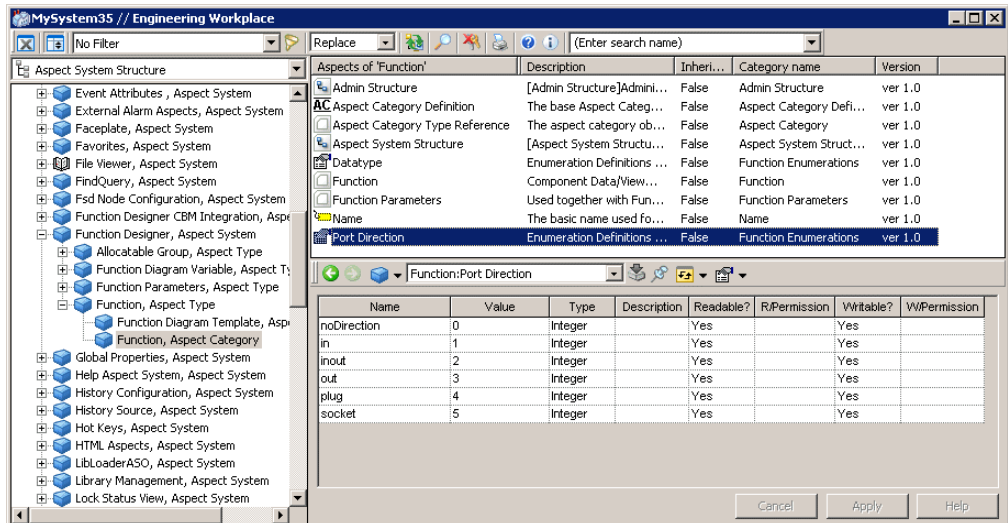


Figure 11. Enumeration Definition on Aspect Category

Port Direction defines the direction of a port parameter:

noDirection: general port parameter with no specific direction.

in: Input parameter in the sense of a IEC61131-3 Var_Input.

out: Output parameter in the sense of a IEC61131-3 Var_Output.

inout: In-Out parameter in the sense of a IEC61131-3 Var_InOut.

plug: snap port, reserved for future use in *Function Designer*.

socket: snap port, reserved for future use in *Function Designer*.

Data Type is no longer used, but it is needed to upgrade Function Diagrams created with former versions of Function Designer.

Additional enumeration data types can be added by adding further Function Enumerations aspects to the Function Aspect category, which is to [Aspect System Structure]Function Designer/Function/Function.

– **Access Type**

Defines the access type for the aspect property. The access type can be selected from a drop-down list.

None: Property is neither readable nor writable from other aspect systems or Bulk Data Manager. Used internally in Function Designer only.

Read: Property is readable from other aspect systems and from Bulk Data Manager, but not writable.

Write: Property is writable from other aspect systems and from Bulk Data Manager, but not readable.

Read & Write: Property is readable and writable from other aspect systems and from Bulk Data Manager.

Read permission and write permission of individual properties equals the read/write permission of the owning Function aspect or Function Parameters aspect.

– **Inherited**

Defines whether the property is inherited or not.

Inherited properties are stored within the Function aspect which is typically inherited from an object type.

Copied ones are stored within the Function Parameters aspect which gets typically copied from an object type.

– **Value**

Defines the value of the parameter

– **Publish as**

This is an enhanced setting allowing to read/write component properties/parameters through the parent diagram. It is mainly used when copying diagrams or setting diagram parameters through the Bulk Data Manager. For more details see [Bulk Data Manager Support and Bulk Operations](#). Enter the name that should be used to access the property from the parent diagram. Enter “-” or an empty string if the property should not be published.

Predefined Aspect Properties

The following aspect properties are predefined:

Table 4. Predefined Properties for Components

Property Name	Description
ComponentSaveTime (String) ParameterSaveTime (String) ObjectTypeSaveTime (String) ObjectTypeSaveInst (String) InterfaceSaveTime (String) InterfaceSaveTypeSaveInst (String) XMLTypeSaveTime (String) AutoParametersSaveTime (String) SymbolInstanceSaveTime (String) ReferenceSaveTime (String) AllocGroupsSaveTime (String)	Used internally to update open component views and diagram views with component instances. Always disabled for modifications.
DiagramReferenceType (String) VariableDataTypeIDs (String) VariableDataTypeSaveTimeInst (String)	Used internally to indicate the direction of diagram references and to update variable data types.
InstantiatedByDesignerVersion (String)	Function Designer version string that instantiated the component. Always disabled for modifications.

Table 4. Predefined Properties for Components (Continued)

Property Name	Description
NumberOfInputs (Integer) TypeOfInputs (String)	Current number and data type of inputs. Only available for components which support a variable number of inputs, e.g. AND (3).
VariableDataType (String) VariableDataTypeGroup (String) VariableAttributes (String)	Data type, type group (library or application) and attributes for local variables, diagram parameters, and diagram references. The read-only VariableDataTypeGroup property gets automatically updated on library connection changes on open diagram.
CrossCommunication (String) CrossCommunication.CycleTime (String) CrossCommunicationValid (String)	Used for input diagram references that need cross communication because of connections across different applications. See Prerequisites .
ComponentHidden (Bool)	If set to true, components are hidden on a diagram. Used e.g. for signal groups.
AspectObject (Bool)	If set to true, component is an aspect object and thus visible both in the diagram and its child in the diagram. If set to false, component is a symbol object visible on the diagram only.
ObjectTypeName (String) ObjectTypeGroup (String)	These read-only properties equal the object type name and the object type group (for library or application defined types). They get automatically updated on <ul style="list-style-type: none"> • Change Type (symbol objects only) • Library connection changes (aspect objects at once, symbol objects on open diagram)
ObjectName (String) ObjectDescription (String)	For aspect objects, these properties equal Name.Name and Name.Description. For symbol objects, these properties define name and description, respectively.

Table 4. Predefined Properties for Components (Continued)

Property Name	Description
ControlBuilderName (String) ControlBuilderDescription (String)	For aspect objects (CBM function blocks and control modules), these properties equal Control Builder Name.Name and Control Builder Name.Description. For symbol objects, these properties define CB name and description, respectively.
Position.Page (String) Position.X (String) Position.Y (String) Position.Unit (String)	Page number and top-left position of a component, given in units of the diagram's measurements (0.01mm or 0.001 inch). These properties are written on File > Save diagram. On open diagram, the position and page number of a component is synchronized against these properties. This is useful to bulk position components by e.g. Bulk Data Manager, refer to Create Diagram(s) through Bulk Data Manager .

Table 5. Predefined Properties for Diagrams

Property Name	Description
DiagramSaveTime (String) ParameterSaveTime (String) ContentsSaveTime (String) DelCompSymbolId (String)	Used internally to update open Diagram views. Always disabled for modifications.
DiagramIsBad (Boolean)	Boolean value indicating that recent diagram check has failed. Always disabled for modifications.
InstantiatedByDesignerVersion (String)	Function Designer version string that instantiated/created the diagram. Always disabled for modifications.
DiagramSize.Width (String) DiagramSize.Height (String) DiagramSize.Unit (String)	Diagram width and height, given in units of the diagram's measurements (0.01mm or 0.001 inch). These properties are written on File > Save diagram.

Table 6. Predefined Properties for Port <Parameter>

Property Name	Description
<Parameter> (String)	Port parameter value. Holds connection string, see Connections/Networks .
<Parameter>.In (String)	Port parameter value for inout parameters. Holds connection string, see Connections/Networks .
<Parameter>.Out (String)	Port parameter value for inout parameters. Holds connection string, see Connections/Networks .
<Parameter>.DataType (String)	Port parameter data type, e.g. "RealIO".
<Parameter>.Direction (Port Direction)	Port parameter direction, e.g. "in".
<Parameter>.Initval (String)	Port parameter instance specific initial value.
<Parameter>.InitvalOfObjectType (String)	Port parameter initial value defined at the object type.
<Parameter>.Inverted (Boolean)	True if port is inverted. Only available for invertable ports, see Inversion .
<Parameter>.Visible (Boolean)	True if port is visible, false if port is hidden, see Change Type .

Adding an Aspect Property

Perform the following to add an aspect property:

1. Select the object to which an aspect property (for example a symbol's port) is to be added.
2. Open the **Aspect Properties** dialog by one of the following methods:
 - Double-click the required component.
 - Right-click and select **Aspect Properties...** from the context menu.
 - Choose the **Edit > Aspect Properties** menu command.
3. Select the **All** tab (A new aspect property can be added only from the **All** tab)

4. Right-click the free background of the **All** tab.
5. Select **Add Property** from the context menu. A new empty row is inserted.

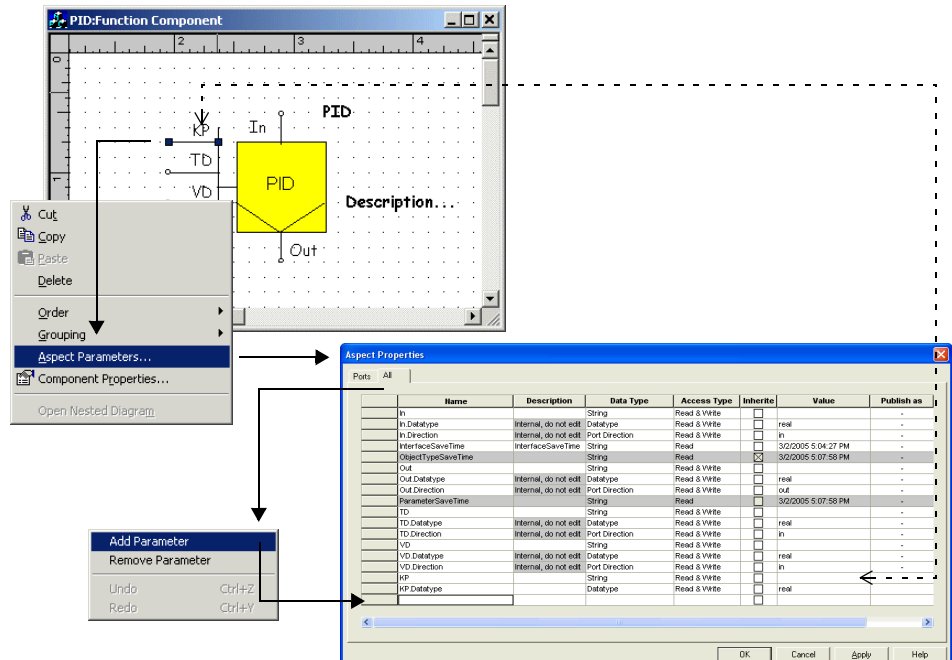


Figure 12. Adding an Aspect Property

6. Enter the required values into the fields using the drop down lists offered for the fields **Access Type**, **Data Type** and **Value**.
7. Perform steps 3 - 5 three times (for <portname>, <portname>.Datatype and <portname>.Direction) to define all the necessary properties of a port.
8. Click **OK/Apply**.

Removing an Aspect Property

Perform the following to remove an aspect property:

1. Select the required object from which the aspect property has to be deleted.


2. Open the **Aspect Properties** dialog by one of the following methods:
 - Right-click and select **Aspect Properties...** from the context menu.
 - Choose the **Edit > Aspect Properties** menu command.
3. Select **All** tab (An aspect property can be deleted only from the **All** tab).
4. Right-click the row containing the aspect property which is to be deleted.
5. Choose **Remove Property** from the context menu.
6. Click **OK/Apply**.

Aspect Property References

Instead of setting static text into a text or label component, references can be defined to any aspect property / parameter. By that, the text or label subscribes to and displays dynamic data, e.g. live data from an OPC data provider, a PID gain factor, or some document field defined in a document aspect.

The text or label components on a function diagram are automatically updated when the subscribed aspect property changes. So consistent and up-to-date function diagrams are obtained.


Perform the following to set a property reference in a text or label component:

1. Select the required object.
2. Right-click and select **Component Properties...** from the context menu to open the **Component Properties** dialog.
3. Select the **Field** tab.
4. Choose the **Aspect Property Reference** option in the **Field Code** group box.
5. Click  to open the **Aspect Property Reference** dialog.
6. Select **Structure**, **Select Object**, **Aspect** and **Property** from the corresponding list box. The **Reference String** text box displays the selection.
7. Choose the Reference Type and click **Apply**.
8. Click **OK**.

The **Aspect Property Reference** dialog is closed.

9. Click **Add** to enter the selected parameter reference into the **Field** text box.
10. Click **OK**.

Aspect Property Reference

Use the **Property Reference** dialog to define the **Reference String** for the required aspect property. Click  in the **Component Properties Field** page, to open the **Aspect Property Reference** dialog:

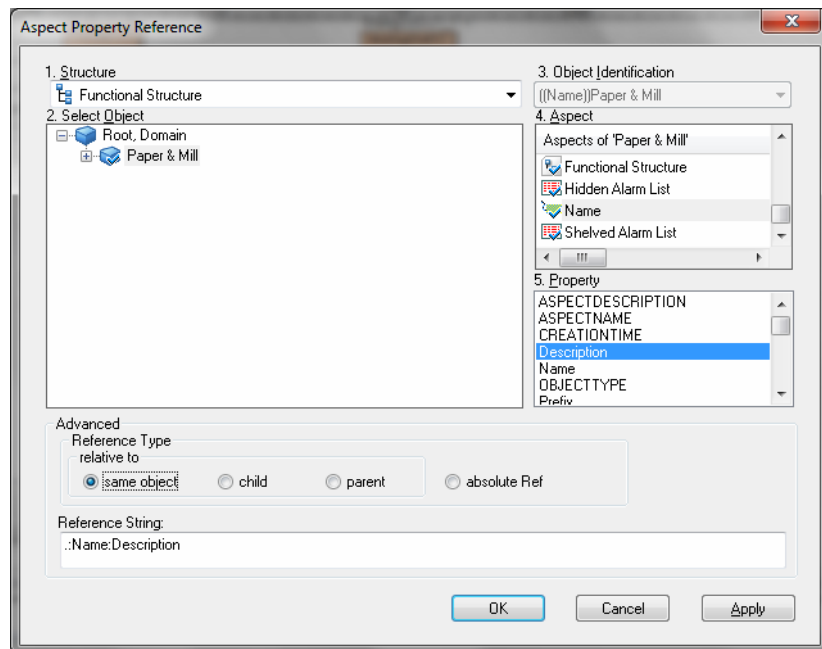


Figure 13. Aspect Property / Parameter Reference Dialog

For more information on the syntax for aspect property references, refer to *System 800xA Engineering, Engineering Studio (3BDS011223*)*.

Defining Function Components (Component View)

Function components can be defined in Function Designer's component view.

Perform the following to define a function component:

1. Choose Object Type Structure from the structure browser of **Engineering Workplace** or **Plant Explorer Workplace**.
2. Browse to the location where the new function component is to be located.
3. Use the **New Object...** context menu to create a new object type using the **New Object** dialog.
4. Use the **New Aspect...** context menu to create a Function aspect for the Object Type.
5. Select the Function aspect from the list of aspects, and open the Function Designer's Component view.
6. Use the **File > New** command to define a new function component.
7. Design the symbol of the function component using the graphical facilities of Function Designer.
8. Add aspect properties, if any. Refer to [Adding an Aspect Property](#).
9. For each (input/output) parameter of the function component, add at least the following aspect properties:
 - a. Parameter Name
 - b. Parameter Direction (in, out, in-out, none)
 - c. Parameter Data Type

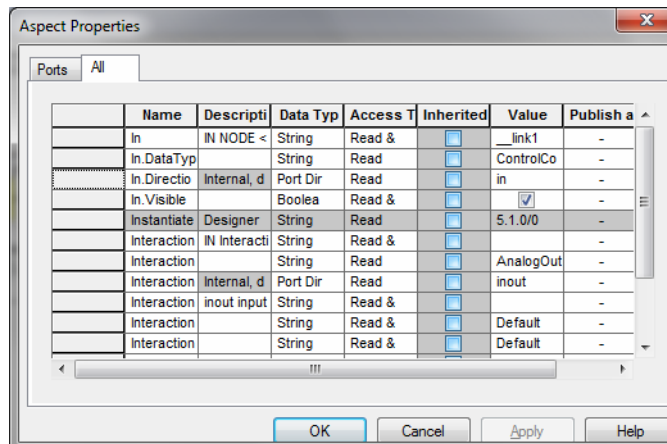


Figure 14. Aspect Properties

A complete function component including graphical symbol, ports, and aspect properties can be defined and this feature is available for:

- CBM_Signals.
- Diagram parameters.
- Diagram references.
- HART devices.

Perform the following to map a graphical port with such a parameter:

1. Open the port's **Component Properties** window.
2. In the **General** tab, enter the name of the required parameter.
3. Click **OK**.

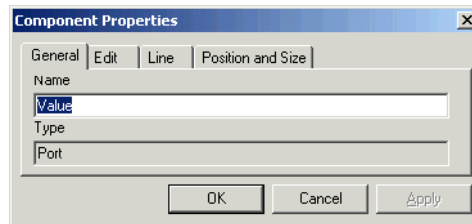


Figure 15. Port Name Mapped to (out) Parameter

Function Components with XML Type Description (Component View)

The following object types use an alternative to the complete definition of a function component including graphical symbol and aspect properties:

- CB system functions, refer to [Figure 16](#)
- CB system procedures
- CB function blocks, refer to [Figure 17](#)
- CB control modules, refer to [Figure 18](#)
- Variables

The Function aspect of these types store an XML type description with

- Type name
- Type description
- Interface description (parameters, data type of parameters)

- Optional extensible parameters

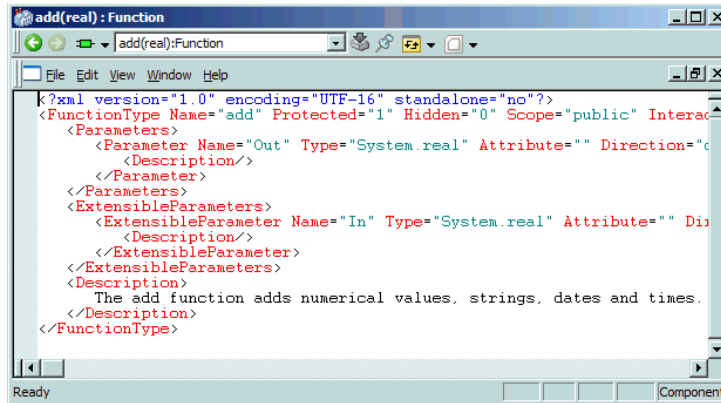


Figure 16. XML Type Description of an add (real) Function

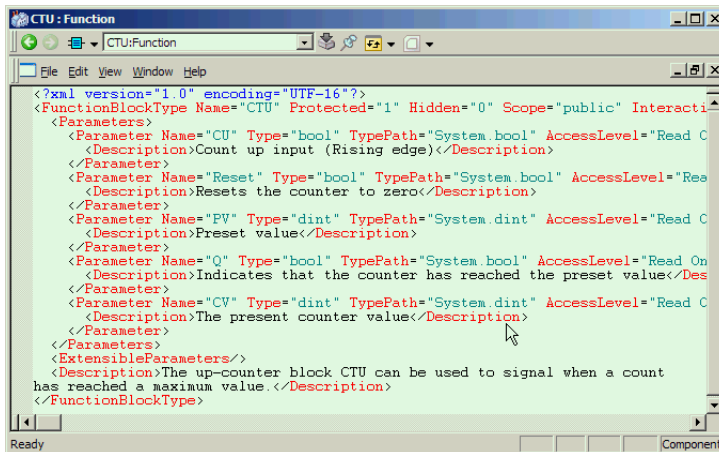


Figure 17. XML Type Description of a CTU Function Block

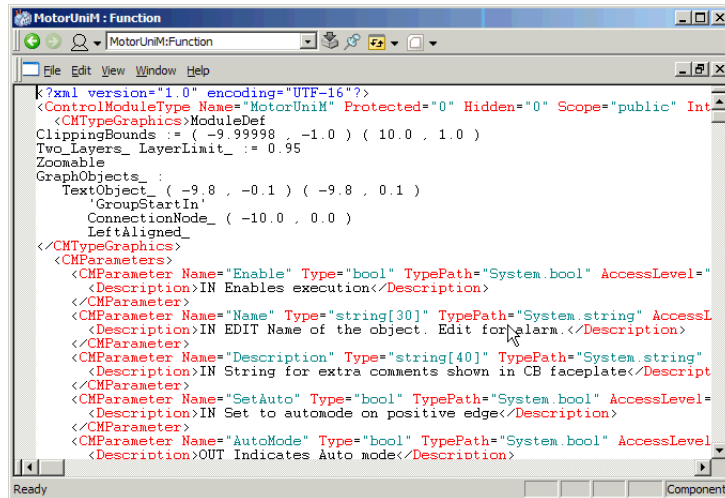


Figure 18. XML Type Description of a MotorUniM Control Module

On instantiation, aspect properties get automatically created from that XML type description, and a graphical symbol gets automatically calculated from a component template.

The component template is also stored within a Function aspect. It can be defined at one of the following four levels:

1. at the type.
2. at the parent type group.
3. at any parent of the type.
4. at the aspect category.

This also equals the search order for component templates. First a component template is searched at the type, then at its parents, and then at the aspect category.

The component template definition includes the following graphical components, see Figure 19:

- Label InstanceName.
- Text TypeName.

- Rect MainComponent.
- Rect ShadowComponent (optional).
- Polygon HiddenPortsIndicator.
- Text InputPortName.
- Port InputPort.
- Text OutputPortName.
- Port OutputPort.
- Text InoutPortName.
- Port InoutInputPort.
- Port InoutOutputPort.

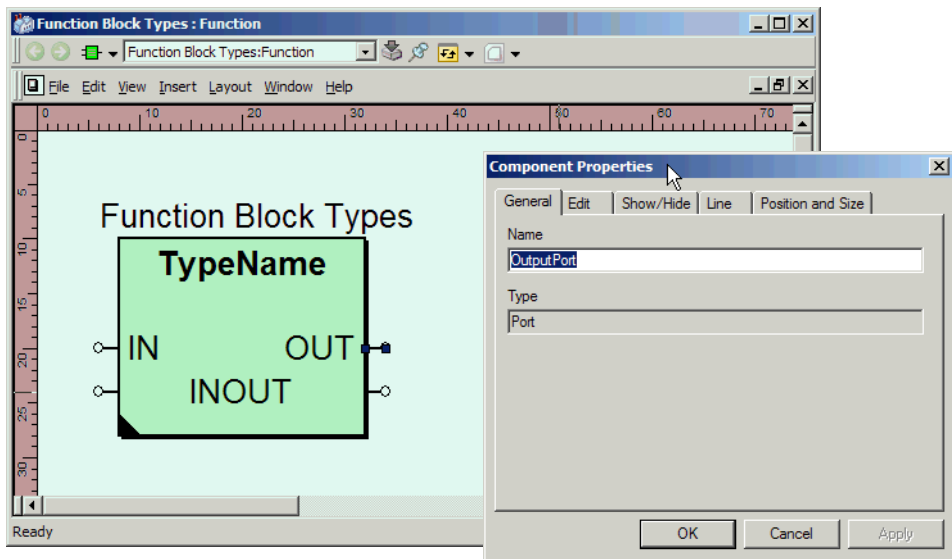


Figure 19. Component Template for Function Block Types

The XML type description for CB system functions and procedures is stored within the functions and procedures Function aspect. The corresponding object type group [Object Type Structure]Object Types\Control System\AC800 M/C

Connect\System Functions and a component template gets loaded by the system extension **Function Designer for AC800M Connect**.

The XML type description for CB function blocks and control modules is automatically retrieved from the corresponding CB library if a Function aspect gets created.

Libraries

For each base library loaded by an AC 800M system extension, Function Designer system extensions add corresponding Function, Function Parameter, and Object Type Definition Extension aspects through a corresponding extension library. The Object Type Definition Extension aspect is needed to add aspects to an already released library. To save the additional aspects for further use an extension library is recommended. See [Figure 20](#).

- **AC800M Connect** gets extended by **Function Designer for AC800M Connect** system extension.



For restored Function Diagrams, browse to **Connected Library > Connect Library** in Control Builder M and connect the latest version of BasicLib.

Perform the following for user-defined libraries with function blocks and control modules:

1. Open a Control Builder project and add the required library.
2. In the Library Structure create an extension library for the newly added library.
3. In Object Type Structure, add a Function aspect for each function block and control module type of the library. Use Bulk Data Manager to perform this action in bulk operation.

The Function Parameters aspect gets automatically created together with the Function aspect. An Object Type Definition Extension aspect gets automatically created if the library was already closed/released. If the library is still open, no Object Type Definition Extension aspect is necessary.

4. Optionally add function component templates to the types or to the parent type group.
5. Release Reservation for the extension library.

- Export the extension library (either in Open or Closed or Released state, according to further usage) by using the Export Library button in the Config View on the Extension Library Version Definition aspect. See [Figure 20](#).

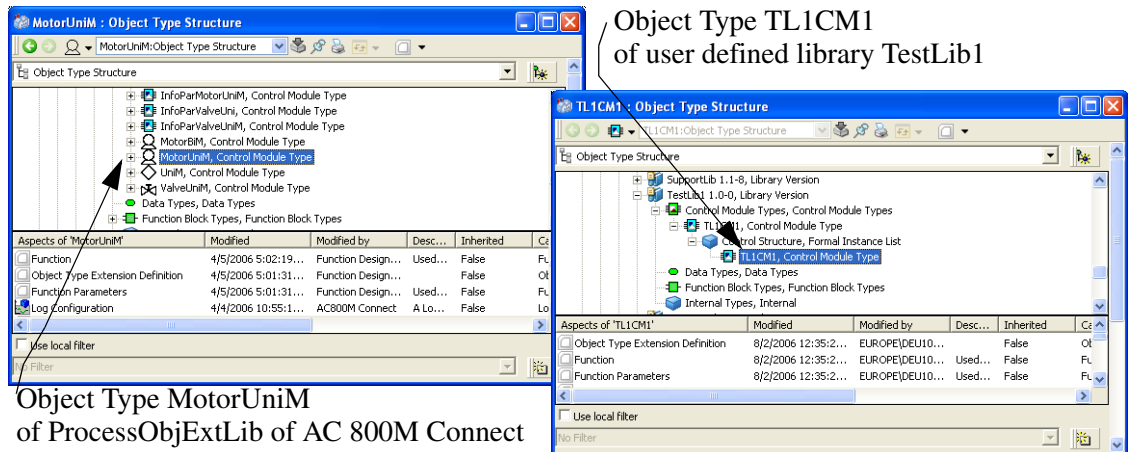


Figure 20. Library / Object Type Extensions Added by Function Designer

The XML type description stored in the Function aspect gets automatically updated on interface changes. Instances also get automatically updated on interface changes.

For instance, if:

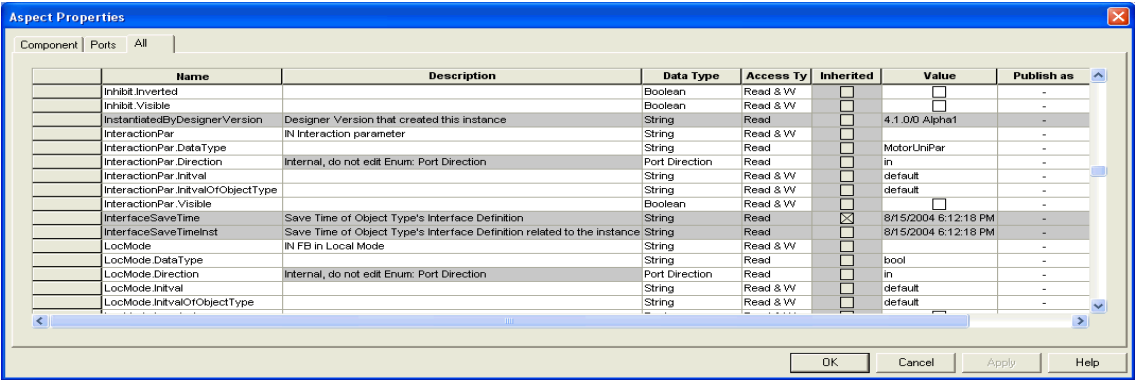
- there is a user-defined control module type MyCMT with two inputs and one output parameter.
- there are two instances MyCM1, MyCM2.

MyCMT.Function aspect holds a corresponding XML type description and inherits an aspect property **InterfaceSaveTime**. Instances copy that property as **InterfaceSaveTimeInst**. The instances MyCM1 and MyCM2 also display a symbol.

If the type's internals are only changed (add some glue logic or some internal variable), Function Designer detects that the interface is not changed. The inherited property **InterfaceSaveTime** is not modified and the instances, MyCM1 and MyCM2 are not updated.

If the types interface are changed (add/delete some parameter or change its data type), Function Designer detects that change and updates the inherited property **InterfaceSaveTime**. On opening the Function Diagram, the instances and their symbols MyCM1 and MyCM2 are updated due to the mismatch of the **InterfaceSaveTime** and **InterfaceSaveTimeInst** properties. The diagram state gets modified due to the changed interface, see [Instance Specific Initial Values](#).

The above scenario is applicable for interface changes due to changes in connected libraries. In such a case, aspect object instances get updated due to an internal Change Type. Symbol objects get updated on open diagram.



Name	Description	Data Type	Access Ty	Inherited	Value	Publish as
Inhibit.Inverted		Boolean	Read & W	<input type="checkbox"/>		-
Inhibit.Visible		Boolean	Read & W	<input type="checkbox"/>		-
InstantiatedByDesignerVersion	Designer Version that created this instance	String	Read	<input type="checkbox"/>	4.1.0/0 Alpha1	-
InteractionPar	IN Interaction parameter	String	Read & W	<input type="checkbox"/>		-
InteractionPar.DataType		String	Read	<input type="checkbox"/>	MotorUnitPar	-
InteractionPar.Direction	Internal, do not edit Enum: Port Direction	Port Direction	Read	<input type="checkbox"/>	in	-
InteractionPar.Initial		String	Read & W	<input type="checkbox"/>	default	-
InteractionPar.InitialOfObjectType		String	Read & W	<input type="checkbox"/>	default	-
InteractionPar.Visible		Boolean	Read & W	<input type="checkbox"/>		-
InterfaceSaveTime	Save Time of Object Type's Interface Definition	String	Read	<input checked="" type="checkbox"/>	8/15/2004 6:12:18 PM	-
InterfaceSaveTimeInst	Save Time of Object Type's Interface Definition related to the instance	String	Read	<input type="checkbox"/>	8/15/2004 6:12:18 PM	-
LocMode	IN FB in Local Mode	String	Read & W	<input type="checkbox"/>		-
LocMode.DataType		String	Read	<input type="checkbox"/>	bool	-
LocMode.Direction	Internal, do not edit Enum: Port Direction	Port Direction	Read	<input type="checkbox"/>	in	-
LocMode.Initial		String	Read & W	<input type="checkbox"/>	default	-
LocMode.InitialOfObjectType		String	Read & W	<input type="checkbox"/>	default	-

Figure 21. InterfaceSaveTime and InterfaceSaveTimeInst Aspect Properties



Do not delete and recreate the Function aspect in order to update a type. If the Function aspect is recreated, the InterfaceSaveTime property gets reset, and all diagrams will change their state to modified, even if the interface of the type did not change.

An update of the XML type description can be forced by the Function aspect verb **Update XML Type Description**.



Do not use **Delete from All Structures** in Object Type Structure.

Data Types

Variables of library- or application defined (structured) data type calculate their graphical symbol on the base of a variable component template and the selected **DataType** aspect. Variable instances get automatically updated on an open diagram in case of:

- changes in the data type definition.
- changes due to connected libraries.

This is done by the aspect properties `VariableDataTypeSaveTimeInst` and `VariableDataTypeIds`. The latter points to the `DataType` aspect.

Customizing a Function Component Type

The default function component type can also be customized by any of the following methods:

- [Adding a Function Icon.](#)
- [Copying and Modifying an Existing Function Component.](#)
- [Creating a Complete Function Component Definition.](#)

Adding a Function Icon

A Function icon aspect supports the display of Function icon picture components in the block representation of Functions, Function Blocks, and Control Modules as shown in [Figure 22](#).

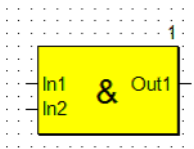


Figure 22. Function Block with Function Icon

To add a Function icon:

1. In the Object Type Structure navigate to the object type to be equipped with the Function icon.
2. Create a Function Icon aspect.
3. Open the Function Icon aspect.
4. Click **File > New**.
5. Select **Component**.
6. In the empty Function Component window draw the icon symbol.
7. Open the Aspect Properties dialog and add the new aspect property PictureComponentType (Data Type = String, Value = Centered).
8. Additionally the properties PortNameVisibility (Data Type = Boolean) and TypeNameVisibility (Data Type = Boolean) may be needed to define the layout of standard function like AND, OR, etc. where no port and type names are displayed.
9. Click **OK**.
10. Click **File > Save**.
11. Navigate to [Object Type Structure]Functional Planning.Settings
12. Open the Function Settings aspect and insert the new property UseFunctionIcon (Value = True, Type = Boolean).
13. Click **Apply** and **Cancel**.

As a result an instance of the object type on a function diagram will show the configured Function Icon, for example as shown in [Figure 22](#).

Allowed values for the PictureComponentType property are:

- Centered: The icon is positioned at the center of the main component.
- XCentered: The icon is only centered horizontally (relative to the main component) and keeps its y position as defined.
- YCentered: The icon is only centered vertically (relative to the main component). In contradiction to Centered is it located nearby the port names and therefore reduces the width of the resulting symbol (but leading to an asymmetric layout).

Function Icon aspects shall be collected in the extension libraries as described in [Libraries](#).

Copying and Modifying an Existing Function Component

For more information on copying and modifying an existing function component, refer to the sub section “Creating a Signal Group Object Type (includes copying / changing the graphical symbol)” in *System 800xA Engineering, Engineering Studio Function Designer Getting Started (3BDS100968*)*.

Creating a Complete Function Component Definition

As an example in this subsection we will change the representation of the function component type **ValveUniM** located in the **ProcessObjExtLib** (Process Object Extended Control) library of the AC800M Connect system extension. Changing the function component type will automatically update all instantiated function components with inherited (not overridden) Function aspect.

The object type exists and has a Function aspect. The view of the graphics, e.g. the components symbol definition and or the ports are to be changed. Prerequisite is that the Extension Library that extends the corresponding AC800M Connect library is in state Open.

The Function aspect (Component view) of control module type **ValveUniM** only defines a XML type definition. To add a customized function component definition (fixed symbol) for the representation in a function diagram, do as follows:

1. Select the function aspect from the **Aspects of ‘ValveUniM’** list.
2. Choose *Function Designer’s* **File > New** menu command. The **New** dialog appears.
3. Choose **Component** and click **OK**. A new window is opened in *Function Designer* to define the (fixed) graphical symbol.

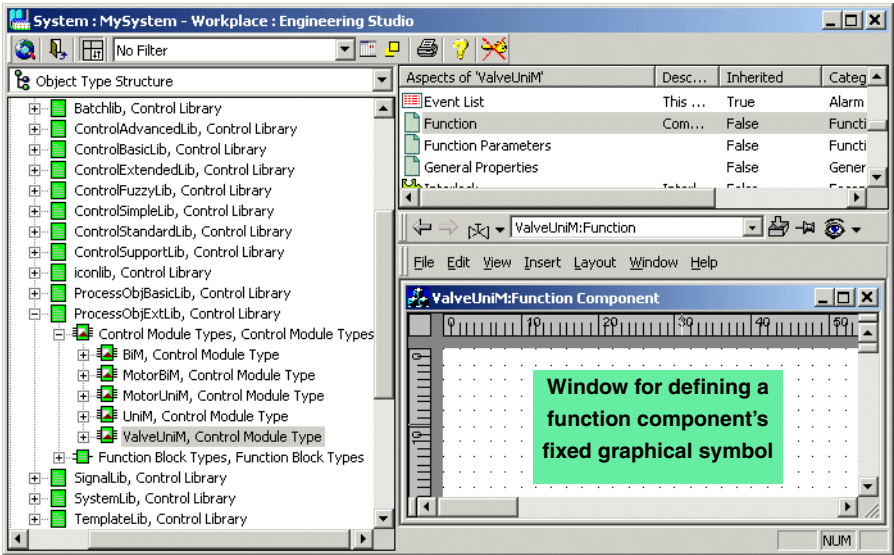





Figure 23. Creating Complete Function Component Definition (Fixed Symbol)

In *Function Designer* Component view, use the menu and/or toolbar commands (see [Working with Toolbars/Working with Toolbars](#)) to build up/modify the symbol for representing the function component in function diagrams. Table 7 lists some toolbar buttons which can be used for creating/modifying a symbol. The final customized symbol is displayed in Figure 24.

Table 7. Tools for Creating a Symbol

Tool	Description
	To use for drawing the box (rectangle).
	To use for adding line ports for connections to the symbol.
	To use for adding a text component to a symbol.

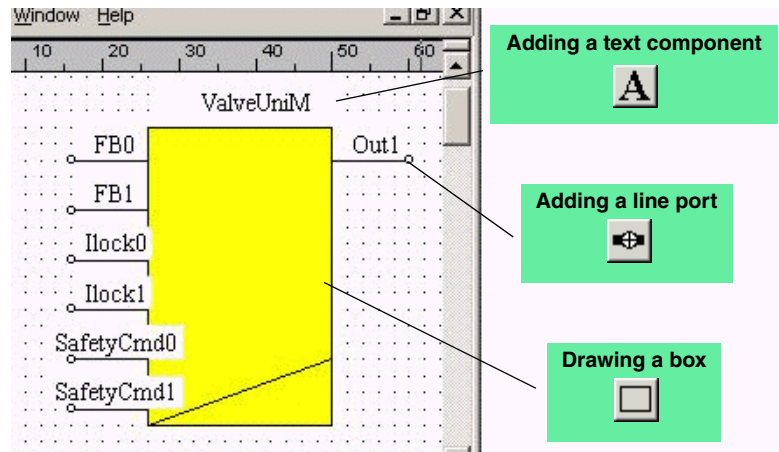



Figure 24. Customized Symbol for a ValveUniM Control Module Type

To customize the symbol:

1. The Component view of the function component type should be still open. If not, right-click the Function aspect in the **Aspects of** list of the **Workplace** window and choose **Component** from the context menu.
2. Click  and draw a box as body for the valve symbol.
3. Right-click the rectangular area to open the context menu and select **Component Properties**. The **Component Properties** dialog appears.
4. Choose the **Fill** page in the **Component Properties** dialog and set the colors according to the following figure.

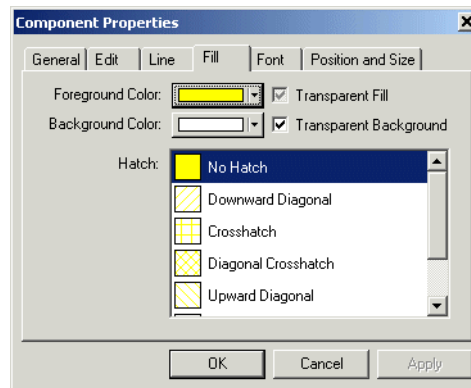




Figure 25. Setting symbol color.

5. Click  and draw line ports for the input and output connections.
6. Click  and insert text components for designating the ports according to the parameter name of the corresponding control module, function, or function block.
7. Set the references between the (line) ports and the parameters of the corresponding control module, function, or function block by entering the parameter names (used also as port designation in the previous step) into the **Name** field of the **General** page on the ports [Component Properties](#) dialog.

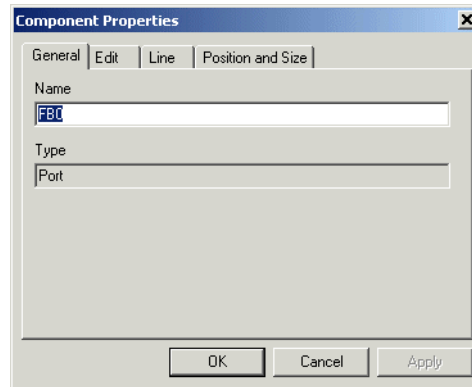




Figure 26. Setting the port name to the designation used in the symbol.

8. Click  and add a text component (symbol name) with an aspect property reference to **ValveUniM's** Name Aspect. Changes in the instance name are automatically reflected in Function Diagrams. Perform the following and also refer to [Figure 27](#):
 - a. Open the **Field** page on the [Component Properties](#) dialog.
 - b. Choose **Aspect Property Reference** and click on . The **Aspect Property Reference** window is opened, the **ValveUniM** control module type is selected.
 - c. Choose the **Name** Aspect's **Name** Property and as **Reference Type** **relative to same obj**, then click **OK**.
 - d. In the **Component Properties** dialog, click **Add**, then **OK**.

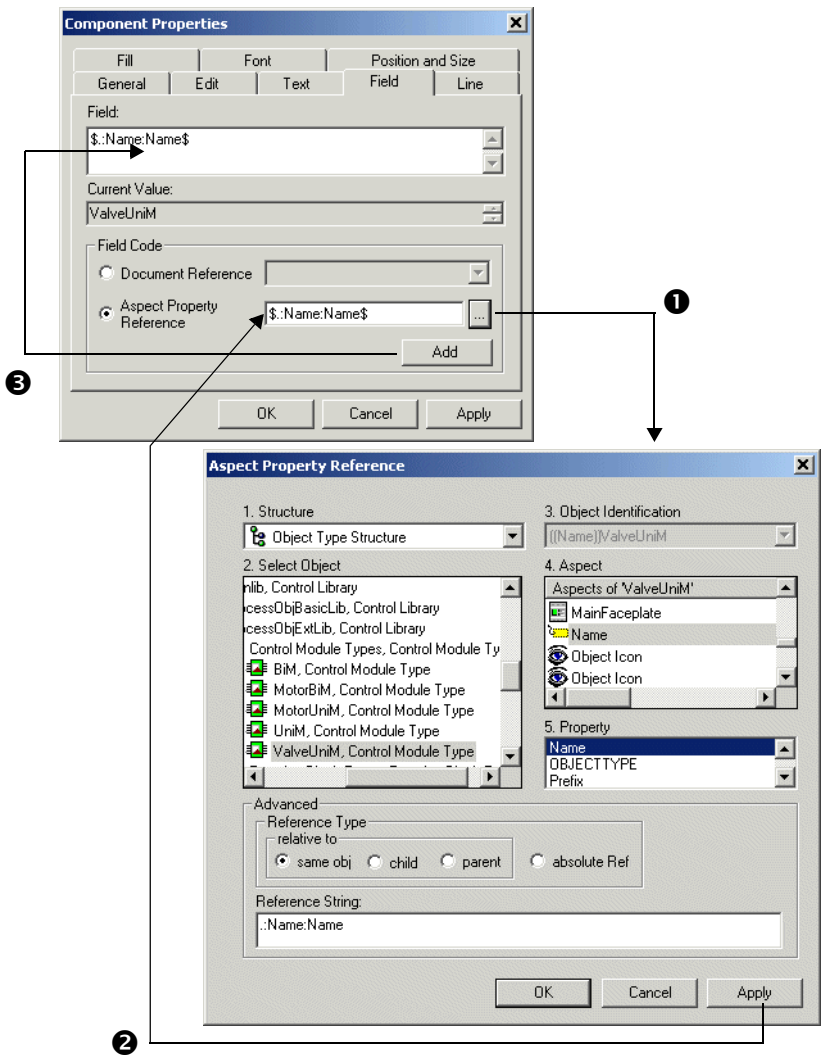


Figure 27. Setting a Reference to the Symbol Name



Line (circle) ports should be named exactly as the parameters of the associated control module, function or function block. By using this naming convention, a reference is created between the graphical port and the aspect property/parameter. The name referencing is case insensitive. The exact name of ports is visible in XML description.

Aspect Objects and Symbol Objects

Function components get instantiated

- either as aspect objects (visible in the diagram and in Functional Structure, optionally in Control Structure)
- or as symbol objects (stored and visible only in the diagram).

Symbol objects reduce the number of aspect objects. It is possible to switch the kind of existing instances from aspect to symbol object or vice versa:

- Either by component **Aspect Object**. Can be applied to a single or multi-selection of components in a Function Diagram
- Or by aspect property **AspectObject**, useful for bulk operations through Bulk Data Manager.

Table 8 lists objects that can be instantiated as symbol objects and as aspect objects.

Table 8. Symbol Objects and Aspect Objects

Default Symbol Objects	Default Aspect Objects
CB system functions CB system procedures Variables Splitters Joiners Diagram references	CB function blocks CB control modules CBM IO signals (significant data in aspects like Signal Parameter) HART devices (significant data in aspects like Fieldbus and Device Management)

Both aspect objects and symbol objects support the aspect properties listed in Table 9.

Table 9. Symbol Object and Aspect Object Properties

Aspect Property	Description
ObjectName	Equals <code>::Name.Name</code> for aspect objects
ObjectDescription	Equals <code>::Name.Description</code> for aspect objects
ControlBuilderName	Equals <code>::Control Builder Name.Name</code> for aspect objects
ControlBuilderDescription	Equals <code>::Control Builder Name.Description</code> for aspect objects
AspectObject	Set to true for aspect objects Set to false for symbol objects.

The instantiation kind for new instances on a diagram is determined by the Function Settings:**InstantiateComponents** property. It can take the values described in [Table 10](#).

Table 10. Function Settings:*InstantiateComponents*

InstantiateComponents	Description
AccordingCB AccordingCBAlarmOwnerFlag	Default from 800xA 4.0. Functions, procedures, variables and diagram parameter get instantiated as symbol object. Function blocks and control modules get instantiated as in CB: If they hold the AlarmOwner or the InstantiateAsAspectObject flag, they get instantiated as aspect object. E.g. an TOn gets instantiated as symbol object, a MotorUniM as aspect object.
AccordingTypeDefCreateInfo	Default in 800xA 3.1. See Object Type Definition Create Info . Functions, procedures, and variables get instantiated as symbol object. Diagram parameter, function blocks and control modules get instantiated as aspect object. E.g. a TOn gets instantiated as aspect object.

Table 10. Function Settings:*InstantiateComponents* (Continued)

InstantiateComponents	Description
AsAspectObject	Instantiate all objects as aspect object.
AsSymbolObject	Instantiate all objects as symbol objects. Exceptions are listed in Table 8 .

Objects created outside Function Designer are always instantiated as aspect object:

- Plant Explorer New Object dialog
- Bulk Data Manager

They can be changed into symbol objects by the Aspect Object or the AspectObject aspect property.



Always generate configuration data after adding or deleting any blocks created using Function Designer, which is used inside the Control Module Types.

In case error messages associated with unconnected ports appear during configuration data generation and user can not navigate to the respective blocks from the error message; then reopen the Function Diagram and delete the respective blocks before regenerating configuration data.

Object Type Definition Create Info

In case Function Settings:**InstantiateComponents** (see [Table 10](#)) equals **AccordingTypeDefCreateInfo**, the **Create Info** options in the **Object Type Definition** aspect determines if an object gets instantiated as an aspect object or as a symbol object:

1. Open Engineering Workplace or Plant Explorer Workplace.
2. Select the object type from the Object Type Structure.
3. Select the object type’s Aspect with category name **Object Type Definition** (for example, CBM_AIS Type Definition).
4. Select **Create Info** tab.
5. Check the options **Can be created as root**, **Can be created outside Structures** and **Can be created in all Structures** of the **Structure Info** group and modify them in the following way:

- To define instantiation kind aspect object, all **Structure Info** options must be selected, and **Explicit Object Type** must be cleared. See [Figure 28](#).

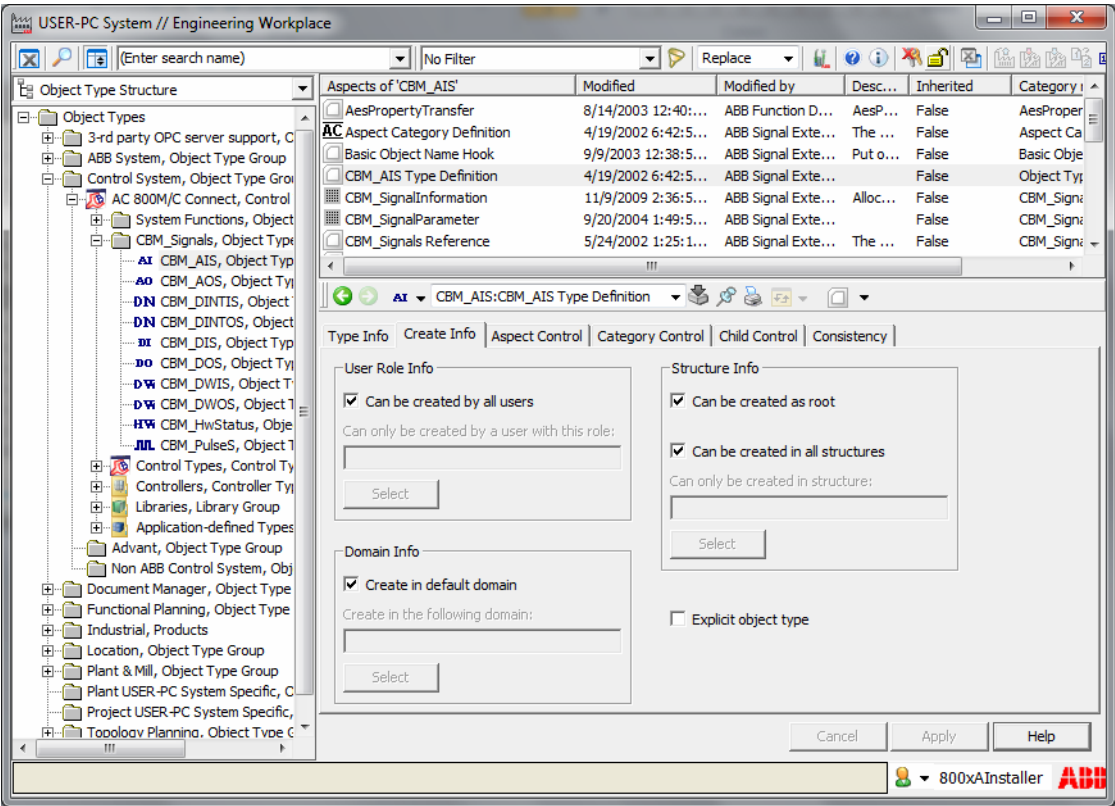


Figure 28. Create/Structure Info for Instantiation Kind Aspect Objects

- To define instantiation kind symbol object, clear the **Structure Info** options and select **Explicit Object Type**. See [Figure 29](#).

For glue logic like AND, OR and so on, the **Create Info** options are not set, that means such functions are instantiated as symbol objects and created and stored on function diagrams only.

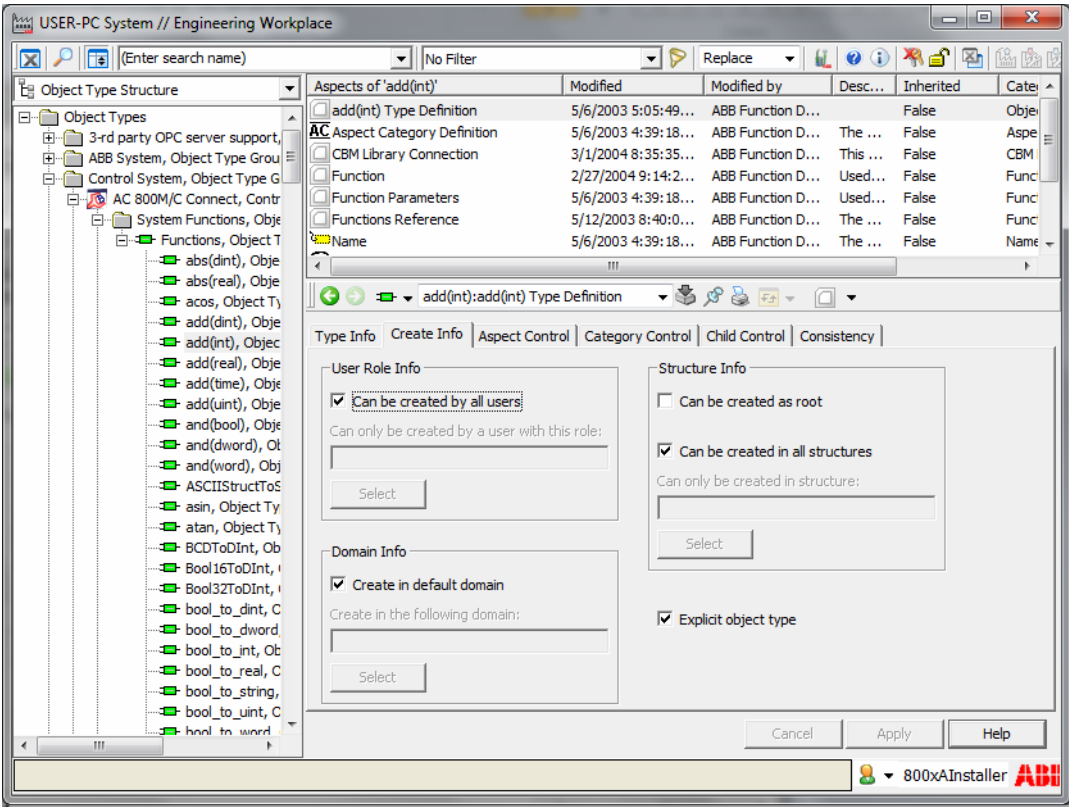


Figure 29. Create/Structure Info for Instantiation Kind Symbol Object

Inherited Function Components

If a new function component is created in a Function Diagram, the settings of the **Object Type Definition** aspect or **Object Type Definition Extension** aspect (for example, CBM_AIS Type Definition) define how the instance gets updated if its type changes

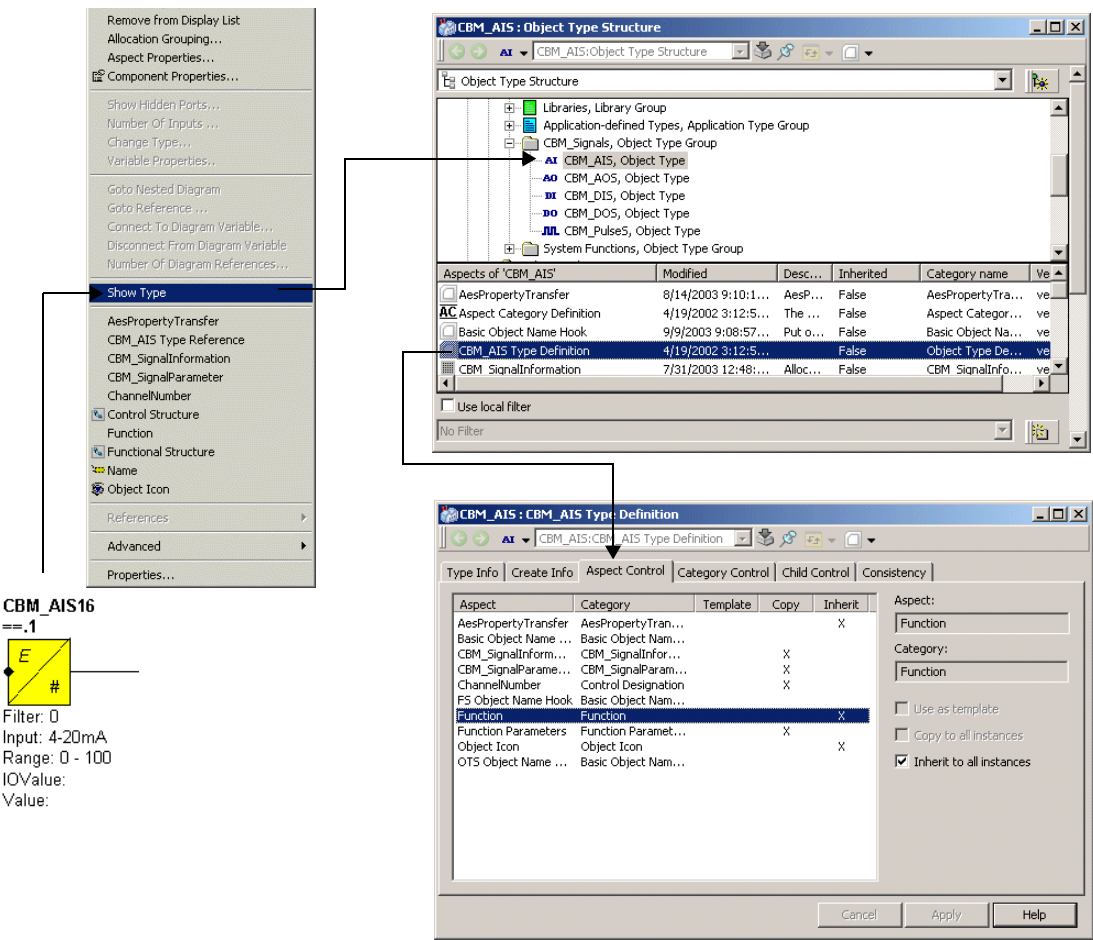


Figure 30. Aspect Control for Components with Inherited Function Aspect

- Set the option **Inherit to All Instances** for the Function aspect in the **Aspect Control** tab of the **Object Type Definition** aspect (for example, CBM_AIS Type Definition), to inherit the component's symbol definition from the object type to the instance. Modifications of the symbol definition - and thus of the Function aspect- will automatically result in an update of all existing object instances. In this case, the already existing components of the Function Diagrams are automatically drawn/printed using the updated symbol definition.
- Set the option **Copy to All Instances** for the Function aspect in the **Aspect Control** tab of the **Object Type Definition** aspect (for example, CBM_AIS Type Definition), to inherit the component's symbol definition from the object type to the instance. In this case, the function aspect will be copied on object instantiation. Later modifications on the type will have no effect on an already existing object instance.
- The Function Parameters aspect holding copied aspect properties and connect strings gets typically copied to all instances. However, formal instances of diagram types inherit the Function Parameters aspect in order to prevent modifications of the generated control module type.

Perform the following to adjust/modify the Aspect Control properties (as shown in [Figure 30](#)):

1. Open the **Object Type Structure** dialog by choosing the **Show Type** command of the component for which the object type definition aspect control is to be changed.
Alternatively, open the Object Type Structure in the Engineering Workplace and select the object type.
2. Select **Object Type Definition** aspect.
3. Open the **Aspect Control** tab and select the corresponding **Aspect**.
4. Set the **Copy/Inherit** options as required.

[Figure 31](#) displays indicators for an inherited component, which is a component with an inherited Function aspect:

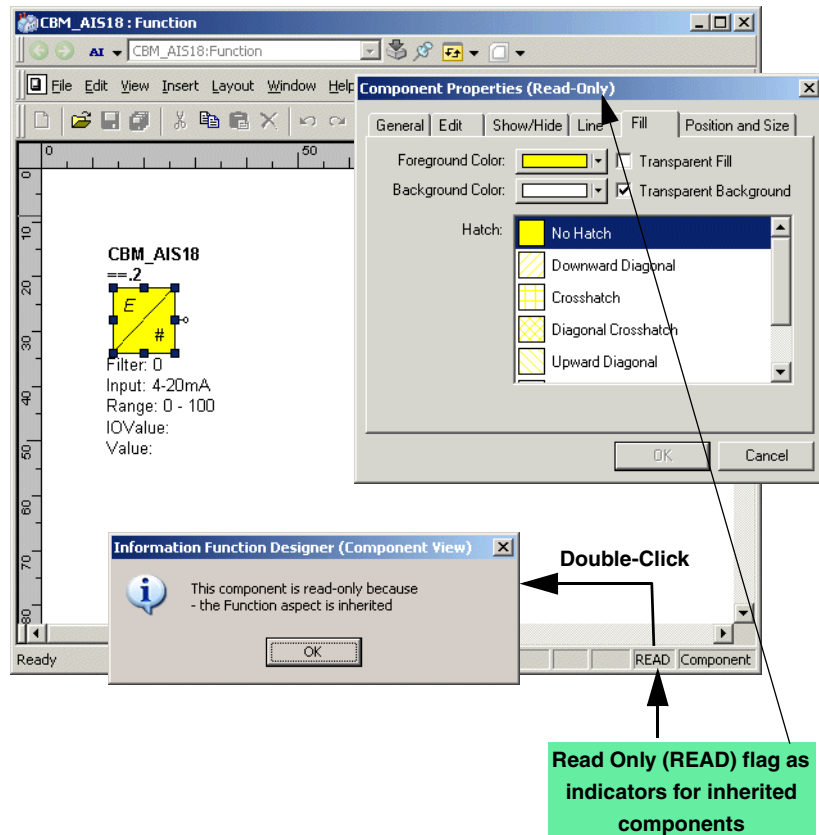


Figure 31. Indicators for Inherited Components

- **Read Only** hint in the Component Properties dialog.
- **READ** indicator in status bar of Function Designer. Double-click the indicator to get more detailed information such as “This component is read-only because its Function aspect is inherited”.

Labels on Inherited Components

Function Designer symbols (object types) can additionally use labels to define some property references. Component instances display informations such as the component name with such labels. As labels can be moved and modified at an instance, they are not inherited like the graphical symbol itself.

The following label modifications done at the object type are reflected at the instances on open diagram, if the label instances have not been modified manually:

- Fill properties.
- Line properties.
- Font properties.
- Text field properties.
- Text alignment.

Overridden Components

User can override inheritance of individual components as follows:

1. Select the function component (symbol) from Functional Structure.
2. Right-click the Function aspect from the **Aspects** list and select **Override** from the context menu.

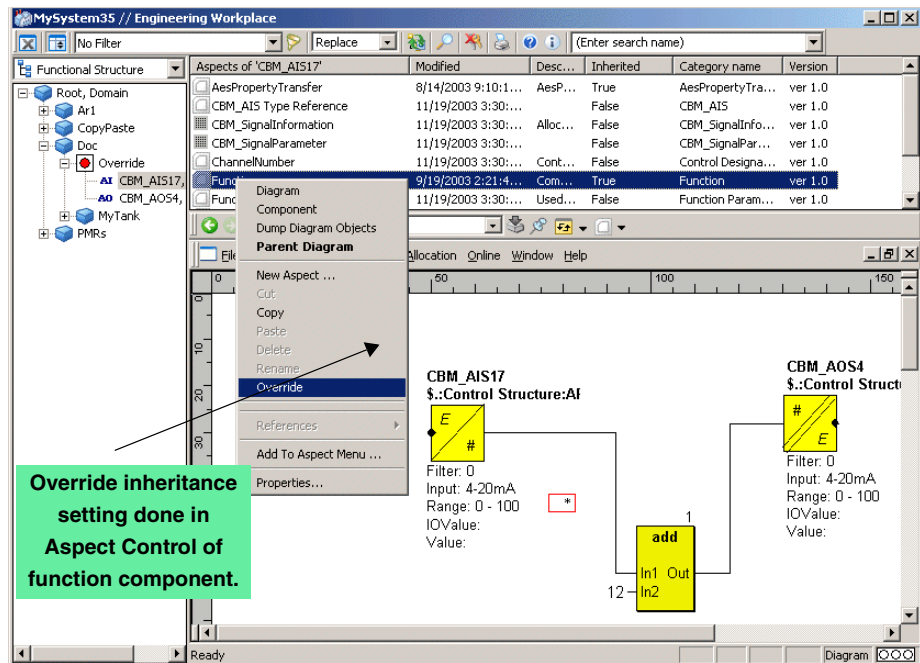


Figure 32. Override Inherited Function Component



When the user overrides an aspect, the aspect gets copied from the object type to the component instance. User can undo Override by the **Delete Override** command. In that case, the copied aspect gets deleted, so that the inherited one is seen again from the component instance.

Close and re-open the diagram to see the effect of **Delete Override**.

3. Modify the symbol definition in Component View.

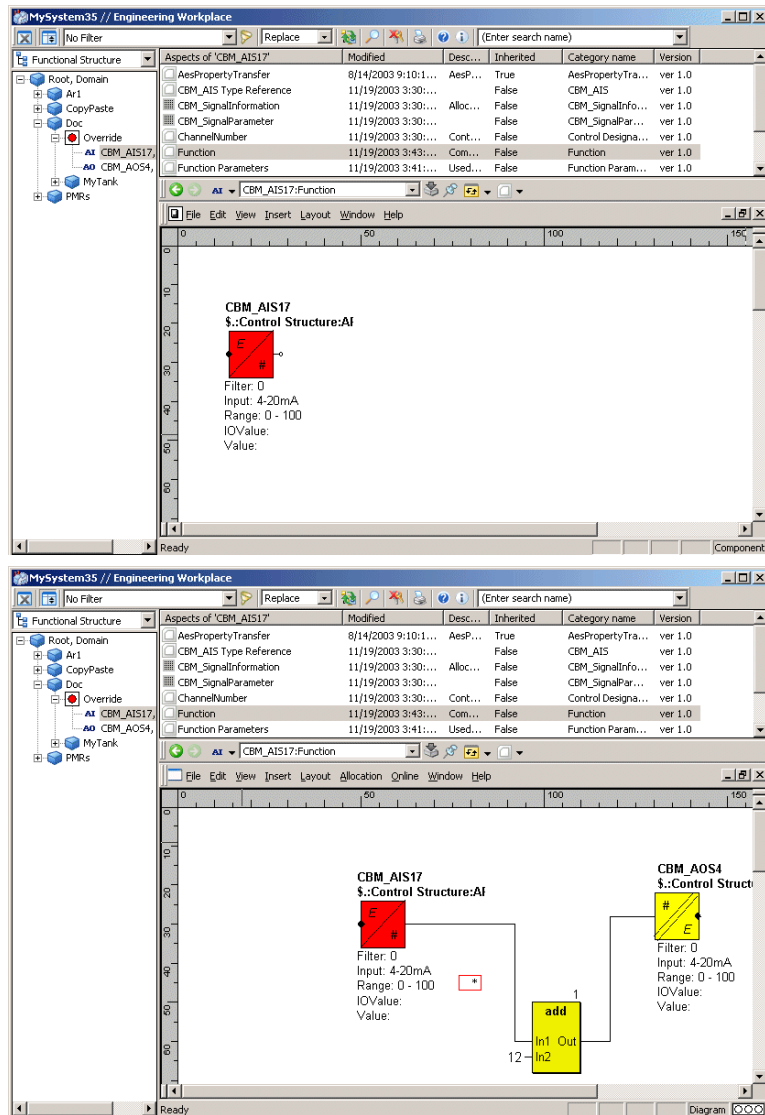


Figure 33. Diagram with Overridden Symbol

Default Aspect

User can define an aspect of an Aspect Object as default aspect. This default aspect will appear as uppermost entry in the context menu of the specific Aspect Object.

Perform the following to define an aspect (for instance: Function aspect) as the default aspect:

1. Select the Function aspect from an Aspect Object within the Plant Explorer.
2. Right-click and select **Details...** from the context menu.
3. Choose the **Aspect Info** tab in the **Details** dialog.
4. Select the **Default aspect** check box.
5. Click **Apply**.
6. Click **OK**.

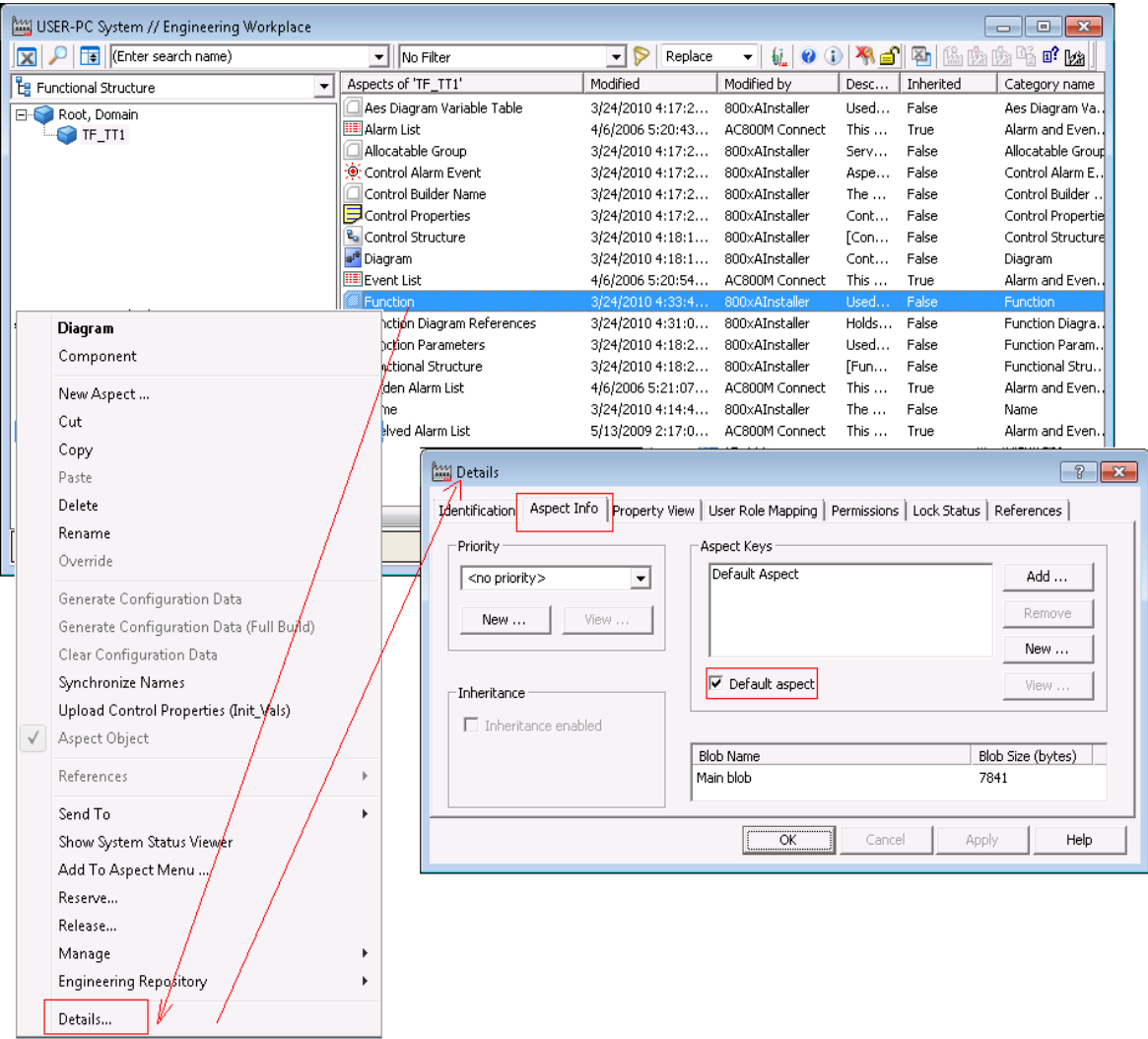


Figure 34. Defining Aspect as Default Aspect

Instantiating Function Components in Diagram View

User can instantiate Function components in the Diagram view of Function Designer through the menu command or through drag-and-drop. For more information, refer to [Inserting/Creating Components](#).

Change Type

Changing Type of Component Instances

The Change Type functionality is available for symbol objects and aspect objects. For aspect objects change type is only possible to a type of same kind. User can change a Function to a Function, Procedure to a Procedure, Function Block to a Function Block, and Control Module to a Control Module type. Connections and aspect properties are preserved as much as possible by keeping values for matching property names.

1. In Diagram view, right-click a component and select the **Change Type ...** command from the context menu. The menu command is disabled in case you selected an aspect object.
2. Select the new object type from dialog **Change Type**:

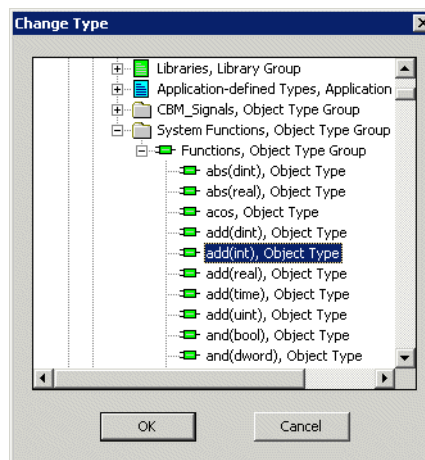


Figure 35. Dialog Change Type

Changing Number of Inputs

Certain components e.g. AND, ADD, OR... define an aspect property **NumberOfInputs** which controls the number of single ports (or port groups). The value of this aspect property is explicitly set when creating an object instance in a Function Diagram. When creating an object instance in Plant Explorer (or through Bulk Data Manager, etc.) a default value for this parameter is used.

Follow the steps below to change the number of inputs:

1. Right-click a component in Diagram view and select **Number Of Inputs ...** command from the context menu. This menu command is not available in case a component without modifiable number of input parameters is selected.
2. Select the new value from the **Number of Inputs** dialog as shown in [Figure 36](#).

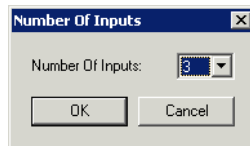


Figure 36. Dialog Number of Inputs

Changing the value of the **NumberOfInputs** aspect property is also possible through the **Aspect Properties** dialog (**Component** tab and **All** tab). It is recommended to use the **Component** tab from the **Aspect Properties** dialog or the **Number Of Inputs ...** command. When changing the value of the **NumberOfInputs** aspect property in the **All** tab of the **Aspect Properties** dialog or instead it might be necessary to close and reopen the diagram to update the symbol.

As the values of other existing aspect properties remain untouched, port connections are kept when increasing the value. When decreasing the value of **NumberOfInputs** the superfluous dependent port related aspect properties (and related connections) are removed.



While invoking the **Number of Inputs / Number of Outputs** dialogs, on functions or function blocks supporting extensible parameters (e.g. **In1..InX** for logic function **OR**), it may display a value different from the current configuration.

This does not affect the configuration and the graphical representation.

Show/Hide Ports

By default, newly instantiated components do not display all port parameters.

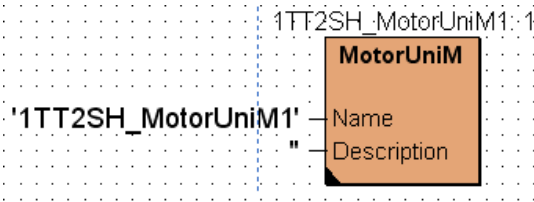


Figure 37. MotorUniM with Hidden Ports

Symbols with hidden ports are marked by a small black triangle displayed in the lower left corner, see [Figure 37](#).

The hidden ports can be revealed by the following steps:

1. Select a component in Diagram view with hidden ports.
2. Right-click the component and invoke the **Show Hidden Port(s)** command from the context menu.

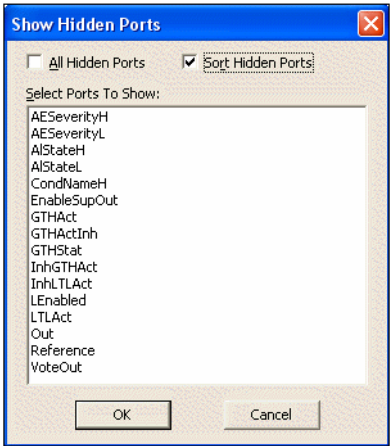


Figure 38. Show Hidden Ports Window

3. **Show Hidden Ports** dialog appears as shown in [Figure 38](#).
4. Select the **Sort Hidden Ports** check box to sort all port names in ascending order or alternatively, shortcut ALT+R can be used.
5. Select the **All Hidden Ports** check box or select the required port from the **Select Ports to Show** list.
6. Click **OK**.

Alternatively, user can set the **Visible** properties of the port parameters in the component's **Aspect Property** dialog (**Ports** tab), as shown in [Figure 39](#), or through Bulk Data Manager.

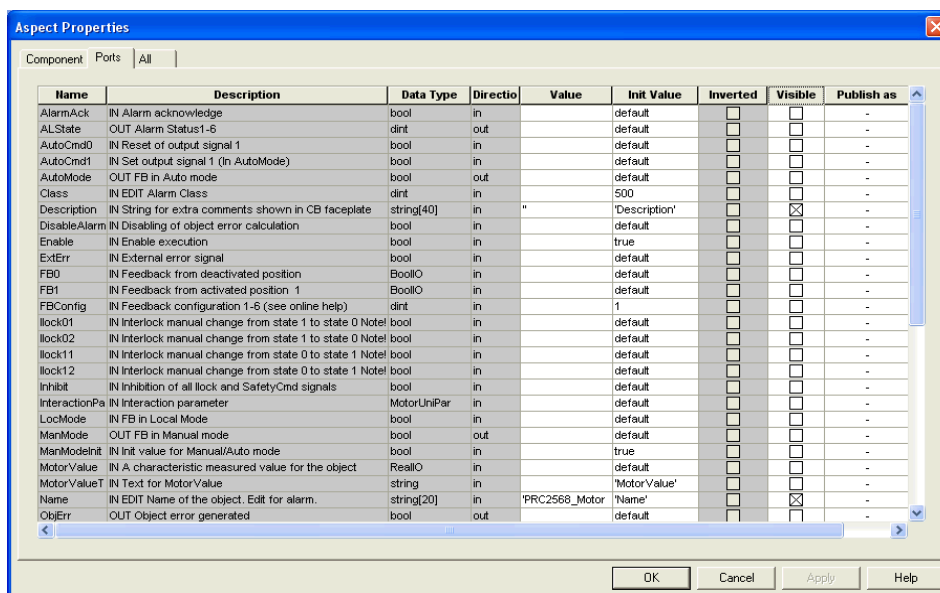


Figure 39. Port Parameters of MotorUniM



For information on how to configure the default visible ports of a function block, refer to the subsection **Configuring Default Visible Ports** in *System 800xA Engineering, Engineering Studio Function Designer Getting Started (3BDS100968*)*.

User can hide individual ports by the following steps:

1. Select the ports to be hidden from the Diagram view of the required component.
2. Right-click on the port and invoke the **Hide Port(s)** command from the context menu.

Connections/Networks

A **port** of a function component symbol references an aspect property/parameter by its name. For more information, refer to [Function Components and Diagrams](#).

This aspect property holds the **connect string**. A connect string can be of type **constant**, e.g. 123, or 'def', or of type **variable**, e.g. abc. It can be edited by any of the following methods:

- in the drop-down combo box of the [Connection](#) toolbar,
- in-cell in the connector symbol displayed at the port,
- in the [Aspect Properties Dialog \(Ports Tab\)](#),
- or in Bulk Data Manager.

If there are two ports on a diagram connected to the same variable, e.g. abc, a connection link is automatically created and routed (corresponding link will not be visible in the Function Diagram Editor in Control Builder M). If the ports are located in different pages, **off-page connector** symbols are automatically created on both pages. **CB project constants** use the same syntax as variables. But, they behave like constants and are not connected through links or off-page connectors.



Do not create a project constant in Control Builder M while a Function Diagram is open. Otherwise, if a project constant is connected twice on different ports the Function Designer is drawing a short circuit line. User must close and re-open the Function Diagram to get the proper connection displayed.

A **network** is a set of 1:1, 1:N connections between ports. A connection network has 1 source and 1...N sinks, and is defined by its unique network name, that equals the connect string of type variable. All network components, i.e. all links and off-page connectors, share the variable/network name as component name property. The variable/network name is also accessible by a label attached to network links.

Automatically created links get the default variable/network names link, link1, link2...

Consider the following examples:

- **Figure 40** shows symbol port S1:In1 connected to constant 123, and symbol ports S2:In1 connected to variable abc and S2:In2 connected to string constant 'def'. The connect strings are reflected in the ports aspect property, and in Bulk Data Manager. If user changes the connect string in the aspect property dialog, or in Bulk Data Manager, the diagram gets automatically updated.

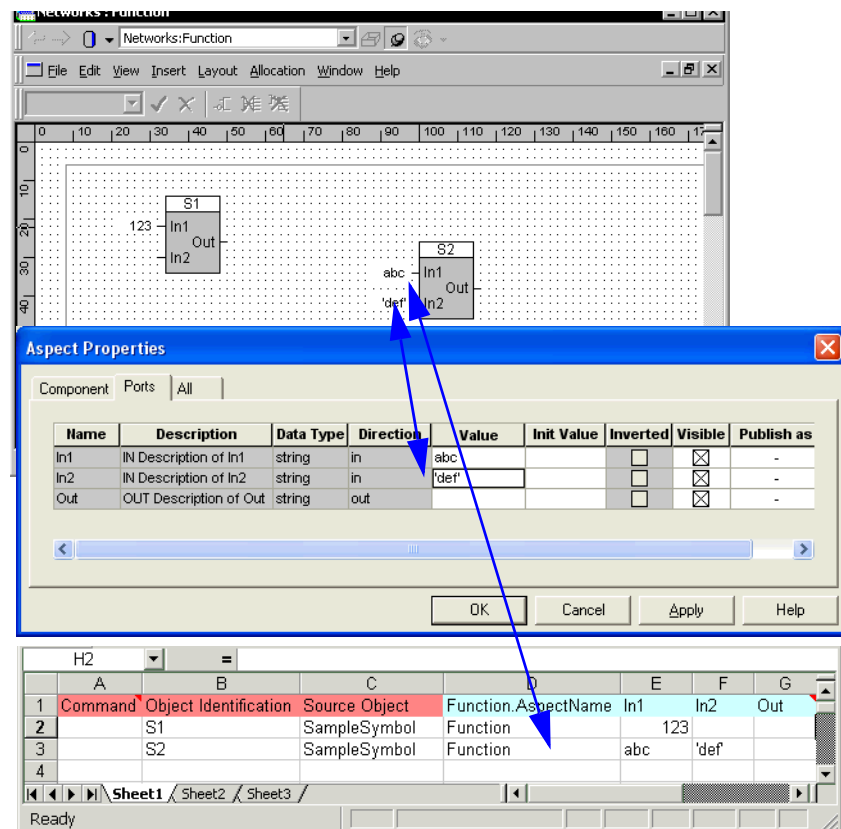


Figure 40. Connection to Constant and to Variable

- In [Figure 41](#), S1:Out was connected to variable abc. A 1:1 network with one connection link gets automatically created, connecting S1:Out with S2:In1. The link is labeled abc, and the links name property equals abc. This is also reflected in the aspect properties, and in Bulk Data Manager. And vice versa, user can automatically create networks with links by updating connect strings in the aspect properties dialog, or in Bulk Data Manager, see also [Create Diagram\(s\) through Bulk Data Manager](#).

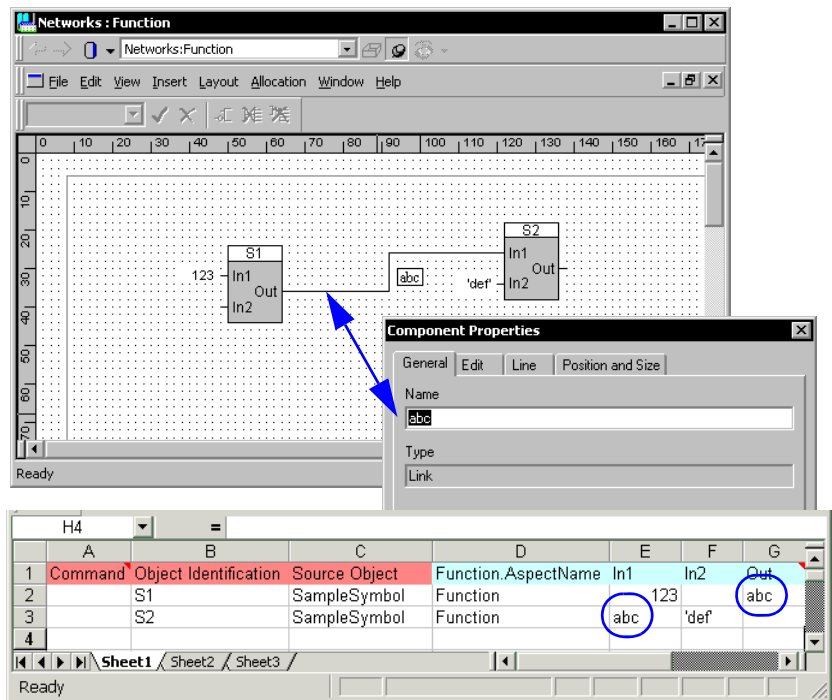


Figure 41. 1 : 1 Network

- In Figure 42, S3:In1 was connected to variable abc. A 1: N network, $N = 2$, gets automatically created with two links named abc.

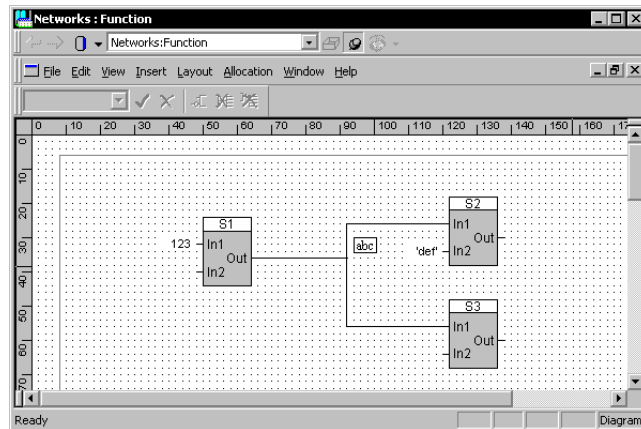


Figure 42. 1 : N Network ($N = 2$)

In Figure 43, S2 was moved from page 1 to page 2. Off-page connectors with page references get automatically created. Page referencing is not reflected in the aspect properties nor in Bulk Data Manager, i.e. S1:Out and S2:In still hold the connect string abc. In other words, off-page connectors are also automatically created when connecting symbols on different pages by updating aspect properties, e.g. in Bulk Data Manager.

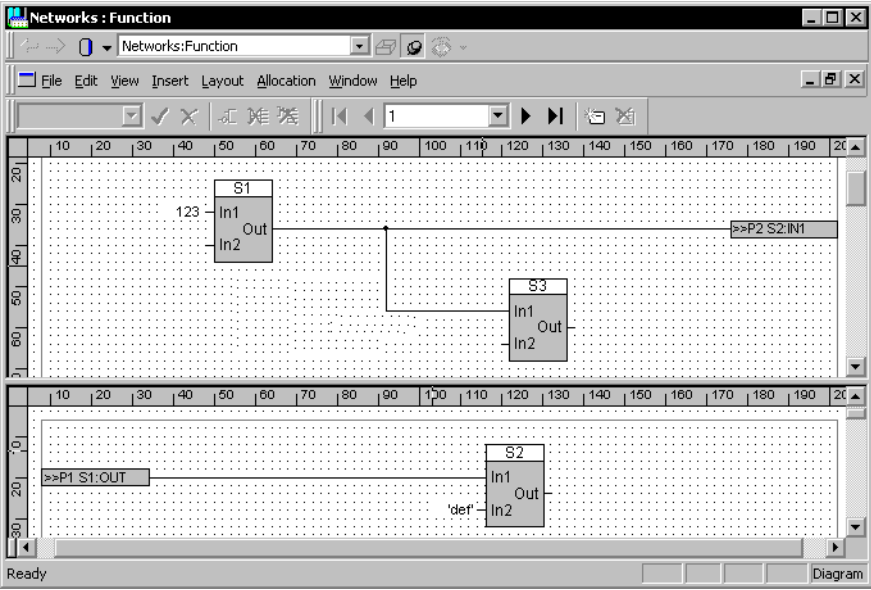


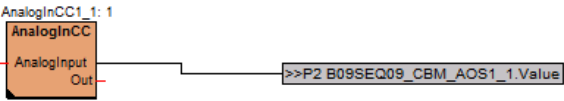
Figure 43. 1 : N Network with Off-Page Connectors

Network Connections between Pages

Interconnected function components can be moved to other pages considering the following connection rules:

- 1. 1:1 connections between pages:

1:1 connections between Function Components can be performed based on the direction rules in [Figure 71](#). Inout (Out) connection to other pages is not possible (refer to figure below).



2. 1:N Connections between pages
 - a. 1:N connections between pages can be created between System Functions, Function Blocks and Control Modules or between Variables like CBM signals, Local Variables and Communication Variables. Mixed networks are possible using Move block, Splitter / Joiner blocks between Off-Page connectors and variables.
 - b. Each Off Page connector sink needs a corresponding Off Page connector source as displayed in the figure below

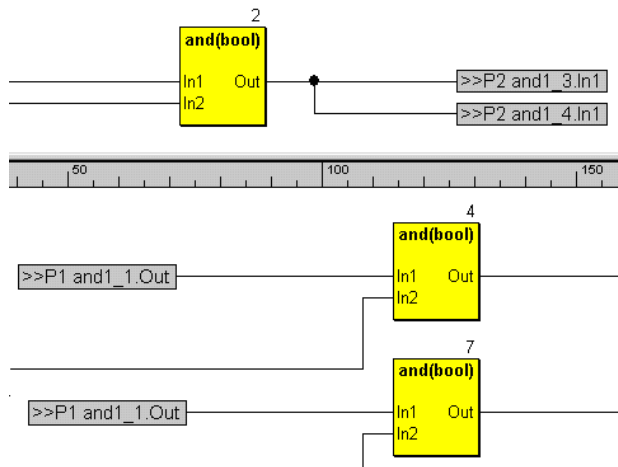


Figure 44. Connection Between Off Page Connector Sink and Off Page Connector Source

It is not possible to allot a connection on a destination page, to several other destination ports on the same page. For instance, refer to [Figure 44](#) it is not allowed to create a connection from AND(4).In1 to AND(7).In1 directly.

Guidelines for using Page Connectors

A few guidelines for using the page connectors are listed below.

- Usage of empty diagram references in page connectors results in error.

- When a communication variable is connected to a block via page connector and if the communication variable is disconnected from the input diagram reference, generating configuration data throws error.
- Connecting an InOut port to Out port via a page connector results in error during generation of configuration data.
- Do not make the following page connections.
 - a. OUT -> INOUT // Not allowed in CBM
 - b. INOUT->IN // Not allowed in CBM
 - c. INOUT-INOUT // Not allowed in CBM
- When a connection is made from Out port to InOut port via a page connector, if the port name and data type name are same then it results in an error during generation of configuration data.

Connect Using Keyboard

Perform any of the following methods below to enter a connect string:

- Select a port in the Function Diagram, edit the connect string in the drop-down combo box of the [Connection](#) toolbar, and press <Enter> key or click **Enter connect string**, as shown in [Figure 45](#).
- Select a port in the diagram and select an already existing connect string from the drop-down combo box of the [Connection](#) toolbar, see [Figure 45](#).
- Select a connector symbol displayed left or right at a port in the diagram, in-cell edit the connect string, and press <Enter> key.
- Select a function component symbol in the diagram and edit its connect strings in the [Aspect Properties Dialog \(Ports Tab\)](#) or [Aspect Properties Dialog \(Component Tab\)](#) dialog.

- Update connect strings in Bulk Data Manager.

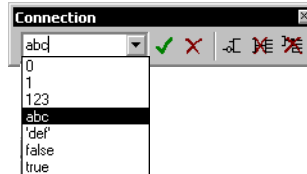


Figure 45. Connection Toolbar

To connect two function component ports on the same or on different pages using keyboard:

1. Navigate to the first port by using the cursor keys in Network traversal mode and mark it by 'Numpad'- refer to [General Accelerator Keys](#).
2. Navigate to the second port by using the cursor keys in Network traversal mode.
3. Press ALT + Numpad to connect the selected port to the marked one.
4. In case of connecting ports on different pages, off-page references are automatically created.
5. Network link(s) get created with next free default variable/network name link, link1, link2, ... and automatically routed.

Connect Using Mouse

There are two alternate methods to connect function component ports using the mouse as described below:

To create a new connection (alternative 1)

1. Press the left mouse button on the source port.
2. Move the mouse to the sink port or already existing connection on the same page.
3. Release the mouse button.
4. The connection link gets automatically routed. It gets the next free default variable/network name link, link1, link2... .

No modifier keys are used in this case and the user need not provide any 'connection mode switch on the toolbar'.

To create a new connection (alternative 2)

1. Click and select the required source port.
2. Click ALT+left on sink port on the same or another page.
3. In case of connecting ports on different pages, off-page references are automatically created.
4. Network link(s) get created with next free default variable/network name link, link1, link2, ... and also gets automatically routed.

If a connection (sink) is already existing, additionally a new vertex is created automatically.

Manual routing by inserting/freezing vertices on connection links

1. Click and select a connection link.
2. Click ALT+left on the connection link where a new vertex is to be inserted.
3. Click CTRL + SHIFT + left on the new vertex to pin it, i.e. to freeze it for auto routing. The pinned vertex is indicated in red color.
4. Unpin a frozen vertex by CTRL + SHIFT + left mouse click.

5. Delete a vertex by CTRL + left click.



In some instances, the automatically routed connection lines overlap. Move related symbols within diagram to overcome the overlapping.



In some instances, connections lines overlap after a close and reopen of a Function Diagram. Modification of distances between links and symbols by using **Edit > Options > Autorouting** overcomes this overlapping.

Disconnect

Perform any of the following to disconnect a port or some port(s) from a network:

- Select a port in the diagram, reset its connect string to an empty one in the [Connection](#) toolbar.
- Select a port in the diagram, and click 'Disconnect port only' or 'Disconnect complete network' from the [Connection](#) toolbar.
- Select a connector symbol displayed left or right at a port in the diagram, and delete it.
- Select a function component symbol in the diagram and reset its connect strings in the [Aspect Properties Dialog \(Ports Tab\)](#) dialog.
- Reset connect strings in Bulk Data Manager.

Inversion

Perform any of the following to invert an input port of data type 'bool':

- Select a port in the diagram, and press the 'Invert' button of the [Connection](#) toolbar.
- Select a function component symbol in the diagram and edit the port's inversion properties in the [Aspect Properties Dialog \(Ports Tab\)](#) dialog.
- Update inversion properties in Bulk Data Manager, see [Figure 46](#).

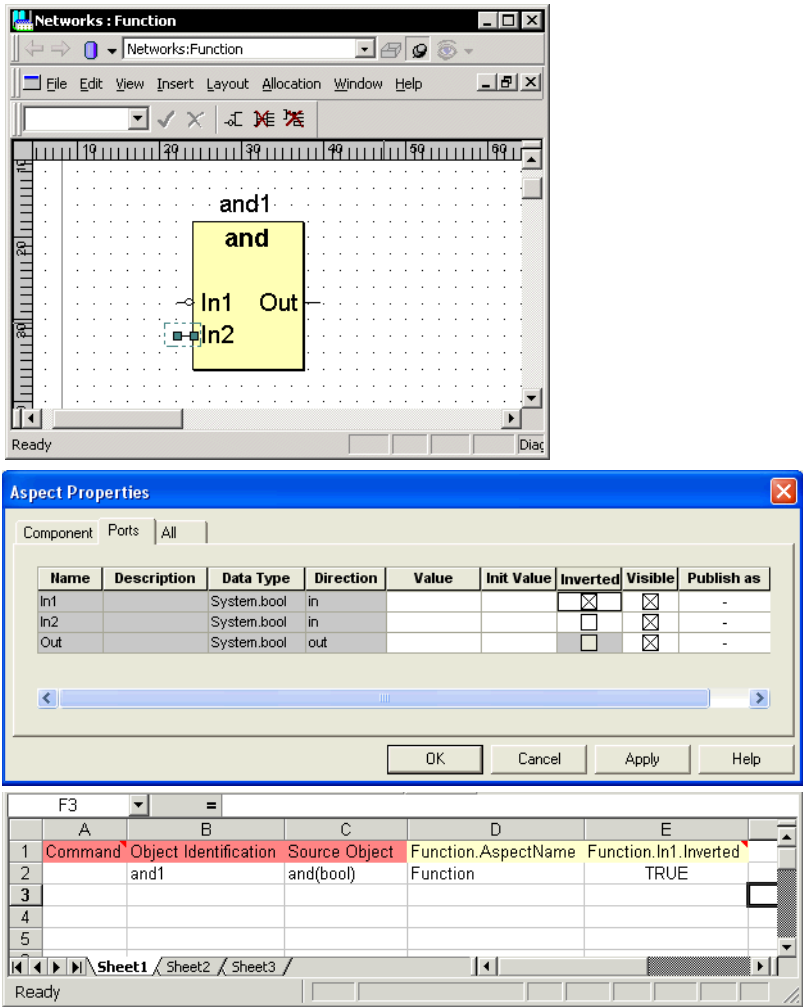


Figure 46. Inversion of Boolean Inputs

Vertical Navigation

For a better overview on Function Diagrams, it is possible to build a hierarchical Functional Structure with parent-child diagrams. Functions on a lower level can be summarized in a separate diagram and displayed in an overview diagram on a higher level as a single function component with input and output ports. Diagrams on lower level are also called child or nested diagrams. They can be opened from an overview diagram by choosing [View > Goto Child Diagram](#) or **Goto Child Diagram** from the context menu. This is also called vertical navigation - top down, bottom up, throughout the functional hierarchy.



For more information about vertical navigation through Single Control Modules, refer to the [Vertical Navigation](#).

Horizontal Navigation

Function Diagrams can also be split using **(Off-) Diagram References**. User can navigate through a group of Function Diagrams that are connected through diagram references. This is also referred to as horizontal navigation.

There is no restriction in connecting/referencing diagrams located anywhere in the Functional Structure. Double-click the required diagram reference to open and access the referenced diagram directly.

Alternatively, user can navigate from one diagram to another by any of the following methods:

1. Select the referenced object and choose [View > Goto Referenced Diagram](#).
2. Select the referenced object and choose **Goto Reference...** from the context.

Two variants of input or output diagram references are supported:

- **Reduced:**
Only the variable name and the path to the connected reference is displayed.
No Description fields for references and variable are displayed. They save drawing space on the diagram.
- **Non- reduced (as in earlier system versions):**
Additional Description fields for references and variable are displayed.

Inputs and outputs of referenced diagrams are function components of type [Object Type Structure]Object Types/Functional Planning/Generic Function Components/Connectors/Off Diagram Reference/**Input (Output) Reference** or **Reduced Input (Output) Reference**:

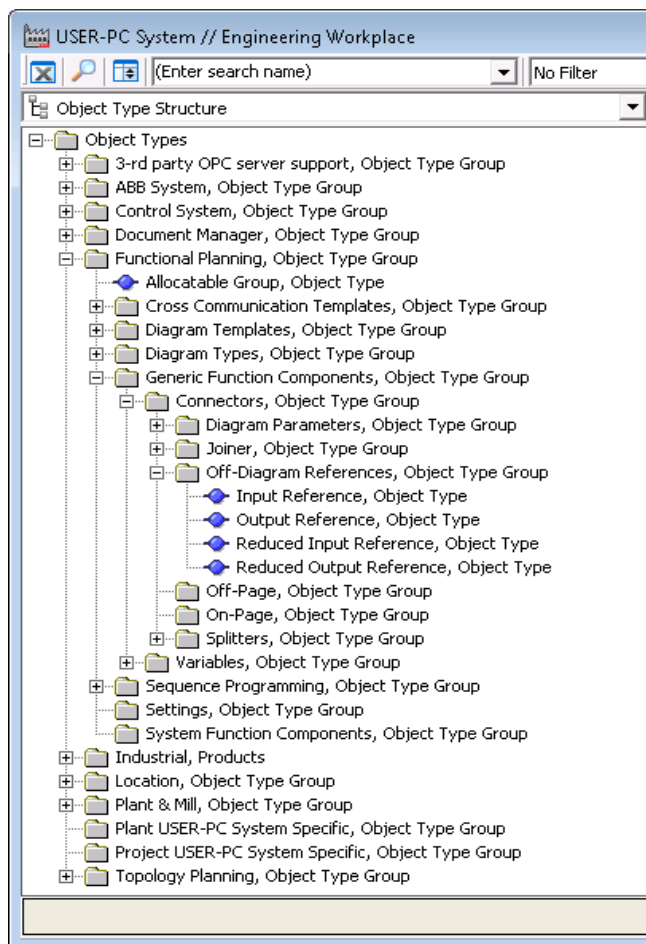


Figure 47. Diagram References in Object Type Structure

User can include diagram references to diagrams in Functional Structure only. They are not supported by diagram types in Object Type Structure. In [Figure 48](#), we have an overview diagram D0 and two diagrams D1 and D2 with (off-) diagram references. D1:Out is connected to D2:In, and D2:Out is connected to D1:In1 and D1:In2. This is indicated by automatically calculated cross-references in D1 and

D2. The blue colored field available in the diagram reference symbol as shown in [Figure 48](#), indicates that communication is established between the referenced diagrams.

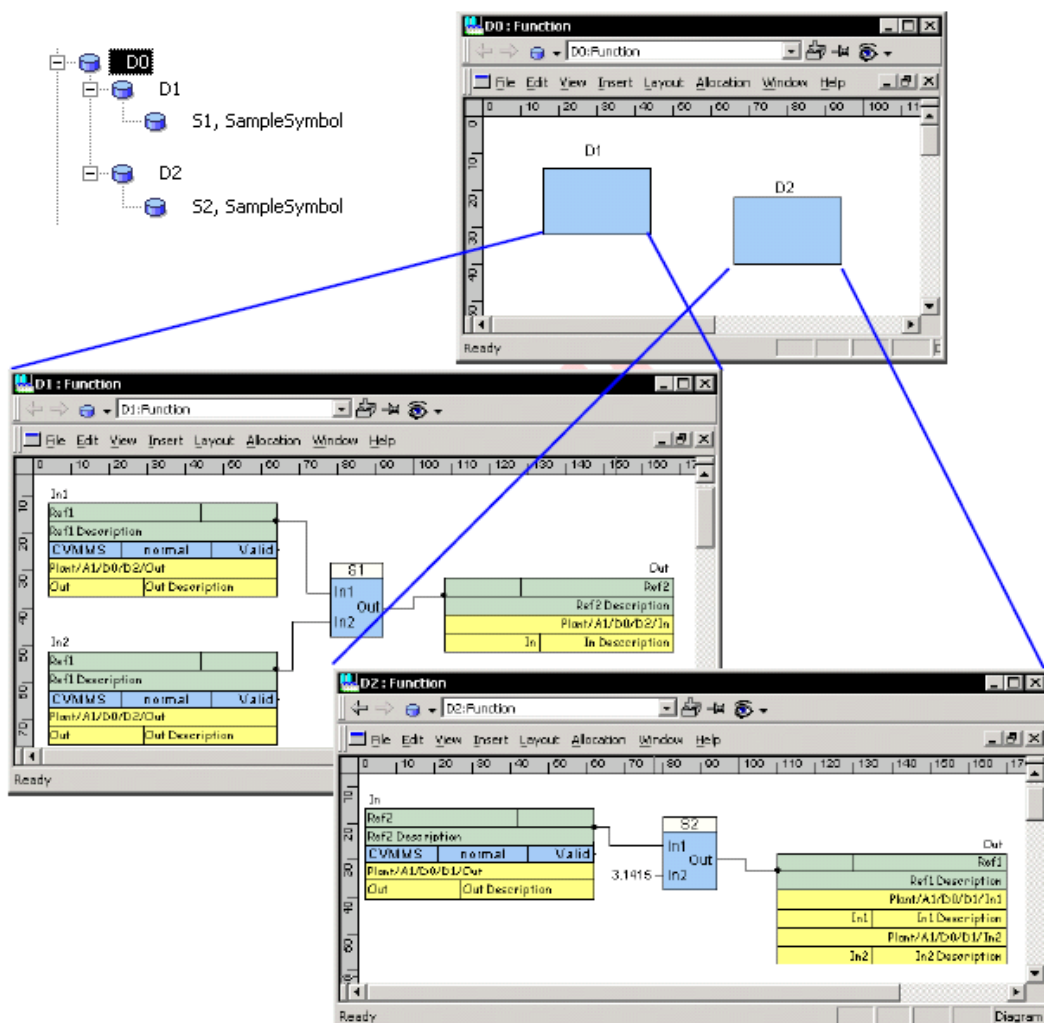


Figure 48. Referenced Diagrams

Reduced Input (Output) Diagram References only show the path information of the reference to save diagram drawing space.

Global Variable and Communication Variables

The global variable facilitates communication between restored diagrams (created upto System Version 5.0 SP2).

The communication variable facilitates communication between:

- Newly created Function Diagrams in System Version 5.1 or later.
- A restored diagram (created upto System Version 5.0 SP2) and a newly created diagram in System Version 5.1 or later.
- Restored Function Diagrams.

In effect, diagram references consist of an input/output reference and **communication variable**.

Diagram references can be connected through Bulk Data Manager, see [Working with Diagram References \(Symbol Objects\) Using Bulk Data Manager](#).



In the sense of a structural programming language, diagram references and diagram variables refer to global or communication variables.

The global variable and the communication variable takes the name, description, data type, initial value and attributes of the diagram variable.



Two engineering templates named **BDM_DiagramRef_Var_Basic** and **BDM_DiagramRef_Var_Advanced** are used to modify diagram references and diagram variables represented as symbol objects.

Connect To Diagram Variable

User can interactively create (off-) diagram reference and connect to diagram variables by the following steps:

1. In Diagram view, insert a function component of type [Object Type Structure]Object Types/Functional Planning/Generic Function Components/Connectors/Off Diagram Reference/**Input (Output) Reference** or **Reduced Input (Output) Reference**.

2. The **Variable Properties** dialog appears, enter Name and Description of the diagram reference.
3. Click **OK** if the user wants to leave the diagram reference without having it connected to a diagram variable.
4. If the user wants to create a diagram variable and connect to it or if the user wants to connect to an existing diagram variable, click **Connect** instead of OK.
5. In the dialog **Connect to Diagram Variable** enter the name of the variable or select an existing variable from the list.
6. Click **OK**.



Double-click the Diagram Reference which does not have a variable connection to invoke the **Connect To Diagram Variable** dialog.



Double-click an **Output Diagram Reference** which is connected to a variable but has no reference to the input, to invoke the **Variable Properties** dialog.

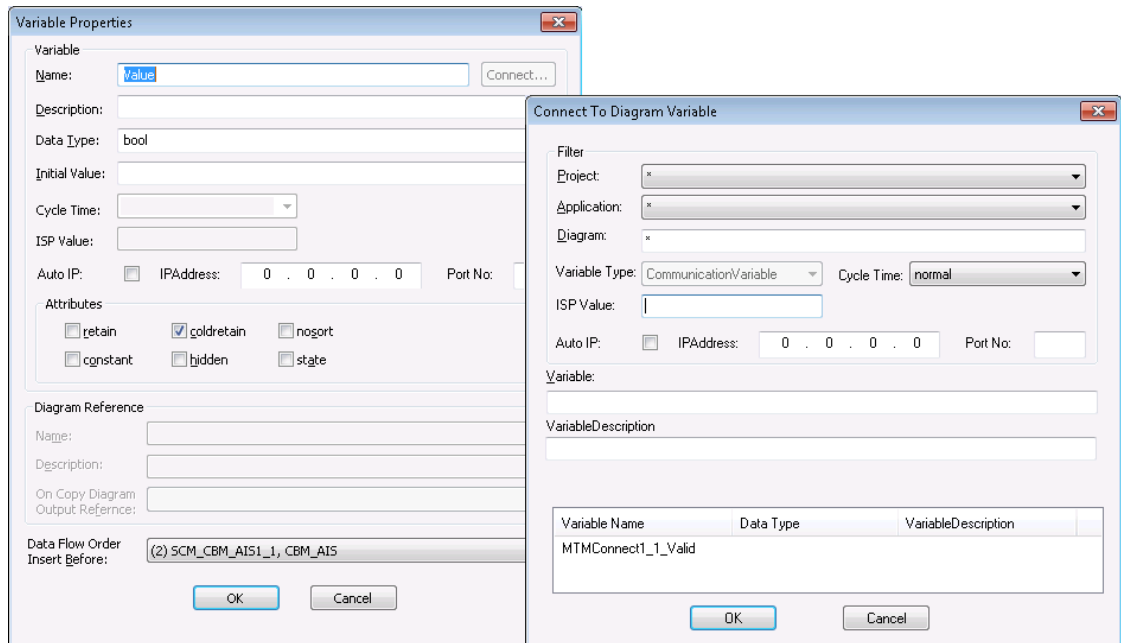


Figure 49. Connect to Diagram Variable

7. A new diagram variable gets created or an existing one gets inserted in Functional Structure.
8. The diagram reference is connected to that diagram variable. Cross References are automatically updated in the diagram as shown in [Figure 48](#). The datatype of the Valid bit present in a diagram reference used for cross communication is dword. When a proper communication is established, the value of this bit is 192.



If user drags and drops a port to the left border of a function diagram, a new reduced diagram reference gets automatically created with corresponding data type. The default name of the diagram reference equals <PortName>.

In addition, the user can connect to an existing diagram variable or create a new one. The default name of the diagram variable equals <DiagramName>_<PortName>.

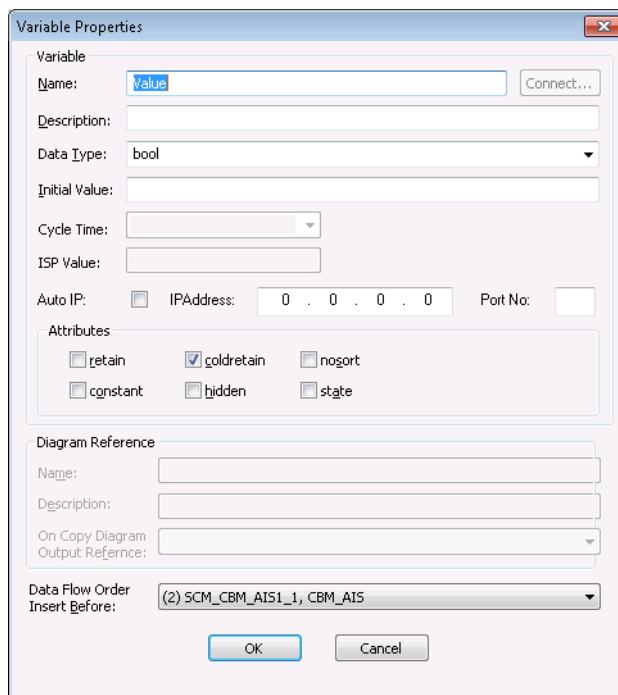
The same applies for reduced output references that get automatically created on drag and drop of a port to the right border of a diagram.

This functionality is not available in Function Diagrams containing SPL components. User is recommended to use the context menu **New Diagram Output Reference...** or **New Diagram Input Reference...** based on the requirement.

Connection between Restored Diagrams (created upto 800xA 5.0 SP2) and New Diagrams (created from 800xA 5.1 onwards)

Follow the steps to connect an output diagram reference of a restored diagram (created upto System Version 5.0 SP2) to an input diagram reference of a new diagram created in System Version 5.1 or later:

1. Reserve the required restored Function Diagram.
2. Right-click the required output port and select **New Diagram Output Reference...** from the context menu.
3. Click **Connect...** in the appeared **Variable Properties** dialog as in [Figure 50](#), to open the **Connect to Diagram Variable** dialog.



The Variable Properties dialog box is shown with the following fields and options:

- Variable**
 - Name: Value
 - Description:
 - Data Type: bool
 - Initial Value:
 - Cycle Time:
 - ISP Value:
 - Auto IP: ☐ IP Address: 0 . 0 . 0 . 0 Port No:
- Attributes**
 - ☐ retain ☒ goldretain ☐ nogort
 - ☐ constant ☐ hidden ☐ state
- Diagram Reference**
 - Name:
 - Description:
 - On Copy Diagram Output Reference:
- Data Flow Order**
 - Insert Before: (2) SCM_CBM_AIS1_1, CBM_AIS

Buttons: OK, Cancel

Figure 50. Variable Properties Dialog

- In the **Connect to Diagram Variable** dialog, select **Variable Type** as **Communication Variable** from the drop-down list as in [Figure 51](#).



In Function Diagrams restored to 800xA 5.1, the variables list includes both diagram variables and communication variables.

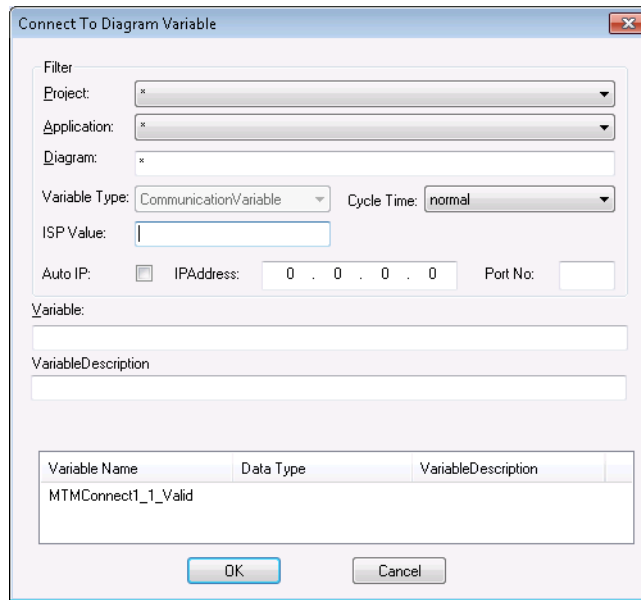


Figure 51. Connect To Diagram Variable dialog

5. Enter the required communication variable name in the **Variable** field of **Connect To Diagram Variable** dialog.
6. Click **OK**.
7. Click **OK** in the **Variable Properties** dialog.
8. Open the new diagram created in System Version 5.1 or later.
9. Right-click the required input port and select **New Diagram Input Reference...**
10. Click **Connect...** in the **Variable Properties** dialog window.
11. Select the communication variable created in [Step 5](#).
12. Provide the required ISP value in the **ISP Value** field.

The **ISP (Input Set as Predetermined)** is applicable only for communication input variables. This field defines the ISP value to be set for the **In** variable.

This value can only be set for basic data types. If an ISP value is not specified, the default value is the last good value. If no last good value exists (because of a communication failure), the init value is applied. For structured data types, the ISP values can be set only in the data type for each individual component. Hence, it is not possible to configure instance specific ISP values for structured data types.

13. Select the required cycle time from the **Cycle Time** drop-down option.

Communication cycle time is applicable for peer-to-peer communication. The possible values are fast, normal, slow, very fast, and very slow. The default value is normal.



In Function Diagrams restored to 800xA 5.1, the **ISP Value** and **Cycle Time** are unavailable to the user.

14. Clear the AutoIP check box, to provide the IP address manually together with the port number details.

IP address is applicable only for communication input variables. This field defines the IP address of the controller that contains the corresponding communication output variable (with the same name) in any of its applications. When no value for the IP address is entered, the editor automatically fills in the default value 'auto'. The IP address is resolved during compilation.

15. Click **OK** in the **Connect To Diagram Variable** dialog.
16. Click **OK** in the **Variable Properties** dialog.

To connect an output diagram reference of a new diagram created in System Version 5.1 or a later version to an input diagram reference of a restored diagram (created upto System Version 5.0 SP2), follow the above procedure with the following modifications:

- Create a **New Diagram Output Reference** at the output port of the required newly created diagram.
- Create a **New Diagram Input Reference** at the input port of the required restored Function Diagram.

Disconnect From Diagram Variable

User can interactively disconnect from a diagram variable by the following steps:

1. In Diagram view, right-click on a diagram reference symbol and choose 'Disconnect from Diagram Variable' from the context menu.
2. All references are disconnected, and corresponding function diagrams get automatically updated.



From 800xA 5.1 onwards, the output diagram reference symbol indicates if the connected variable is a communication variable (CV) or a global variable (GV).

Goto Reference

User can navigate from one diagram to another one by the following steps:

1. In the Diagram view, select a diagram reference symbol and choose [View > Goto Referenced Diagram](#), or right-click on the diagram reference symbol and choose **Goto Reference** from the context menu. Alternatively double-click on the diagram reference symbol or use the shortcut CTRL+T.
2. If one reference is available the referenced diagram is opened with the corresponding Diagram Reference selected.
3. If more than one reference is available (on output diagram references) the **Goto Reference** dialog appears.

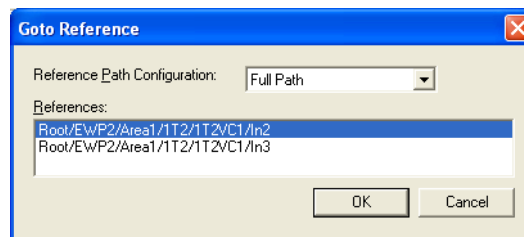


Figure 52. Dialog Goto Reference

4. Select a reference to open the corresponding diagram. The corresponding diagram reference symbol gets selected in the referenced diagram.

Number of Diagram References

By default, diagram reference symbols grow from top to bottom to display all references. As the number of references displayed is limited by the actual page size, user can define the maximum number that gets displayed in one column by the following steps:

1. In Diagram view, select a diagram reference symbol, right-click and choose **Number Of Diagram Reference...** from the context menu.
2. Define the maximum number of diagram references to be displayed in one

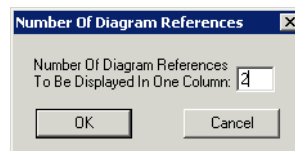


Figure 53. Dialog Number of Diagram References

column. If the actual number of references exceeds this number, the diagram reference symbol automatically adds additional columns and grows from left to right.



Diagram references are sorted according to diagram names, or, for equal diagram names according to diagram reference names.

Diagram Variables on Copy/Paste

Links on a single diagram define 1:1 or 1:N networks. Same can be considered for diagram variables which define 1:1 or 1:N networks between diagrams.

Keep connection to source:

To keep input or output diagram references stay connected to the diagram variable of the copy source, select the option **Keep connection to source** on **Variable Properties** dialog box. This allows you to quickly connect several input diagram references to one and the same variable of an output diagram reference.

Disconnect always:

To disconnect input or output diagram references from the diagram variable of the copy source, select the option **Disconnect always** on **Variable Properties** dialog box. Then the user has to connect / create diagram variables manually.

Make new unique connection:

Select **Make new unique connection** on **Variable Properties** dialog box to create new unique diagram variable.

Disconnect if source is outside copied range:

Select **Disconnect if source is outside copied range** on **Variable Properties** dialog box to disconnect the input diagram reference from the diagram variable if the variable was created outside of the copied diagram or copied diagram parts. This option is only available for input diagram references.



If an empty reference without Communication Variables or Diagram Variables is connected across the pages through page connector, then Function Diagram gives an error during Configuration Data Generation. To avoid this make sure that empty input reference is on the same page of connecting function block.

System Functions

Function Designer is an aspect system in the 800xA system and supports therefore all system functions like reservation, electronic signature, audit trail, user log-over, backup/restore, and export/import. Most functions are covered by the 800xA system, however *Function Designer* provides some extensions as described below.

Reservation

Function Designer requires reservation of an existing, non-blank Function Diagram for further modification work.

Not reserved Function Diagrams are read only, they show READ indication in the status bar.



The reservation of a created generic object with Function aspect in Functional Structure is released automatically, if the Function aspect is opened, modified and closed.

This happens only if the diagram is not allocated to an application.

Functions related to an application such as Allocate or Generate Configuration Data require reservation of the Control Application.

1. Click **File > Reserve Diagram** to get exclusive modify access to a Function Diagram. Click **File > Reserve Diagram** again to release the diagram. User can perform the same function by pressing the corresponding **Reserve / Release** toggle button of the Standard toolbar:



A new diagram is reserved by default. Closing the diagram releases the diagram automatically.

Press the CTRL+W keys to reserve an opened Function Diagram.

2. Click **File > Reserve Application** to get exclusive modify access to the application including all Function Diagrams.
Note that reserving the application could be done already in Control Builder Professional or in the Manage dialog of Plant Explorer.
3. Click **File > Release Reservation** to cancel reservation of diagram or application.



In the diagram view of a reserved Function Diagram, all the modifications will be saved automatically, while releasing the reservation through Plant Explorer using the **Release...** command in the context menu.

Graphical changes such as, moving the existing Function Blocks / Control Modules, Field Devices, CBM_Signal objects and their signal connections are not saved by this operation.

It is always recommended to release a Function Diagram from the Function Designer.

Audit Trail

Function Designer creates specific Audit Trail Events if the Audit Trail feature is enabled within the current 800xA system:

- On **File > Save**, an Audit Trail event **Save Information FuD** is written. An additional audit trail log message can be appended, see [Dialog Audit Trail Log Message](#).
- On any diagram state change, an audit trail event **InfoChangeDiagramState** is written, see [Diagram States and Transitions](#).

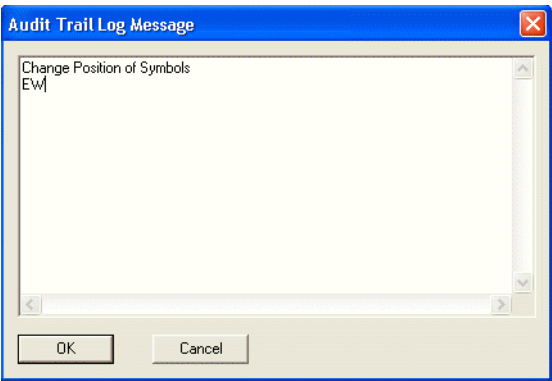


Figure 54. Dialog Audit Trail Log Message

Event Time	Object N	Component	Message Description	INFOTEXT
20.11.2003 10:44:59	ESC4	Audit Trail configuration	Audit Trail activated	
20.11.2003 10:46:56		Audit Trail FuD	InfoChangeDiagramState	The State has changed to Modified.
20.11.2003 10:47:39		Audit Trail FuD	InfoChangeDiagramState	The State has changed to Modified.
20.11.2003 10:59:19		Audit Trail FuD	Save Information FuD	Change Position of Symbols
20.11.2003 11:14:38		Audit Trail FuD	Save Information FuD	
20.11.2003 11:15:03		AesParameter	Aspect "CBM_SignalParameter" modified: PropName	
20.11.2003 11:15:04		Audit Trail FuD	InfoChangeDiagramState	The State has changed to Generated.
20.11.2003 11:17:36		No description exists.	CB System Variable DownloadWarning changed	
20.11.2003 11:17:36		No description exists.	CB System Variable DisableDownload changed	
20.11.2003 11:17:36		No description exists.	CB System Variable ReauthenticateDL changed	

Figure 55. Audit Trail Events Written by Function Designer

Permissions and user roles

Users of *Function Designer* who want to modify and configure diagrams and components need to hold the **Application Engineer** user role. An application engineer has good knowledge about working with the Aspect Object Framework, Engineering Workplace, Control Builder M, and FieldBus Builder.

Users of *Function Designer* who want to open or print diagrams and components can be a system engineer, an operator, or a maintenance employee.



It is recommended that the Operator user role should not have access to Control Builder M.

Authority

Function Designer supports the 800xA system's authority concept by checking

- permissions granted to a user and
- roles connected with this user.

This is done against the permissions required for actions in *Function Designer* and roles required for user interface availability.

To create, edit, or generate configuration data you need the application engineer role. In case you do not hold that role, or in case a **Security Definition Aspect** restricts configuration permissions for specific users, user groups, or locations, the corresponding Function Diagram and Component views are in **Read-only** mode. This is indicated by:

- A Read indicator in the status bar. Double click on that indicator to get more detailed information.
- A Read-Only indicator in the title of various dialogs, e.g. Aspect Properties dialog and Component Properties dialog.

User Log-Over

You can use the Log-Over function with *Function Designer*. After log-over the current Diagram View / Component View is closed, the next Diagram View / Component View you open is then shown according to the new user rights.

AC 800M Integration

Introduction and Requirements

AC 800M integration of *Function Designer* describes the workflow of functional planning, allocation of functions to an AC 800M controller within Control Structure and generation of configuration data for *Control Builder M*.

For function component allocation and controller configuration, the system extension *Function Designer for AC800M Connect* is needed. This extension is dependent on system extensions *Function Designer*, *AC800M Connect*, and *Signal Extension for AC800M Connect*. All tools have to be installed, all system extensions have to be loaded.

To load the system extension *Function Designer for AC800M Connect*, use the **Configuration Wizard** via **System Administration > (System Name) > System Extension Load** as displayed in [Figure 56](#):

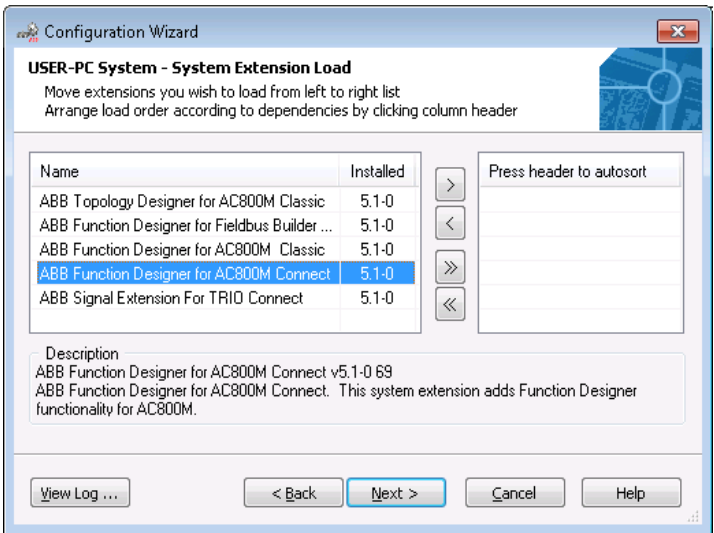


Figure 56. System Extension Function Designer for AC800M Connect

With Function Designer users can:

- Create diagrams including function components for Control Builder M. A newly created Function Diagram in Functional Structure represents a Diagram, a diagram in Object Type Structure represents a Control Module Type. A diagram includes:
 - Functions,
 - Function block types,
 - Control module types,
 - I/O signal types,
 - Sequences (IEC 61131 compatible or Sequence2D).
- Define data flow order.
- Define variables for your control code.
- Configure channels on I/O hardware units.
- Display on-line values in the diagram and in the Watch Window.
- Establish cross-communication automatically

Workflow Overview



The basic workflow to work with Function Diagrams and Sequences is described in *System 800xA Engineering, Engineering Studio Function Designer Getting Started (3BDS100968*)*. How to configure Sequences with Function Designer is described in this introductory manual only.

The following sub-sections give you detail information on functions you use when you work according to this workflow:

- [Building up Functional Structure and Function Diagrams](#)
- [Defining Data Flow Order](#)
- [Explicitly and Implicitly Defined Variables](#)
- [Building up Control Structure](#)
- [Allocating Function Diagrams](#)
- [Variable Creation](#)

- [Connection and Inversion Checks](#)
- [Instance Specific Initial Values](#)
- [Consistency](#)
- [Enhanced Naming](#)
- [Communication Between Applications using Cross Communication](#)
- [On-line Display](#)
- [Watch Window](#)

Building up Functional Structure and Function Diagrams

Function Designer is used for function oriented engineering of an automation system. For details, refer to subsection Create Process Functions in *System 800xA, System Planning (3BSE041389*)*.



Diagram Parameters are not supported in Functional Structure from system version 5.1 onwards.

Figure 57 shows the allocatable group aspect for a diagram inside Functional Structure:

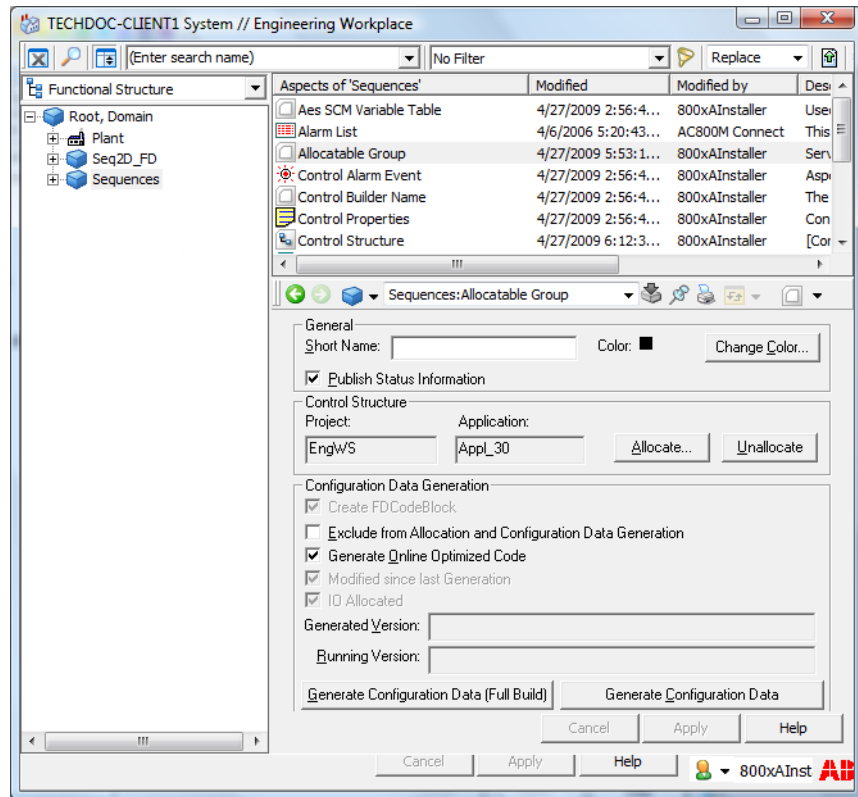



Figure 57. Allocatable Group Created in Functional Structure



If the Control Builder Application does not appear when user clicks **Allocate**: Check if Control Builder is in on-line mode. Switch Control Builder into off-line mode before invoking the **Allocate**.

Defining Data Flow Order


When instantiating a new component, a default "Data Flow Order" number is assigned. If the "Extended Name" dialog or the "Variable Properties" dialog pops up during instantiation this is used to select a certain entry in the new "Insert Before" combo box of these dialogs.

Click Auto Sort Order  icon from the **Quick Access** toolbar option to automatically sort the Data Flow Order for the various symbols of a Function Diagram. The shortcut key for this menu option is **Ctrl+Shift+D**.

The Data Flow Order can also be assigned through the **Data Flow Order** dialog. All components of an allocatable group can be moved up or down to get the required Data Flow Order.

The Data Flow Order number begins with 1.

Follow the steps to define the Data Flow Order:

1. Open the required function diagram.
2. Choose the menu command **Allocation > Define Data Flow Order** or click the  icon of the **Allocation** toolbar. The **Data Flow Order** dialog is opened.

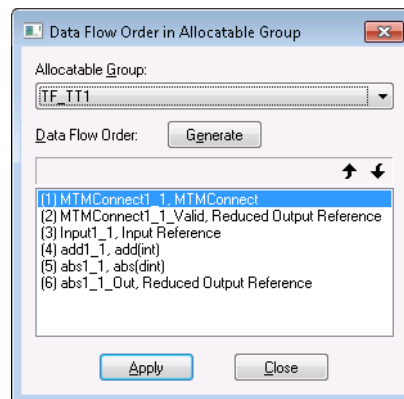




Figure 58. Data Flow Order Dialog

3. Select a function component from the **Data Flow Order** list and move it to the required location:
 - Click  to move up the selected function component.
 - Click  to move down the selected function component.
4. Click **Apply** to assign the new Data Flow Order to the respective function components. .



In Function Designer, always generate the data flow order after any changes in the configuration.



The **Generate** button can be used to sort the listed function components in the Data Flow Order in **Allocatable Group** window according to the default Data Flow Order. The default order is calculated according the rule “left up to right down”, as shown in [Figure 59](#). After generating the default Data Flow Order, click **Apply** to assign the Data Flow Order to the function components accordingly.



If a connection is made between output port of a block having higher dataflow order and the input port of another block having lower dataflow order, attribute of connected link variable is set to nosort. In order to ensure that the values are retained, it is advisable to make connections via local variables.

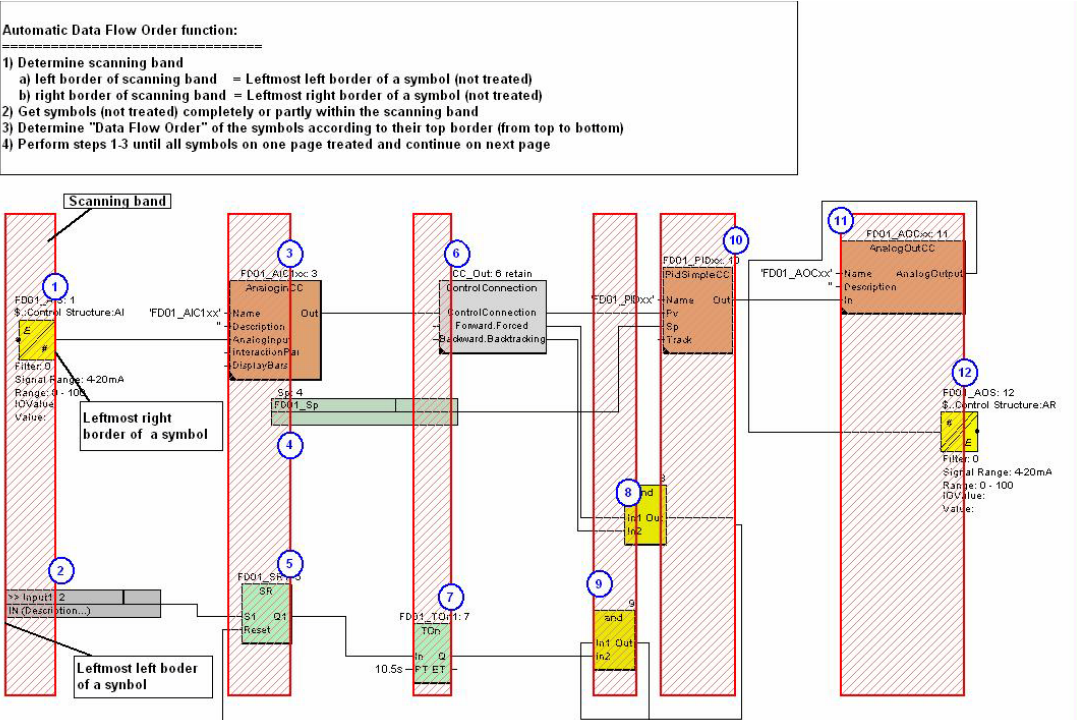


Figure 59. Algorithm for Calculation of Data Flow Order

Explicitly and Implicitly Defined Variables

Variables can be declared explicitly, or they get implicitly created whenever needed during configuration data generation. Explicitly declared variables are:

1. Diagram Variables which are further available as:
 - 1.Global variables
 - 2.Communication variables
2. Local variables, see [Figure 60](#).

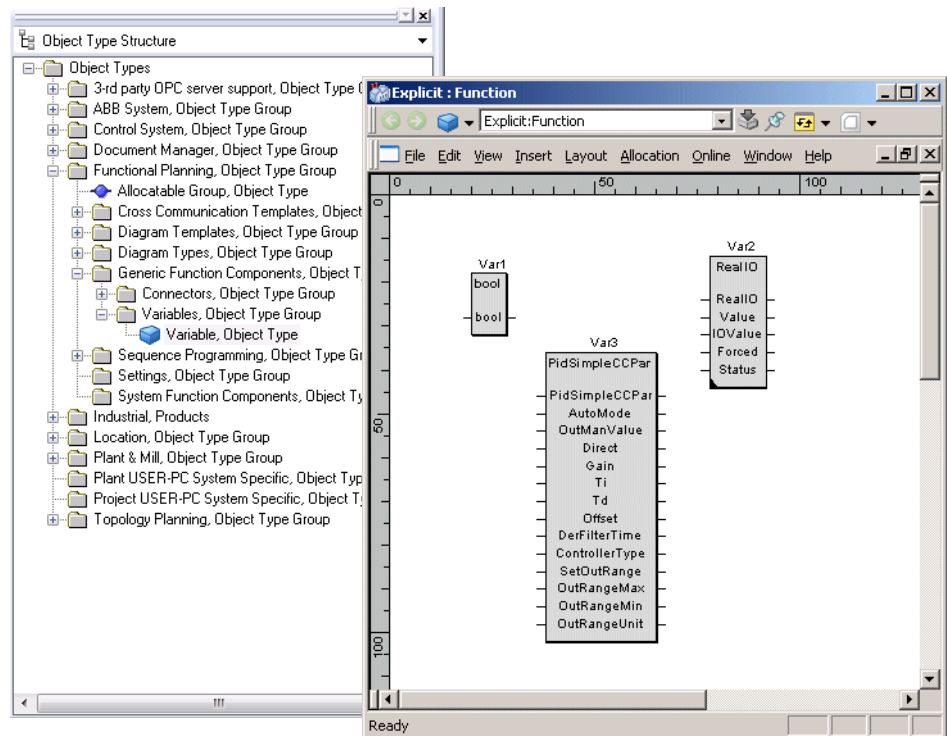


Figure 60. Explicitly Created Local Variables

The dialog **Variable Properties** in Figure 61 is used to define properties for diagram references/diagram variables, and local variables. You can set

- **Name** of variable/reference
- **Description** of variable/reference
- **Data type** of variable/reference. The combo box **Data Type** lists all system-, library- and application defined data types.
- **Attributes** of variable/reference.



Function Designer allows to use the state attribute for variables. For a variable with “state” attribute the following is valid, if the variable is required in two different code blocks:

- Read applies to the “old” value.
- Write applies to the current value.

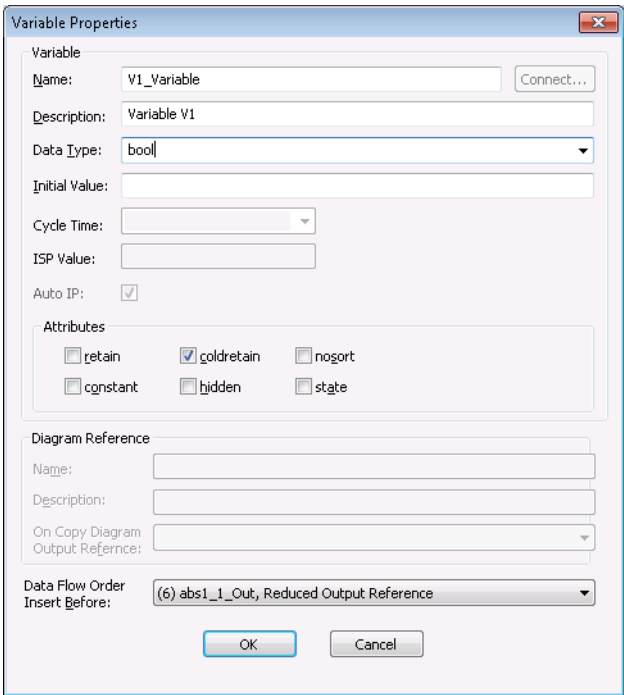


Figure 61. Dialog Variable Properties



If you modify data type or attributes of diagram references you get asked to update the data type or attributes for connected diagram variable as well. In case they differ, you will get corresponding warnings on **File > Check** and on open diagram.

Local variables of structured data type display ports to connect to sub-variables (use the [Show/Hide Ports](#) command to show or hide individual sub-variables). Initial values can be defined via the aspect property **Initval**.

Connection links implicitly create variables whenever needed. The variable name for such a link can be displayed by the dialog in [Figure 62](#). Select a link, right click and select 'Variable Name...' from the context menu.

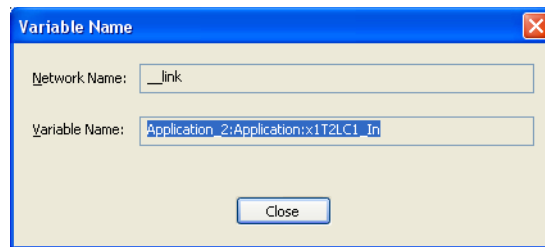


Figure 62. Dialog Variable Name

Conversion of Global Variables (Diagram Variables) into Communication Variables

Workflow to convert Global Variable (GV) used in Function Diagrams into Communication Variable (CV) using Bulk Data Manager (BDM).

1. In the **Function Structure** of **Engineering Workplace** right-click on **Function diagram** and select **Engineering templates** from **Advanced** menu.

2. From the Engineering Templates select **BDM_DiagramRef_Var_Basic.xlsx**.

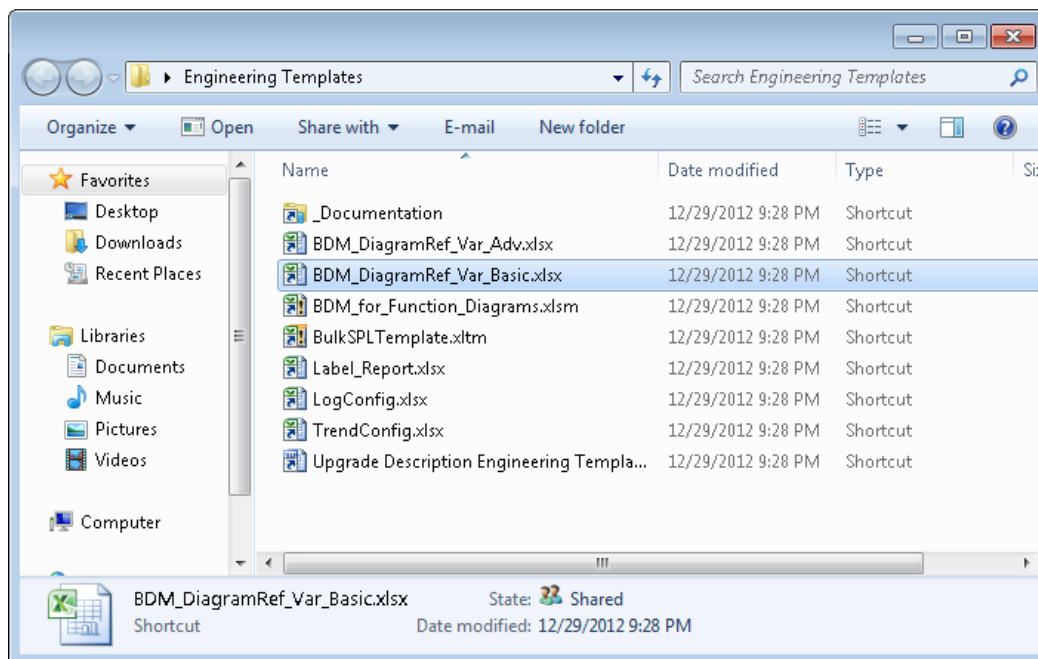


Figure 63. Engineering Templates

3. Drag the Function Diagram from **Engineering workplace** and drop it into the **Template**.
4. Create a copy of the **Template** and save it with a different name then close it.
5. Delete the Variable Names in earlier **Template** and perform **Save all Objects**.

Global Variables are disconnected from the references in function diagram.

Command	Object Identification	Ref Obj Path	Ref Type	Ref Desc	Variable Name	Var Desc
Filter:						
		[DIRECT][Functional StructureRoot/GVtoCV/add1_1]	Input		testCV1	
XML_Expanded		[DIRECT][Functional StructureRoot/GVtoCV/add1_1]	Input		testCV2	
XML_Expanded		[DIRECT][Functional StructureRoot/GVtoCV/add1_1]	Output		testCV3	
XML_Expanded		[DIRECT][Functional StructureRoot/GVtoCV/add1_1]	Input		testCV4	
XML_Expanded		[DIRECT][Functional StructureRoot/GVtoCV/add1_1]	Output		testCV5	

Figure 64. Variable Name String Screen

6. Open the copy of **Template** (saved in [Step 4](#)).
7. Under **CanCreateCV** column replace **False** by **True** and perform **Save all Objects**.

Command	Object Identification	Ref Obj Path	Ref Type	Variable Name	Var Desc	CanCreateCV
Filter:						
		[DIRECT][Functional StructureRoot/GVtoCV/add1_1]	Input	testCV1		FALSE
XML_Expanded		[DIRECT][Functional StructureRoot/GVtoCV/add1_1]	Input	testCV2		FALSE
XML_Expanded		[DIRECT][Functional StructureRoot/GVtoCV/add1_1]	Output	testCV3		FALSE
XML_Expanded		[DIRECT][Functional StructureRoot/GVtoCV/add1_1]	Input	testCV4		FALSE
XML_Expanded		[DIRECT][Functional StructureRoot/GVtoCV/add1_1]	Output	testCV5		FALSE

Figure 65. CanCreateCV Column

Communication Variables are connected to the diagram references.



The output references of Function Diagram may still show Global Variables but they use Communication Variables.

8. Generate the configuration data for the Function Diagram.

Building up Control Structure

The Control Structure gets automatically updated, see [Table 11](#).

Table 11. Changes in Control Structure

Change in Function Diagram or Functional Structure	Change in Control Structure
Instantiate/copy/delete function or procedure	none (Diagram gets updated on configuration data generation)
Instantiate/copy/delete function block or control module as symbol object	none (Diagram gets updated on configuration data generation)
Instantiate/copy function block or control module as aspect object	Function block or control module gets inserted below diagram / allocatable group (Diagram)
Delete function block or control module aspect object in diagram	Function block or control module gets deleted from diagram / allocatable group (Diagram)
Delete function block or control module aspect object in Engineering Workplace Advanced Menu Delete from All Structures	Function block or control module gets deleted from diagram / allocatable group (Diagram)
Delete function block or control module aspect object in Plant Explorer object browser	none. You need to manually remove the instance from the Control Structure.
Instantiate/copy/delete CBM IO signal	none (Diagram/application gets updated on configuration data generation)
Instantiate/copy/delete diagram parameter	none (Diagram gets updated on configuration data generation)
Instantiate/copy/delete diagram reference and connect to diagram variable	none (Diagram/application get updated on configuration data generation)

Table 11. Changes in Control Structure (Continued)

Change in Function Diagram or Functional Structure	Change in Control Structure
Instantiate/copy/delete local variable	none (Diagram gets updated on configuration data generation)
Copy/paste diagram objects	Creates a copy of diagram and put it into the same CBM application. (IO signals need to be reallocated to appropriate hardware units)
Creates a copy of Functional Structure with same parent	Copy/paste diagram objects (IO signals need to be reallocated to appropriate hardware units)
Create copies of related diagrams with same parents	Copy/paste application (IO signals need to be reallocated to appropriate hardware units)
Delete diagram object	Diagram object gets deleted from Control Structure and CBM (and vice versa)
Allocate diagram / allocatable group	Diagram gets moved from unallocated application to corresponding CB application
Generate configuration data	Nested (still unallocated) children get moved to corresponding CB application
Unallocate diagram / allocatable group	Diagram gets moved from CB application to unallocated application



In Engineering Workplace object browser (e.g. Functional Structure):

- Paste moves the cut object to the new parent.
- Cut, Delete, Paste no longer supported.
- Cut, Paste, Paste no longer supported.

Function Designer, Diagram View or Component View

- Cut still copies selected component(s) to the clipboard and deletes them as in previous system versions.
- Paste inserts the components from the clipboard.
- Cut, Paste, Paste still supported.
- Cut in diagram, paste in Plant Explorer object browser (Functional Structure) no longer supported.
- Cut in Plant Explorer object browser (Functional Structure), paste in diagram no longer supported.

Allocating Function Diagrams

After having defined the data flow order, user can allocate the function diagram logic to a specific application. The following requirements must be fulfilled:

- *Control Builder M* (CBM) has to run.
- The corresponding CBM project has to be opened.

Allocation of a Function Diagram

The default allocatable group contains the whole function diagram. The allocated diagram gets a Diagram in the Control Builder M.

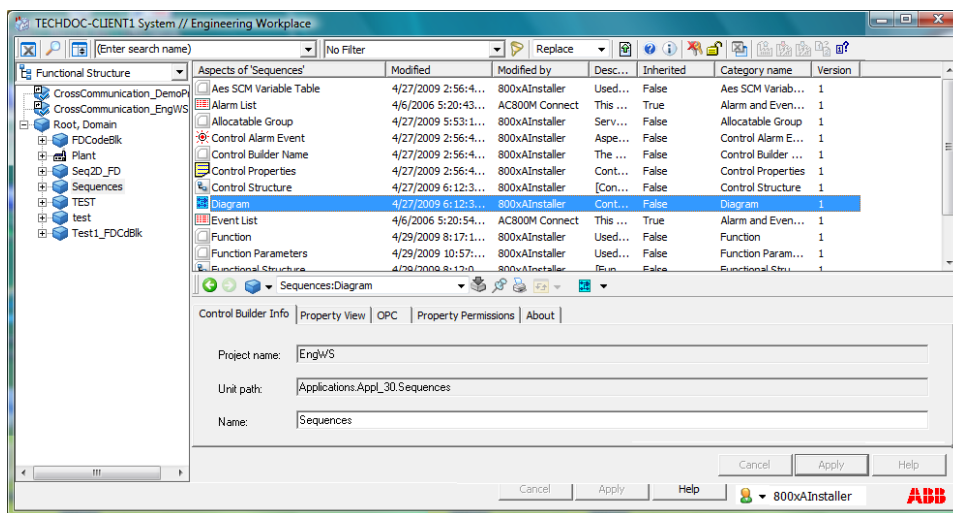


Figure 66. Function Designer Generated Diagram Aspect in Functional Structure

Unallocation

To unallocate an already allocated group, click **Allocate / Unallocate Diagram** from the Quick Access toolbar.

- Child diagrams, if any, are also unallocated.
- Control modules and function blocks are unallocated, i.e. moved from the CBM application to the unallocated application.

Allocating I/O Signals

After having allocated the logic of the function diagrams to the Control Structure, the I/O signals are to be allocated to hardware objects.

I/O parameters like Channel Settings and Scaling are configured in the **CBM_SignalParameter** aspect. User need not configure the Connection parameters Application, Variable, and Type. These parameters are automatically updated by Function Designer. Double-click a CBM_Signal to invoke the **Signal Parameters** dialog.

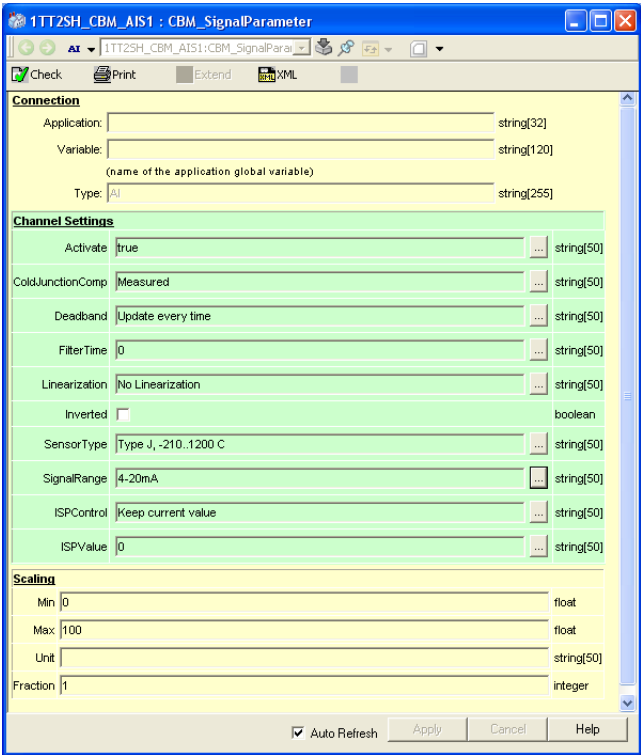


Figure 67. I/O SignalParameter Aspect

The I/O allocation type is configured in the **CBM_SignalInformation** aspect, see [Figure 68](#).

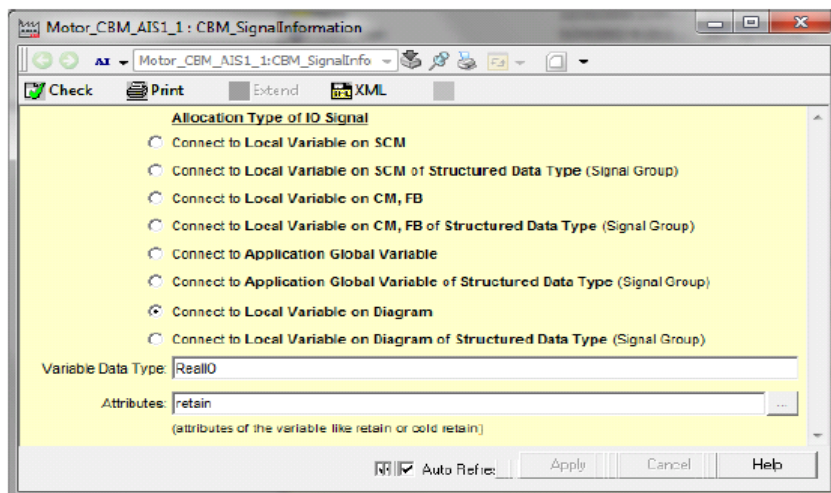


Figure 68. I/O SignalInformation Aspect

Typically, you will create and connect an I/O signal to a local variable. Examples and a description for the other options, e.g. create and connect to local variable, or signal groups, are described in detail in section IO Allocation of *System 800xA Engineering, Engineering Studio (3BDS011223*)*. An example how to configure a signal group is also given in *System 800xA Engineering, Engineering Studio Function Designer Getting Started (3BDS100968*)*, subsection Creating a Signal Group Object Type.

Allocation of I/O signals means insertion into the following Control Structure path (see also Figure 69):

Control Network\CBM_project\Controllers\Controller_n\Hardware\..., for example: **Control Network\CBM_project\Controllers\Controller_n\Hardware\011\1** for a CBM_AIS object below a AI810 I/O board on ModuleBus.

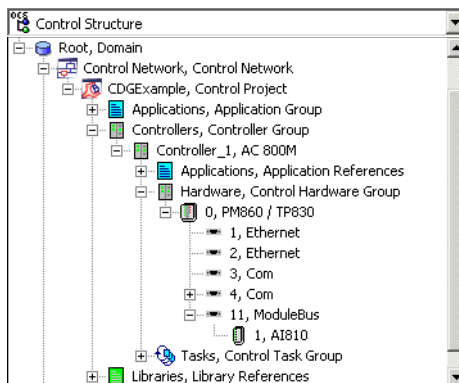


Figure 69. I/O Signal Allocation in Control Structure

The channel number is configured by the I/O signal's ChannelNumber aspect, see [Figure 70](#).

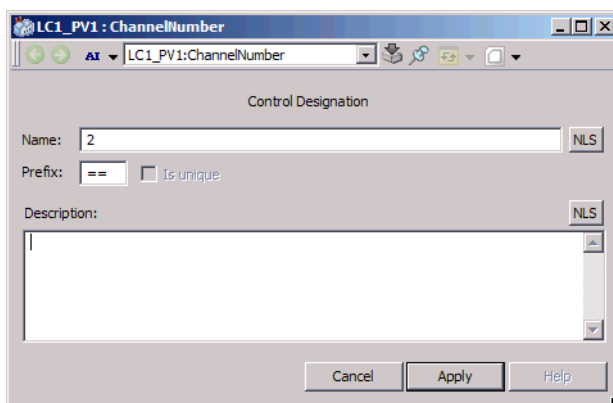



Figure 70. I/O Signal ChannelNumber Aspect

Preferably use the I/O Allocation tool, see *System 800xA Engineering, Engineering Studio* (3BDS011223*).

The I/O Allocation tool can be started in different contexts. To start the tool in the context of a Function Diagram:

1. Either click the IO Allocation  icon from the Quick Access menubar, or click **Allocation > IO Allocation** or right-click on white space of the Function Diagram and click **Advanced > IO Allocation**. The tool shows all CBM_Signals inserted in the diagram. If they are already allocated, the boards are shown to which the signals are allocated.
2. To allocate a signal and to transfer allocation information and signal parameters to Control Builder M:
 - a. Right-click the Boards object.
 - b. Click Insert Element.
 - c. In the dialog's Control Structure navigate to and select an appropriate I/O board hard unit, for example AI810 and click **Add**.
 - d. Click **Close** to close the dialog.
 - e. Select one or several signals in the right pane and drag it/them on to the boards object (or on a board object or on a specific channel of a board) in the left pane.
 - f. Click **File > Generate Configuration Data** in Function Designer or **Edit > Write Allocation into CBM** of the I/O Allocation tool to write results to CBM.



Generating Configuration Data in bulk for the Function Diagrams using Bulk Data Manager, requires more time.

If any Function Diagram is modified, it is advisable to generate Configuration Data for only that Function Diagram.

The graphical symbols for the CBM_Signal object types installed by the Function Designer for AC800M Connect system extension use text and label components with property references in order to display

- I/O signal name
- Designation (location) in Control Structure
- FilterTime (AIS, AOS)



- SignalRange (AIS, AOS)
- Min, Max, Unit (AIS, AOS)
- On-line Values.

Generate Configuration Data

After allocation of the diagrams / groups, user generates configuration data to transfer or update the Function Designer data into Control Builder M. In detail, configuration data generation creates/updates

- FD Code Blocks with parameters, variables, communication variables, function blocks, functions, procedures.
- Control modules and connections.
- Control properties.
- Application global and local variables.
- Hardware connections (due to IO Allocation).
- OPC properties.

Start configuration data generation by:

- Function diagram, **Generate Configuration Data (Full Build)**  icon or **Generate Configuration Data**  icon from the Quick Access toolbar.
- Function diagram, **File > Generate Configuration Data (Full Build)** or **File > Generate Configuration Data**
- Allocatable Group aspect, buttons **Generate Configuration Data (Full Build)** or **Generate Configuration Data**, see [Figure 57](#).
- Allocatable Group aspect verbs **Generate Configuration Data (Full Build)** or **Generate Configuration Data**. This is also useful on [Bulk Generate Configuration Data](#).

Generate Configuration Data (Full Build) updates all selected diagram(s)/group(s) and their children, independent of the diagram state, see [Instance Specific Initial Values](#).

Generate Configuration Data optimizes the build process and updates diagrams in modified (red) state only. Diagrams in already generated (yellow) or loaded (green) state are not changed.



Changes done in Function Designer after generating configuration data can be handled in two different ways:

- **File > Save** or **File > Save All** saves the changes within Functional Structure. The already generated Single Control Modules and Diagrams within the Control Structure are not updated.
- **File > Generate Configuration Data** saves the changes within Functional Structure and also updates the already generated Single Control Modules and Diagrams within the Control Structure.
- **File > Generate Configuration Data (Full Build)** saves the changes within Functional Structure, and (re)generates configuration data for an already allocated solution in Control Structure.

User can select **File > Save** or **File > Save All** to save the intermediate changes while working with Function Diagrams. Finally the Single Control Modules and Diagrams can be updated by selecting **File > Generate Configuration Data** once.

While working with SPL sub-diagrams, user needs to select **File > Save All** to save the intermediate changes in all SPL sub-elements (Steps, Transitions, N_Action) before generating configuration data. The Function Diagram can be updated by selecting **File > Generate Configuration Data** once.



Before performing the Generate Configuration Data for Function Diagrams, set **ConnectLibsOnGenerateConfigData** property to **False**, so that the library version of Function Diagrams are not replaced by latest version during the system update. After performing the Generation of Configuration Data for Function Diagrams, change **ConnectLibsOnGenerateConfigData** property to **True**.

Variable Creation

In *Control Builder M*, variables are used for the following kinds of connection:

- **Connections between Blocks inside an Allocatable Group**
- **Connections across Diagrams (horizontal cross references)**

- [Connections to I/O Channels](#)
- [Connections from I/O Channels](#)

The following table contains the descriptions of abbreviations used in the tables of this section:

Table 12. AC 800M Connect: Abbreviations

Action Abbreviation	Action Description
AccessVar	Create Access Variable
Appl: GlobVar	Create global variable on application (editor) level
CM	Control module
FB	Function block
FUN	Function
Parent SCM	SCM one level above (within nested Allocatable Groups)
SCM	Single control module (= Allocatable group)
SCM: Var	Creates variable on single control module (editor) level
SCM: Ext.Var	Creates external variable on single control module (editor) level
SCM: Par	Creates Parameter on single control module (editor) level
TmpVar	Creates direct logic connection (= temporary variable in ST Code)
Diagram:Comm Var	Create Communication variable on Diagram
Diagram:Var	Create a Variable on Diagram

Connections between Blocks inside an Allocatable Group

If there are connections between blocks (functions, function blocks, control modules) inside an allocatable group, the following variables will be created:

Table 13. Connections of Blocks Inside an Allocatable Group⁽¹⁾

Target ⇨ Source ⇩	FB	CM	FUN
FB	TmpVar	SCM/Diagram: Var	TmpVar

Table 13. Connections of Blocks Inside an Allocatable Group⁽¹⁾

Target ⇨ Source ⇩	FB	CM	FUN
CM	SCM/Diagram: Var	SCM/Diagram: Var	SCM/Diagram: Var
FUN	TmpVar	SCM/Diagram: Var	TmpVar

(1) The abbreviations are described in [Table 12](#)

The variable will be named according to the rule ‘__+link+[unique number]’ if the connections don’t have a label. If you have defined a labelled connection, the name of the label is taken for the variable name (Var instead a TmpVar in the table above).

Connections across Diagrams (horizontal cross references)

If there are cross reference connections across diagrams, the diagram reference objects are used and the following variables will be used:

Table 14. Diagram References⁽¹⁾

Target ⇨ (outside AG) Source ⇩	Diagram Reference
FB	Appl: GlobVar SCM: External Var/Comm Var Diagram: Comm Var
CM	Appl: GlobVar SCM: External Var/Comm Var Diagram: Comm Var
FUN	Appl: GlobVar SCM: External Var/Comm Var Diagram: Comm Var

(1) The abbreviations are described in [Table 12](#)

Table 15. Diagram References⁽¹⁾

Target ⇨ (outside AG) Source ⇩	FB	CM	FUN
Diagram Reference	Appl: GlobVar SCM: External Var / Comm Var Diagram: Comm Var	Appl: GlobVar SCM: External Var / Comm Var Diagram: Comm Var	Appl: GlobVar SCM: External Var / Comm Var Diagram: Comm Var

(1) The abbreviations are described in [Table 12](#)

Connections to I/O Channels

If there are connections to I/O channels, the following variables will be used:

Table 16. Connections to I/O Channels⁽¹⁾

Target ⇨ (outside AG) Source ⇩	I/O: Out
FB	Appl: GlobVar SCM: Ext.Var Diagram: Var
CM	Appl: GlobVar SCM: Ext.Var Diagram: Var
FUN	Appl: GlobVar SCM: Ext.Var Diagram: Var

(1) The abbreviations are described in [Table 12](#)

Connections from I/O Channels

If there are connections from I/O channels, the following variables will be used:

Table 17. Connections from I/O Channels⁽¹⁾

Target ⇨ (outside AG) Source ⇩	FB	CM	FUN
I/O: In	Appl: GlobVar SCM: Ext.Var Diagram: Var	Appl: GlobVar SCM: Ext.Var Diagram: Var	Appl: GlobVar SCM: Ext.Var Diagram: Var

(1) The abbreviations are described in [Table 12](#)

Annotation

For all two kinds of connections (Diagram References - I/Os) 'MOVE' blocks are inserted.

Connection and Inversion Checks

General

When two ports are connected to each other, or when one port is connected to an existing network, variable or to a constant, plausibility checks are executed. The same applies when a port is going to be inverted.

If the plausibility checks fail, the faulty connection or inversion is tolerated and visualized by a red line or red circle, respectively. Corresponding warnings/errors are written to the output window.

Direction Rules for Data Connections

The source attribute indicates a value. This value is transferred to the destination attribute.

Consequently in a data connection, a connection port referring to:

- an input (in) parameter can serve only as a destination.
- an output (out) parameter can serve only as a source.

- an input/output (in_out) parameter can serve only as a destination.



An input/output (in_out) port can not accept connections from input references (communication variables) and other output (out) ports.

Figure 71 provides the possible combinations of sources and destinations for data connections when applied to parameters. Figure 71 does not apply for function return values.

<div>Destination</div> <div>Source</div>	Communication Variables	Local Variables	CBM Signals	Control Modules	Function Blocks	System Functions
Control Modules	Yes	Yes	Yes	Yes	Yes	Yes
Function Blocks	Yes	Yes	Yes	Yes	Yes	Yes
System Functions	Yes	Yes	Yes	Yes	Yes	Yes
Communication Variables	Yes*	Yes*	Yes*	Yes	Yes	Yes
Local Variables	Yes*	Yes*	Yes*	Yes	Yes	Yes
CBM Signals	Yes*	Yes*	Yes*	Yes	Yes	Yes

* - An explicit Move block is inserted automatically while the source and destination are of the same data type.
An Explicit Splitter or Joiner block is inserted automatically while the source and destination have simple or structured datatype . (e.g) Bool - BoolIO etc.

Figure 71. Direction Rules for Data Connections

Checks for Connections

For port-to-port, port-to-network, port-to-variable, and port-to-constant connections the data types of the two objects have to be the same (exception: there is no data type check if one of the related objects has got the data type 'any').

Checks for Inversions

- Output ports and inout ports cannot not be inverted.

- Only ports with data type 'boolean' (or 'generic' or 'any') can be inverted.

Checking of constant literals

- Data Type REAL
A real constant is checked for the legal range
-3.402823466e+38 <= real <= 3.402823466e+38
- Data Type INT, DINT, UINT, WORD, DWORD
The constants are checked for legal range
-32768 <= int <= 32767
0 <= uint <= 65535 (= word)
- 2,147,483,648 <= dint <= 2,147,483,647
0 <= dword <= 4294967295

Numbers to bases other than 10 are represented in base 2, 8 or 16
(prefix 2#, 8# or 16#),

- Data Type BOOL
Valid constants for BOOL are 0, 1, FALSE, TRUE (case insensitive)
- Data Type TIME (= duration)
Time literals are written in the form <n>d<n>h<n>m<n>s<n>ms where
d = days, h = hours, m = minutes, s = seconds, ms = milliseconds,
<n> = a positive integer.
An optional prefix T# or TIME# can be used.
- Data Type DATE_AND_TIME
This type is presented in ISO_Format (YYYY-MM-DD-hh:mi:ss.ttt)
Example: 2003-05-23-14:15:30.025.
The minimum date is 1979-12-31-00:00:00.000.
An optional prefix DT# can be used.

Visualization

Faulty connections / inversions are indicated in red color. For faulty port-to-port connections the link indicating the connection is drawn red. For faulty variable or constant connections the connection string is drawn in red color. A red circle indicates faulty inversions, see [Figure 73](#).

Invocation of the Checks

The connection checks are executed when a port is going to be connected / inverted. If the checks fail a message box is popped up with an according error text. For connections leading to implicit casts a warning is issued in interactive mode only.

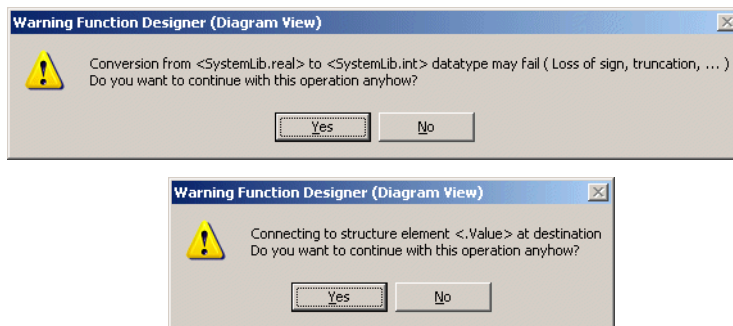


Figure 72. Warning Implicit Casts

When opening a diagram the checks are performed for all connections / inversions on the diagram. The error texts for failed checks are collected and displayed in the output window. A possibility to navigate to the faulty connection by double-clicking the error message in the output window is provided, see [Figure 73](#).

Additionally, the checks run

- On Function Designer menu item **File > Check**. It provides the same functionality as if the diagram is opened (see above). For diagrams with child diagrams this function has an option to check the whole tree of diagrams.
- Before allocating/generating configuration data for Control Builder M. If the checks fail with error, configuration data generation does not start.

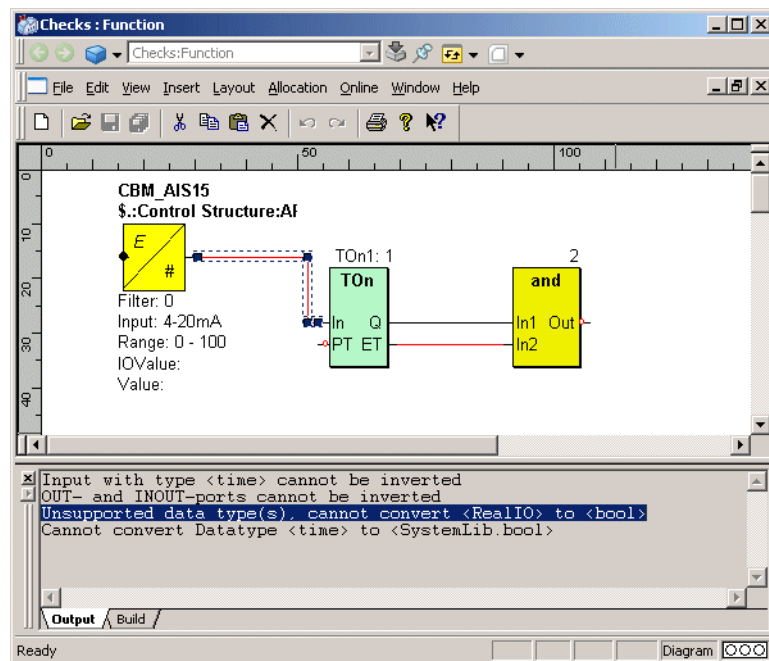


Figure 73. Diagram with Errors in Connections and Inversions

Splitter and Joiner

In systems prior to 800xA 5.1, connection mapping automatically creates and connects the different structured data types by instantiating the local variables. Dependent on the data type, one or more local variables get created and connected between the desired ports.

From 800xA 5.1 onwards the connection mapping between function blocks that do not have Data Flow Order is not supported. Splitter and Joiner blocks which has Data Flow Order, are introduced to establish the connection mapping between the structured and basic data types.

Constant values can not be assigned to local variables but can be assigned to joiner function blocks.

The splitter and joiner function blocks can be invoked by any of the following methods:

1. Select the required port of the control module / function block. Right-click and select **New Splitter** or **New Joiner** options from the available context menu as shown in Figure 74.
2. Select the splitter or joiner function block by navigating to the required object from the **Insert > Object** dialog, as shown in Figure 74.
3. Drag & drop the required splitter or joiner function block from Object Type Structure as shown in Figure 74.

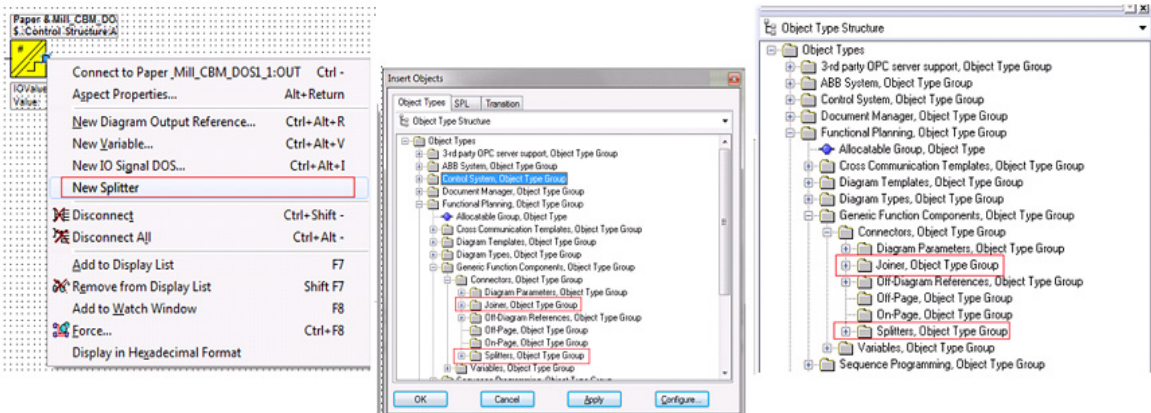


Figure 74. Splitter and Joiner Invocation Methods

A splitter or joiner is inserted based on the following conditions (refer to [Figure 75](#)):

- If the direction of the selected port is *input*, then a joiner is inserted.
- If the direction of the selected port is *output*, then a splitter is inserted.

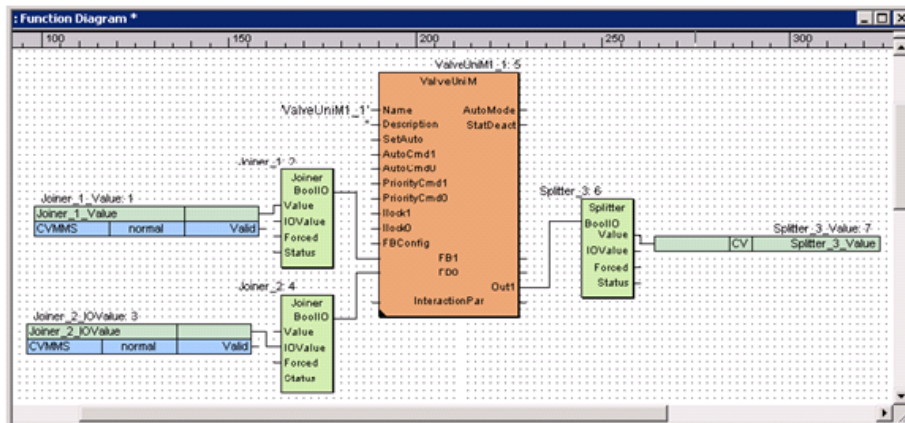


Figure 75. Function Diagram with Splitter and Joiner Connections

Limitations of Splitter and Joiner Blocks

Following are some of the limitations of *Splitter* and *Joiner* blocks:

1. Direct connection from a joiner block to a splitter block is not supported.
2. INOUT ports support splitter and joiner blocks through an implicit Move block.

Connection Mapping

Use the function **Connection Mapping** to connect ports of different structured data type. This function is implicitly invoked for connections that cannot be automatically cast. User can also explicitly invoke the Connection Mapping command from a link or a variable component. In both cases, user can select which substructures shall be interconnected. Necessary splitter or joiner and links get automatically created. For more details, refer to [Connection Mapping](#).

Instance Specific Initial Values

Function Blocks and Control Modules provide the possibility to set instance specific initial values for most of the unconnected ports. Then, independent of the object type, the instance will get an instance specific initial value for that unconnected port, if desired:

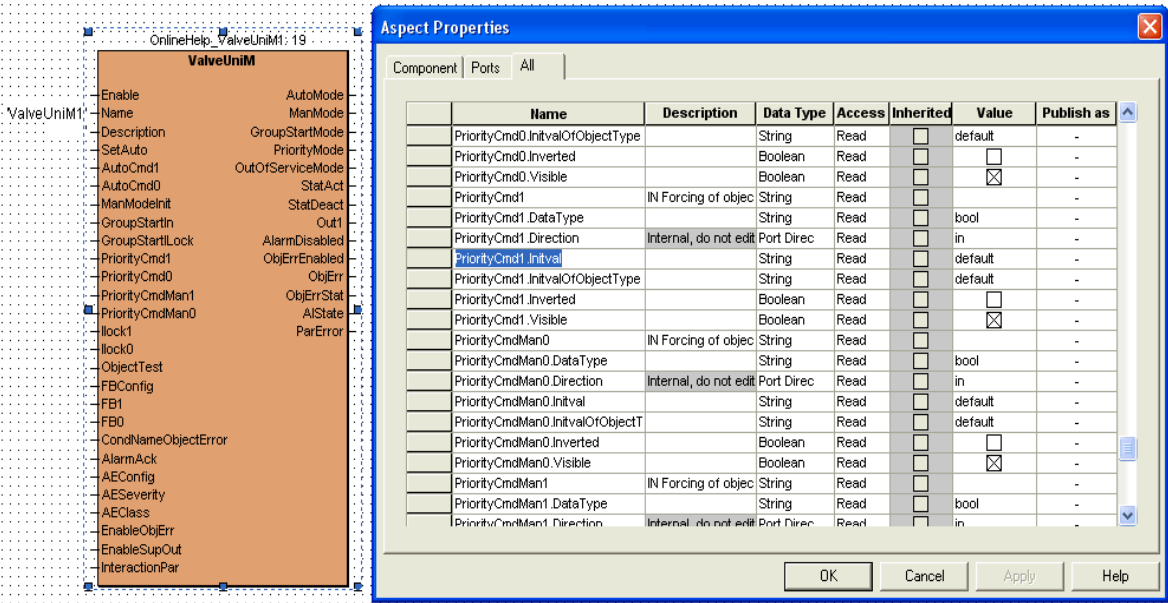


Figure 76. Aspect Properties of a PIDCC Control Module

Most of the ports of a function block or control module contain two aspect properties regarding initial values. For example see Figure 76: Port 'PriorityCmd1' has aspect properties

- PriorityCmd1.InitVal
- PriorityCmd1.InitValOfObjectType

If the PriorityCmd1.InitVal is set to 'true' for example, then - after configuration data generation - this value gets configured in the Control Properties aspect of that

control module. During download into the controller this port gets an instance specific initial value of value 'true'.

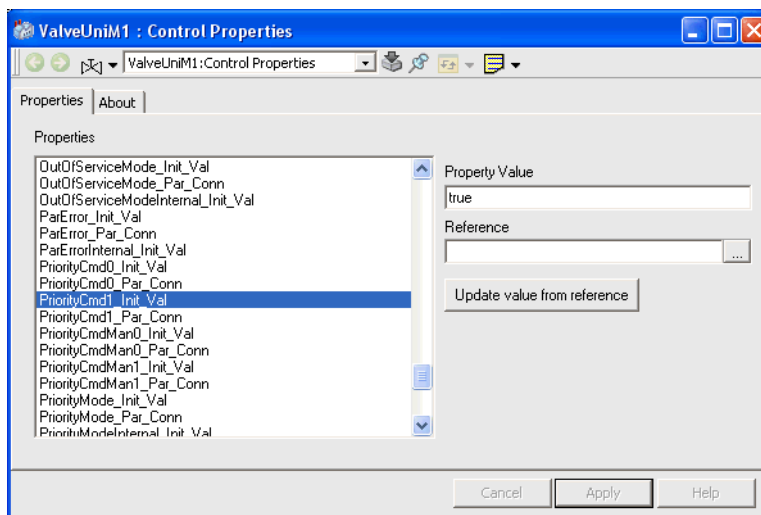


Figure 77. Control Properties After Configuration Data Generation for the Example in Figure 76 Above

Hints:

- For function blocks all unconnected ports can be configured, which provide an 'InitVal' aspect property.
- For control modules, only unconnected ports which have an 'InitValOfObjectType' of Value 'default' can get an instance specific initial value, because only these ports have a corresponding control property. If you specify 'InitVal' at other ports you will get a warning on configuration data generation.
- Modifications of the 'InitVal' property within the Control Properties aspect directly are currently not reflected back to Function Designer.

Consistency

General

To address both authority/access control and consistency of Function Designer generated configuration data, function diagrams follow a state model.

Consistency guaranties, that:

- a. Code blocks that have been generated by Function Designer cannot be modified in Control Builder (locking).
- b. Any modifications of objects that are generated by Function Designer (global variables, Object Names, ...) which are not locked inside Control Builder nor in Plant Explorer are correctly reflected in Function Designer.

In addition, diagram states answer the following questions:

- a. Which diagrams are not yet been allocated to any application?
- b. Which diagrams are actually loaded into the controller?
- c. Which diagrams have been altered since loading them into the controller?

Diagram States and Transitions

The following figure shows the different states of a diagram that is based on a writeable Function aspect in Functional Structure. Function Diagrams that are build in Object Type Structure and Function Diagrams with an inherited Function aspect follow different state models, which are described in [Consistency for Diagrams created from Control Module Types](#) and in [Consistency for Diagrams in Object Type Structure](#).

Audit Trail

Specific Audit-Trail messages are written when the diagram state changes.

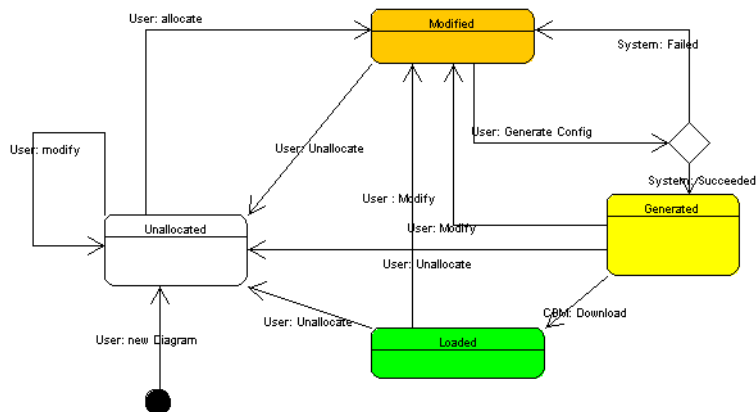


Figure 78. Diagram States

The diagram state is indicated in Function Designer's status bar. Diagram state is explained in *System 800xA Engineering, Engineering Studio Function Designer Getting Started (3BDS100968*)*, subsection Verifying Diagram States.

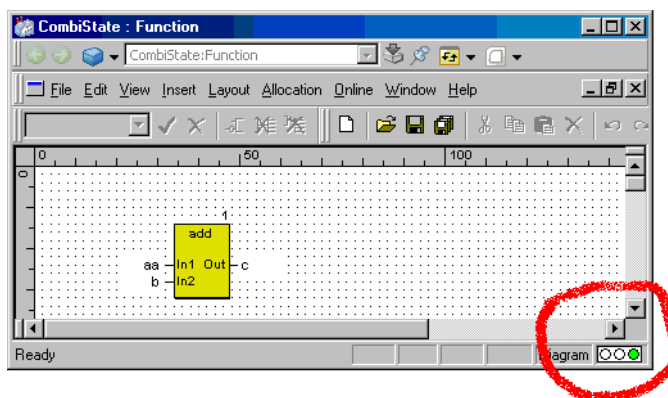


Figure 79. Diagram Status

The status of all diagrams in a (sub) tree of a structure can also be checked using the System Status Viewer.

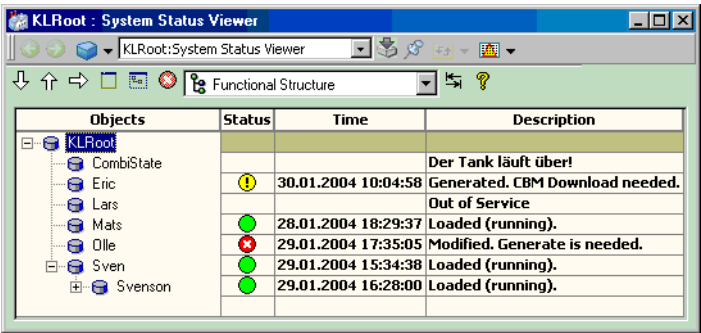


Figure 80. System Status Viewer

Consistency for Diagrams created from Control Module Types

Instances of **Control Modules** are based on a Control Module type. You cannot modify any content of the Control Module. Any changes of

- connections of the Control Module
- initial values of the Control Module

are regarded as being changes to the container of the Control Module. The following objects can act as containers for Control Modules:

- Control Modules
- Diagrams / Single Control Modules
- Applications

For Control Modules, Function Designer does not generate any code. It just writes the connections of the Control Module. So, there is no state ‘Generated’.

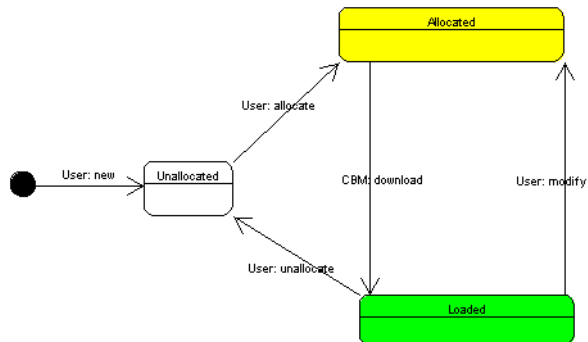


Figure 81. State Model for Diagrams Created from Diagram Types

Consistency for Diagrams in Object Type Structure

Function Diagrams that are created in Object Type structure to build Control Module types follow a different state model. They are never ‘Allocated’ (moved into Control Structure) nor ‘Loaded’.

Figure 82 shows the reduced state diagram for this type of Function Diagrams:

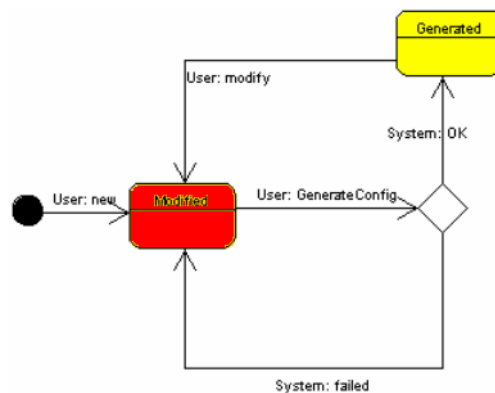


Figure 82. State Model for Diagram Types

Diagram Status Display

A diagram makes its status (Modified, Allocated...) available to the system via a number of properties. These properties are read (for example) by the System Status Viewer. The following table describes these properties.

Table 18. System Status Properties

Name	Description
S_STATUS	The status of the diagram as integer value.
S_DESCRIPTION	A textual description of the diagram status.
S_TIME	Time stamp of the diagram.
S_FUD_STATUS	Same as S_STATUS
S_FUD_DESCRIPTION	Same as S_DESCRIPTION
S_FUD_TIME	Same as S_TIME

For complex objects, the status of the Function Diagram may only represent a fraction of object's status. To get a combined status information, do the following:

1. Select the **Publish Status Information** check box in the Allocatable Group aspect.

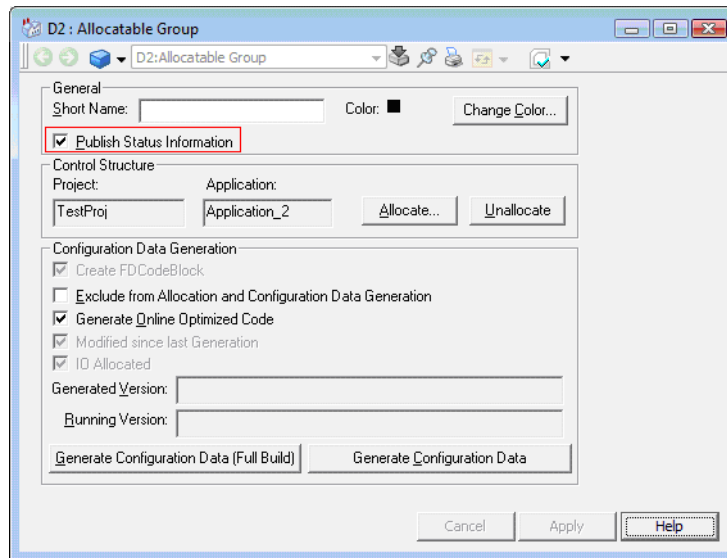


Figure 83. Allocatable Group

This prevents the Function Designer from publishing the status properties 'S_STATUS', 'S_DESCRIPTION' and 'S_TIME'.

- 2. Setup an aspect of the category “Property Translation Extended”.

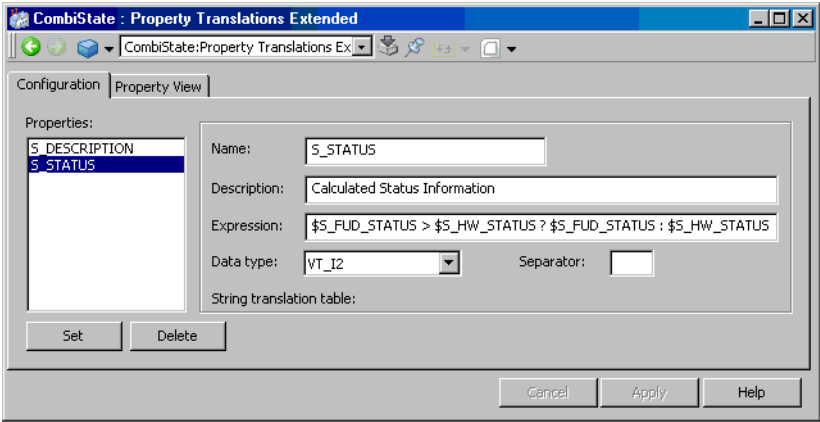


Figure 84. Property Translations

Cross References

Function Designer creates communication variables in Control Builder to implement the data exchange between diagrams. Function Designer informs the user about the usage of these variables in form of a cross-reference list that is attached to the variable as shown in the [Figure 85](#).

Input1: 4

Typical_VC1_Input1		
CVMMS	normal	Valid
Typical_VC1/Value(392684180)		
Typical_VC1/Output1(387611108)		

Figure 85. Graphical Representation of Cross References

Enhanced Naming

This section describes the enhanced name handling functions in the Engineering Workplace of System 800xA. They take care of names according to KKS naming rules and parent-child name concatenation.

Dialog New Component Name

When you insert a new component into a diagram, a dialog prompts for its name and description. How to insert new components, see [Inserting Designer Components](#).

The default name takes the diagram name as prefix, adds a '_', the object type name, and a unique running number 1,2...N. E.g. 1ABC_UniM1. If you have already existing components of the same type, the default name is the name of an existing object with a unique running number. The name pattern used in this dialog is defined in the *Function Settings* aspect, see [Function Settings](#).

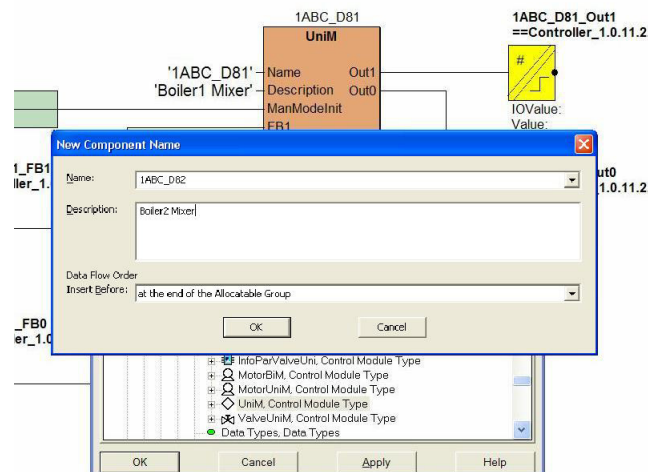


Figure 86. Dialog New Component Name

This dialog may be switched off for specific component types, see [Options](#) and [Naming](#).

Name Conversion

Each Aspect Object in the Plant Explorer has a Name aspect, which contains the corresponding aspect object name and description. For control objects there is an additional Control Builder Name (CBName) aspect, which gets synchronized with the name in Project Explorer, and therefore always follows IEC 61131-3 naming conventions, see also *System 800xA Control, AC 800M Configuration (3BSE035980*)*. If the control object is created in Plant Explorer and the name is in conflict with the IEC 61131-3 name restrictions, the aspect system automatically converts the name so that it follows these rules.

If a name is entered in Plant Explorer that does not follow IEC 61131-3 naming conventions, the Project Explorer name will be changed to fit the IEC 61131-3 naming conventions. If we, in Plant Explorer, name a tank object "12-1456%tank", the Control Builder conversion would look like "x12_1456_tank". The automatic name conversion function always adds an "x" to the beginning of the name and replaces all non-alphanumeric characters with underscore. Names of newly created Function Diagrams and its child objects should not exceed 32 characters. For restored diagrams, if the name is longer than 32 characters, it will be truncated.

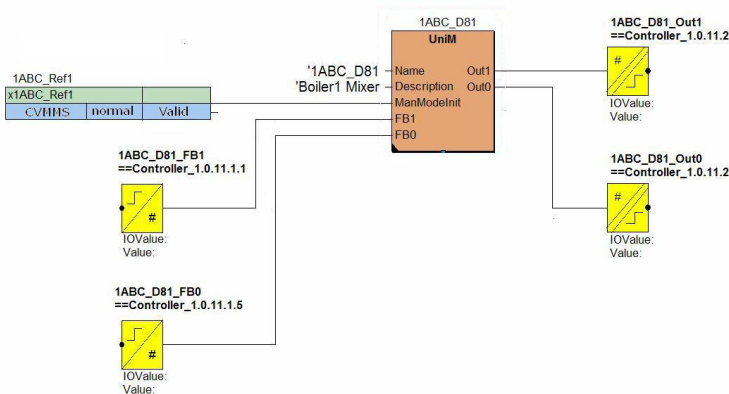


Figure 87. Diagram with Conflicts to IEC 61131-3 Names

Figure 87 shows an example for components with names, which are in conflict with the IEC 61131-3 name restrictions. The Control Module and the signals have an

additional CName aspect. The name of the diagram variable and other local variables have to fulfill IEC 61131-3 name restrictions. [Figure 88](#) and [Figure 89](#) show the Control Builder view for this example.

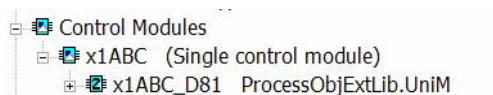


Figure 88. Objects in Control Builder

	Name	Data Type
1	x1ABC_D81_FB0	BoolIO
2	x1ABC_D81_FB1	BoolIO
3	x1ABC_D81_Out0	BoolIO
4	x1ABC_D81_Out1	BoolIO
5	x1ABC_Ref1	BoolIO

Figure 89. Generated Variables in Control Builder

Name Synchronization

For engineering efficiency, Function Designer synchronizes the *Name* aspect, *CName* aspect, *Name* parameter and *Description* parameter. The advantage of having synchronized names is that objects always have the same name in Plant Explorer and Project Explorer with the exception of IEC 61131-3 name restrictions, see [Name Conversion](#).

Many Function Blocks and Control Modules have a *Name* and *Description* parameter, which are important for alarm handling. You connect these parameters to a constant value or to some logic instead. If the parameters are connected to a constant, the values are synchronized to the *Name* aspect.

When an aspect object is renamed, the related *CName* aspect and the *Name* parameter is updated. When an object description is modified, the related *Description* parameter is updated.



Because of that name synchronization you do not need to run Name Uploader for Function Designer engineered diagrams. In effect, if you do so, name synchronization is switched off while Name Uploader is running.



Synchronization between Name aspect and parameters at Function Blocks/Control Modules can be disabled in Function Settings aspect.

This aspect is located in [Object Type Structure]ObjectTypes.Functional Planning.Settings.

In order to disable synchronization for the whole system, set property Naming.SynchronizeParameters to "false". Especially this function setting should be set to "false" in case of an upgrade from 800xA 3.1 or 800xA 4.1 to 800xA 5.1.

Aspect Verb Synchronize Names

Name aspect, Control Builder Name aspect, name parameters and description parameters are synchronized during engineering. If you right-click on the Function aspect of a diagram, you may trigger the name synchronization for the diagram and the related sub-tree manually.

Rename Substructure

When you rename a diagram, the names of the children are renamed too, if these start with the same prefix. In the following example the diagram 1ABC is copied with Bulk Data Manager to 1ABD and 1ABE, see [Figure 90](#) and [Figure 91](#).

	A	B	C	D
1	Command	Object Identification	Source Object	
2		[DIRECT][Functional Structure]Root.Naming.1ABD	[DIRECT][Functional Structure]Root.Naming.1ABC	
3		[DIRECT][Functional Structure]Root.Naming.1ABE	[DIRECT][Functional Structure]Root.Naming.1ABC	
4				
5				

Figure 90. Excel Sheet for Diagram Copy

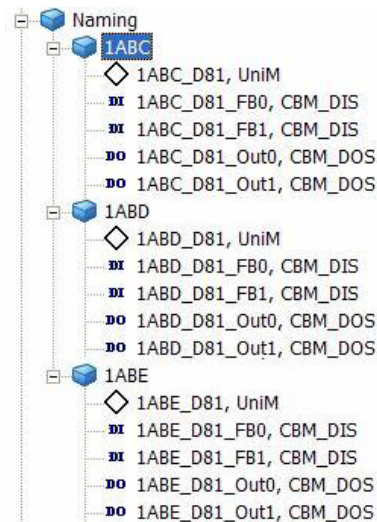


Figure 91. Result of Copy Diagram with BDM in Plant Explorer

When you copy the diagram 1ABC in Plant Explorer, you get a result according to Figure 92. The copied objects have the same object names as in the source.

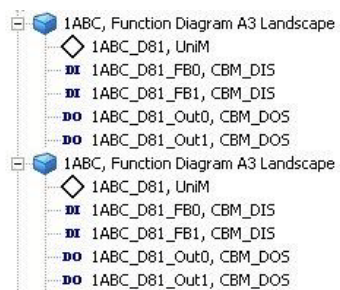


Figure 92. Copy Diagram in Plant Explorer

However, the Control Builder name of the corresponding SCM can be renamed (extended by a number), because of name collisions in the Unallocated_Inst_App

application. To rename the whole substructure you can use the object item **Advanced > Rename Substructure**.



It is recommended not to use **Advanced > Rename Substructure** when **Naming** is set to **None** in the **Function Settings** aspect of the **Settings** object.

Rename Function Diagrams

The following workflow is used frequently with Function Designer:

1. You prepare a Function Diagram in Function Structure.
2. You copy this diagram with Bulk Data Manager in Functional Structure. The copied diagram gets a new name.

Function Designer supports automatic renaming of the components within the diagram according to two concepts. In addition configurable naming rules are supported.

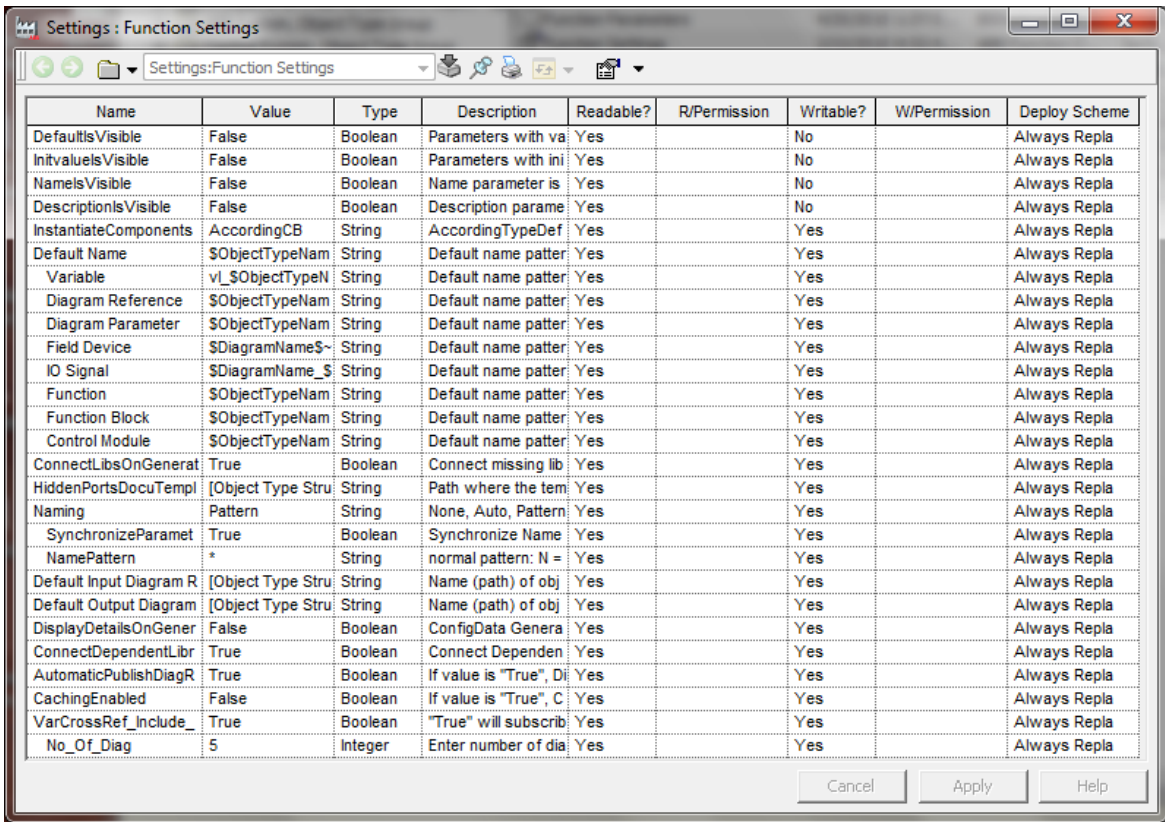
Two naming concepts are supported:

- Naming with name pattern.
- Automatic naming.

The actual used concept is determined by a Property “Naming” stored in Aspect “Function Settings” of Object “Settings” ([Object Type Structure]Functional Planning). It is possible to switch between both concepts or to switch off automatic renaming during copy/paste completely.

- The property Naming may have the following values:
 - None - Do not rename on copy/paste.
 - Auto – use automatic naming.
 - Pattern – use name pattern for renaming.

The name pattern is defined in Sub-Property “NamePattern”.



Name	Value	Type	Description	Readable?	R/Permission	Writable?	W/Permission	Deploy Scheme
DefaultsVisible	False	Boolean	Parameters with va	Yes		No		Always Repla
InitValuesVisible	False	Boolean	Parameters with ini	Yes		No		Always Repla
NamesVisible	False	Boolean	Name parameter is	Yes		No		Always Repla
DescriptionsVisible	False	Boolean	Description parame	Yes		No		Always Repla
InstantiateComponents	AccordingCB	String	AccordingTypeDef	Yes		Yes		Always Repla
Default Name	\$ObjectTypeNam	String	Default name patter	Yes		Yes		Always Repla
Variable	vl_\$ObjectTypeN	String	Default name patter	Yes		Yes		Always Repla
Diagram Reference	\$ObjectTypeNam	String	Default name patter	Yes		Yes		Always Repla
Diagram Parameter	\$ObjectTypeNam	String	Default name patter	Yes		Yes		Always Repla
Field Device	\$DiagramName\$~	String	Default name patter	Yes		Yes		Always Repla
IO Signal	\$DiagramName_\$	String	Default name patter	Yes		Yes		Always Repla
Function	\$ObjectTypeNam	String	Default name patter	Yes		Yes		Always Repla
Function Block	\$ObjectTypeNam	String	Default name patter	Yes		Yes		Always Repla
Control Module	\$ObjectTypeNam	String	Default name patter	Yes		Yes		Always Repla
ConnectLibsOnGenerat	True	Boolean	Connect missing lib	Yes		Yes		Always Repla
HiddenPortsDocuTempl	[Object Type Stru	String	Path where the tem	Yes		Yes		Always Repla
Naming	Pattern	String	None, Auto, Pattern	Yes		Yes		Always Repla
SynchronizeParamet	True	Boolean	Synchronize Name	Yes		Yes		Always Repla
NamePattern	*	String	normal pattern: N =	Yes		Yes		Always Repla
Default Input Diagram R	[Object Type Stru	String	Name (path) of obj	Yes		Yes		Always Repla
Default Output Diagram	[Object Type Stru	String	Name (path) of obj	Yes		Yes		Always Repla
DisplayDetailsOnGener	False	Boolean	ConfigData Genera	Yes		Yes		Always Repla
ConnectDependentLibr	True	Boolean	Connect Depend	Yes		Yes		Always Repla
AutomaticPublishDiagR	True	Boolean	If value is "True", D	Yes		Yes		Always Repla
CachingEnabled	False	Boolean	If value is "True", C	Yes		Yes		Always Repla
VarCrossRef_Include_	True	Boolean	"True" will subscrib	Yes		Yes		Always Repla
No_Of_Diag	5	Integer	Enter number of dia	Yes		Yes		Always Repla

Figure 93. Global Definition of Naming and Name Pattern

Name Pattern

Name Pattern Definitions

There are two different name pattern defined:

- 1. Simple name pattern
- 2. Standard name pattern

Either a simple name pattern or alternatively a standard name pattern has to be used.

1. Simple Name Pattern Definition

A Simple Name Pattern is defined like this:

- ? : any single character
- * : one or more characters

Samples:

Diagram Name	Name Pattern	Comment
PRC1111_FD	*???	Replaces 'PRC1111' in child names
PRC1111_FD	???*???	Replaces '1111' in child names

If the substring is found more than once in child names then only the first substring will be replaced.



Default Function Setting for Naming = “Pattern”, and NamePattern = “*”.



To improve the engineering efficiency, Function Designer deletes the _FD and _FUD indexes automatically from the proposed name when creating new objects or assigning diagram variables. This can be configured in Function Settings Aspect.

2. Standard Name Pattern Definition

A Standard Name Pattern is defined like this:

- N: a digit 0-9
 - L: any character (not a digit)
 - (): parts included in parenthesis are optional
 - \x: x is a specific fixed character, escaped by \
- \x can be used to define a fixed character within the pattern.

Samples:

LLLNNNN	3 characters followed by 4 digits
NNNLLNN(N)	3 digits, then 2 characters followed by 3 or 4 digits
LL(L)NN(N)(N)	2 or 3 characters followed by 2,3 or 4 digits
LL(L)NN(NN)	2 or 3 characters followed by 2 or 4 digits
LLLNNNN_ LLLNNNNN	3 characters followed by 4 digits and the character _ 3 characters followed by 4 digits and the character N

Renaming with Name Pattern

During copy Function Designer checks the name of the source diagram and the name of the target diagram. If both names match a name pattern, the differences within the name pattern are recognized. It is not necessary, that a name pattern matches the complete name. A name pattern defines only a substring within a name, which is subject to be changed during copy. The copied components of the diagram are renamed based on the name changes done for the parent diagram within the name pattern.

A common name pattern should be defined as long as possible. Replacements are more flexible and reliable with longer name patterns.

Characters outside the name pattern are not modified at all. Names of components, which do not fit to the pattern, are not modified.

Samples

Name Pattern LLLNNNN	
Source	Result
PRC1111_FD	PRC1121_FD
PRC1111	PRC1121
PRC1111_IN1	PRC1121_IN1
PRC1111_STO	PRC1121_STO
Value	Value
Alarm	Alarm

Name Pattern NNNLLNN(N)	
Source	Result
123LT2546_Fud	123LT 455 _Fud
123LT2546	123LT 455
123LT2546_1	123LT 455 _1
123LT2546_Intl	123LT 455 _Intl
AlcBlk	AlcBlk
MV_LT_H1_	MV_LT_H1_
MV_GT_H1_	MV_GT_H1_

Name Pattern NNLLNN	
Source	Result
51CD05	52CD07
51CD05_EinR	52CD07 _EinR
51CD05_GrpR	52CD07 _GrpR
51CD05X	52CD07X
51CD05_	52CD07 _

Name Pattern LLL(L)NN(N)(N)	
Source	Result
PRC1111_FD	XYC122 _FD
PRC1111	XYC122
PVCD1111_STO	XYCD122 _STO
PRXY1111_STO	XYXY122 _STO
Inc	Inc
Dec	Dec

Name Pattern LLL(L)NN(N)(N)	
Source	Result
PRC1111_FD	PR D 1111_FD
PRC1111	PR D 1111
PVCD1111_STO	PV D D1111_STO
PRXY1111_STO	PR D Y1111_STO
Inc	Inc
Dec	Dec

Name Pattern LLL(L)NN(N)(N)	
Source	Result
PRCD1111_FD	PRA A 1111_FD
PRC1111	PRA A 1111
PVCD1111_STO	PVA A 1111_STO
PRXY1111_STO	PRA A 1111_STO
Inc	Inc
Dec	Dec

PVCD1111_STO was renamed to PVA1111_STO, because 3rd character was modified from C to A and 4. optional character was removed.

Name Pattern LLL(L)NN(N)(N)	
Source	Result
PRC1111_FD	PRC D 11_FD
PRC1111	PRC D 11
PVCD1111_STO	PVC D D11_STO
PRXY1111_STO	PRX D Y11_STO
Inc	Inc
Dec	Dec

Optional character D was inserted.

Renaming with Name Pattern = *

This specific Name Pattern = * setting handles renaming in two different ways.

1. Splitting source text strings into substring tokens and rename them.
2. Splitting digits present in the source string into separate characters and rename them.

For example PRC1111_FD CONTROL will be divided into substring tokens such as PRC, 1111, _, RD, single space, and CONTROL.

Samples for Renaming with Name Pattern = * setting

Source	Target Name	To-do	Result
PRC33_FD PRC33 PRC33_STO	PRC 34 _FD	Replace substring token 33 with 34	PRC 34 _FD PRC 34 PRC 34 _STO
Source	Target Name	To-do	Result
PRC33_FD MV_PRC33_H1 MV_PRC33_H2	PRC 314 _FD	Replace substring token 33 with 314	PRC 314 _FD MV_PRC 314 _H1 MV_PRC 314 _H2
Source	Target Name	To-do	Result
PRC33_FD_HH MV_PRC33_HH MV_PRC33	PRC 314 _FD_HH_H1	Replace substring token 33 with 314 and add _H1 after HH	PRC 314 _FD_HH_H1 MV_PRC 314 _HH_H1 MV_PRC 314
Source	Target Name	To-do	Result
PRC33_FD PRC33_HH ARC33	HH _PRC 314 _FD	Insert HH_ before PRC and replace 33 with 314	HH _PRC 314 _FD HH _PRC 314 _HH ARC 314

Source	Target Name	To-do	Result
PRC33_FD PRC33_HH1 PRC33_HH2 ARC33_HH1 PRC35_HH1	PR_HH33_FD	Replace PRC with PR and insert _HH between PRC and 33	PR_HH33_FD PR_HH33_HH1 PR_HH33_HH2 ARC33_HH1 PR35_HH1
Source	Target Name	To-do	Result
PRC30_FD PRC31_FD PRC32_HH	PRC40_FD	Replace 30 with 40 or perform character replacement for digits. i.e. replace 3 with 4	PRC 40 _FD PRC 41 _FD PRC 42 _HH

Definition of Name Patterns

A name pattern can be defined in Functional Settings aspect in Object Type Structure, see [Figure 93](#). This name pattern is the global default for all Function diagrams.

Alternatively a name pattern can be defined for specific Function diagrams. In this case an aspect property “NamePattern” can be added to the diagram aspect properties. To do this right mouse click on the background of the diagram and select “Aspect Properties...”. In the upcoming dialog select tab “All”, right mouse click again and select “Add Property”. Add the property according to [Figure 94](#).

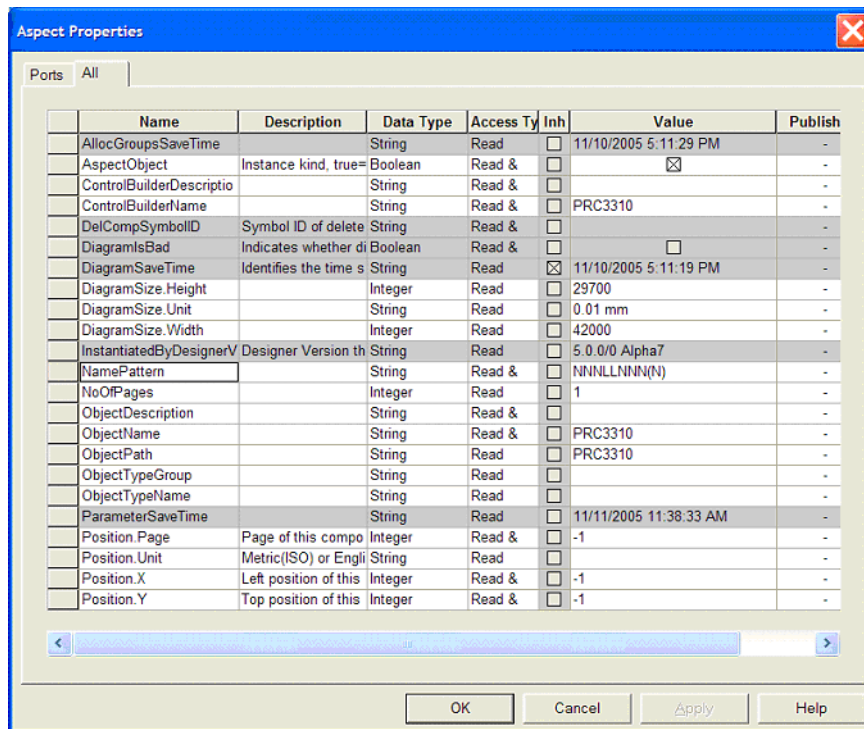


Figure 94. Name Pattern as Aspect Property

Search Strategy for Name Pattern

A name pattern is searched in the following order:

- Function Designer searches first for aspect property "NamePattern" within the copied Function Diagram. If there is one, the pattern is defined locally for the specific diagram. The aspect property will be copied/imported/exported together with the diagram.
- If aspect property is not found, Function Designer searches in Functional Settings aspect in Object Type Structure. In this case the pattern is defined globally for all Function Diagrams, if not overwritten by an aspect property.

- If no name pattern is found or if the name pattern does not fit to the old and the new name, the automatic naming algorithm is used.

The Pattern naming algorithm can be switched off in Functional Settings aspect, see [Figure 93](#).

Automatic Naming

A project may define a name pattern as described in [Name Pattern](#). If the Naming property in Functional Settings aspect is set to Auto (or a name pattern is not defined), see [Figure 93](#), Function Designer does the renaming based on the differences between the source and target name.

For that reason the names are split into tokens identifying unmodified/modified substrings and numbers. The result is a to-do list to be applied to component names.

This to-do list may contain several of the following commands:

- Replace substring by other substring
- Replace number by other number
- Delete substring
- Insert substring
- Add prefix
- Add postfix

Sometimes the position within a component name cannot be located precisely, especially if substrings are very short. In this case the previous or next character are checked in addition in order to find the correct position.

Auto(matic) naming algorithm is set as the default naming algorithm. It may be switched off in Functional Settings aspect, see [Figure 93](#).

Samples for Automatic Renaming

Source	Target Name	To-do	Result
PRC1111_FD PRC1111 PRC1111_STO PRC1111_	PRC1_FD	Replace 1111 by 1	PRC1_FD PRC1 PRC1_STO PRC1_
Source	Target Name	To-do	Result
PRC1111_FD PRC1111 PRC1111_STO PRC1111_FD_INC1 PRC1111_FD	PRC1111	Remove _FD	PRC1111 PRC1111 PRC1111_STO PRC1111_INC1 PRC1111
Source	Target Name	To-do	Result
AIS01_FUD AIS01 AIS01_Intl AIS01_MV AlcBlk MV_GT_L1_ MV_LT_H1_ Value	123AI456_FUD	Add prefix 123 replace AIS by AI replace 01 by 456	123AI456_FUD 123AI456 123AI456_Intl 123AI456_MV 123AlcBlk 123MV_GT_L1_ 123MV_LT_H1_ 123Value

Alternative solution for previous sample with pattern (NNN)LL(L)NN(N)(LLLLL)

Source	Result
AIS01_FUD	123AI456_FUD
AIS01	123AI456
AIS01_Intl	123AI456_Intl
AIS01_MV	123AI456_MV
AlcBlk	AlcBlk
MV_GT_L1_	MV_GT_L1_
MV_LT_H1_	MV_LT_H1_
Value	Value

Source	Target Name	To-do	Result
AIS01_Norm_FUD AIS01_Norm AIS01_Norm_Intl AIS01_Norm_MV AlcBlk MV_GT_L1_ MV_LT_H1_ Value	AIS01Norm_FUD	Remove _	AIS01Norm_FUD AIS01Norm AIS01Norm_Intl AIS01Norm_MV AlcBlk MV_GT_L1 MV_LT_H1 Value

MV_GT_L1_ was renamed to MV_GT_L1 because 1 is previous character in source name. Same applies for MV_LT_H1_.

Alternative solution for previous sample with pattern LL(L)NN(N)(L)LLL\m

Source	Result
AIS01_Norm_FUD	AIS01Norm_FUD
AIS01_Norm	AIS01Norm
AIS01_Norm_Intl	AIS01Norm_Intl
AIS01_Norm_MV	AIS01Norm_MV
AlcBlk	AlcBlk
MV_GT_L1_	MV_GT_L1_
MV_LT_H1_	MV_LT_H1_
Value	Value

Source	Target Name	To-do	Result
PRC1111_FD PRC1111 PRC1111_S PRC1111_FD_IC1 PVCD1111_FD_S PRXY1111_S	PXC1111_FD	Replace R by X	PXC1111_FD PXC1111 PXC1111_S PXC1111_FD_IC1 PVCD1111_FD_S PXXY1111_S

Source	Target Name	To-do	Result
PRC1111_FD PRC1111 PRC1111_S PRC1111_FD_IC1 PVCD1111_FD_S PRXY1111_S	YPRC1111_FD	Add prefix Y	YPRC1111_FD YPRC1111 YPRC1111_S YPRC1111_FD_IC1 YPVCD1111_FD_S YPRXY1111_S

Source	Target Name	To-do	Result
PID01MAST_FuD PID01MAST PID01MAST_Intl PID01MAST_Slav PID01MAST_MV	999TT927_FuD	add prefix 999 replace PID by TT replace 01 by 927 delete MAST	999TT927_FuD 999TT927 999TT927_Intl 999TT927_Slav 999TT927_MV

Paste Rename Functionality

In the regular copy paste operation names are automatically decided by the function settings, where as, Paste Rename functionality enables renaming of Function Diagrams and its various components interactively. Below steps describes the procedure to use Paste Rename functionality.

1. In the Engineering Workplace navigate to the source Function Diagram object under **Functional Structure**, right-click and select **Copy**.

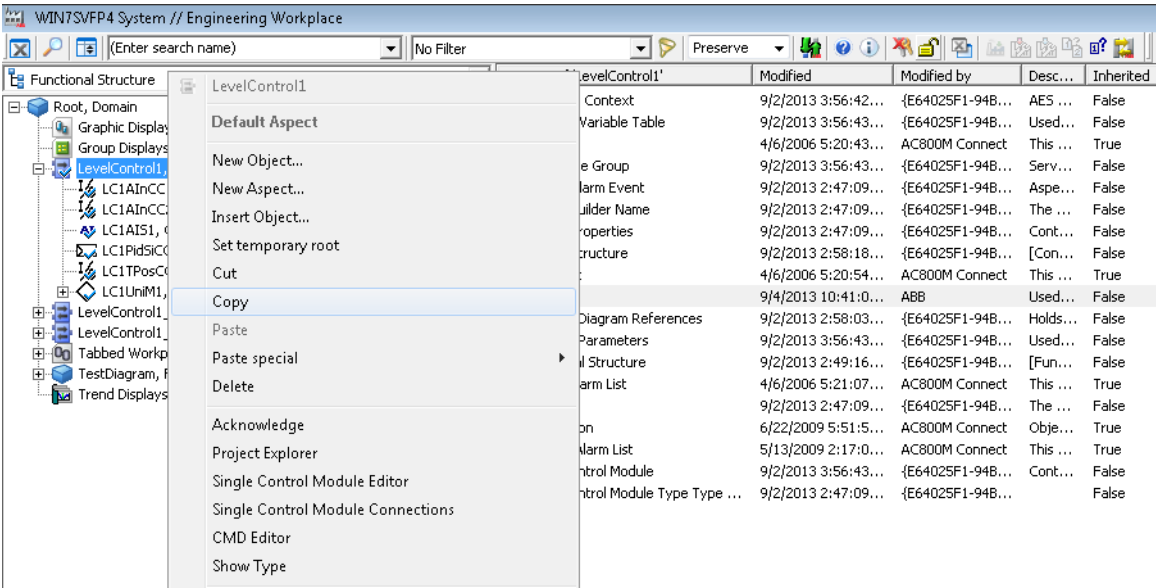


Figure 95. Function Diagram Object Copying

- 2. Right-click on the intended **Object**, navigate to **Advanced** and select **Paste Rename**.

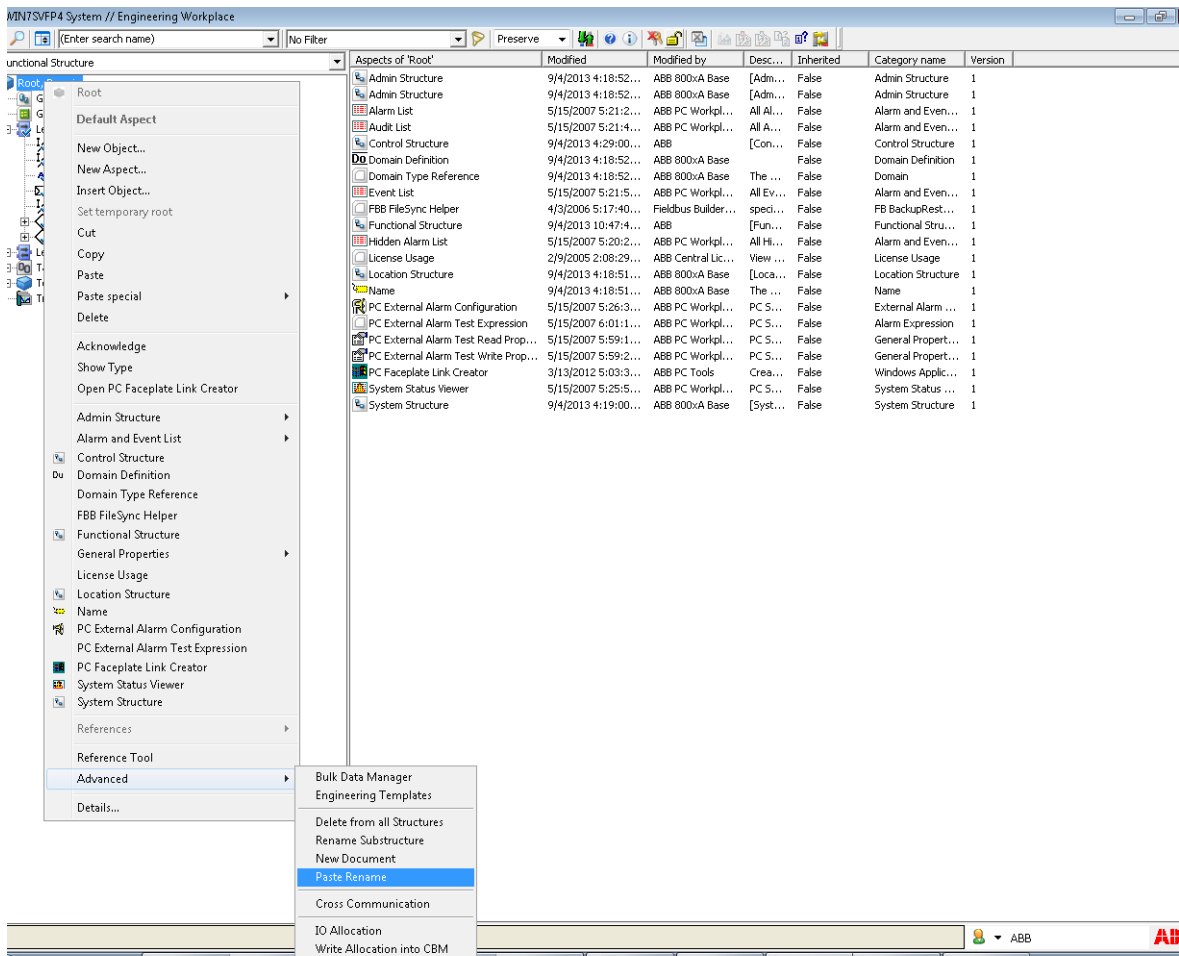


Figure 96. Paste Rename Functionality

3. **Paste Rename** dialog appears with a unique Diagram Name along with the list of all components of Function Diagram, see [Figure 97](#).

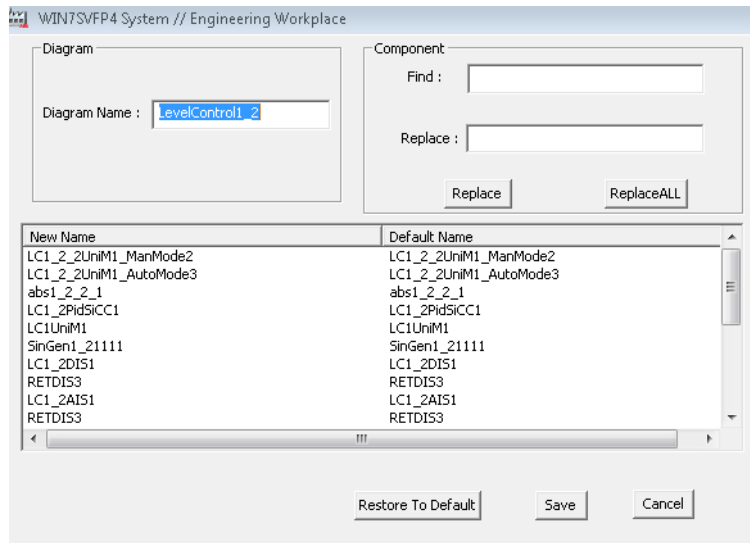


Figure 97. Paste Rename Dialog

4. In **Diagram Name** field enter a new name if required and press **Enter** to commit the change.

- 5. Enter the required strings in **Find** and **Replace** fields, see [Figure 98](#).

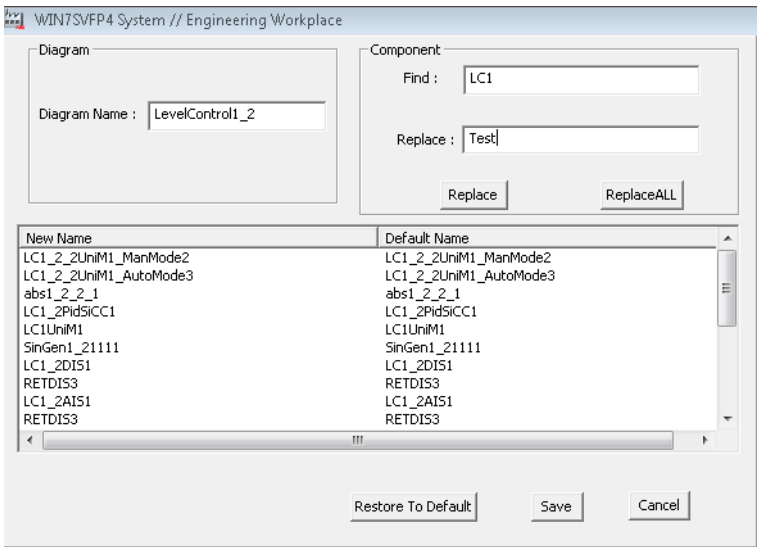


Figure 98. Find and Replace Dialog

- 6. Click **Replace** to rename a single string. As a results last component of the **New Name** list gets replaced.
- 7. Else click **ReplaceAll** to rename all the components having same string.

As an example, **ReplaceAll** is selected for illustrative purpose, see [Figure 99](#). All the components named **LC1** are replaced with **Test** under **New Name** list.

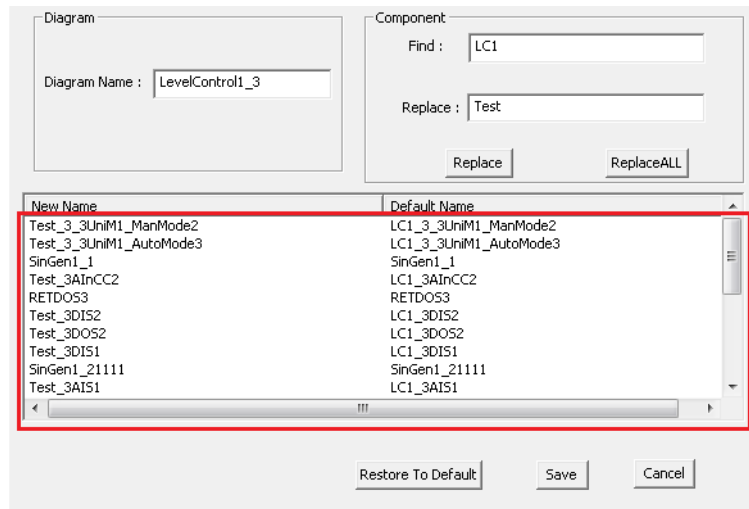


Figure 99. Components Renamed Dialog

8. Click **Cancel** to cancel the changes and close the dialog box.
9. Click **Restore to Default** to revert back the original component names under **New Name** list.
10. Click **Save**, to save the changes in the destination diagram. A popup appears, click **OK**.

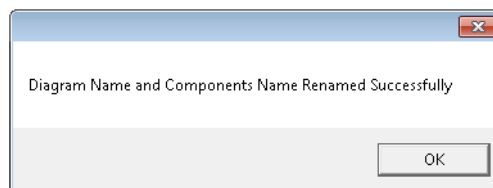
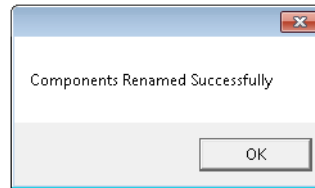


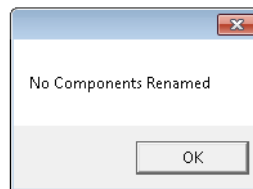
Figure 100. Conformation Popup



If **only** components are replaced and changes are saved, following popup appears:



If **Save** option is clicked without any changes to the Function Diagram name and its components, following popup appears:



Communication Between Applications using Cross Communication

Cross communication links must be configured to establish communication between different applications of an AC 800M controller or several AC 800M controllers. Also, the Function Diagrams must be allocated to different applications of the same or different control projects.



For details on configuring and generating MMS Cross Communication for diagrams created prior to Engineering Studio 800xA 5.1, refer to the [MMS Cross Communication](#).

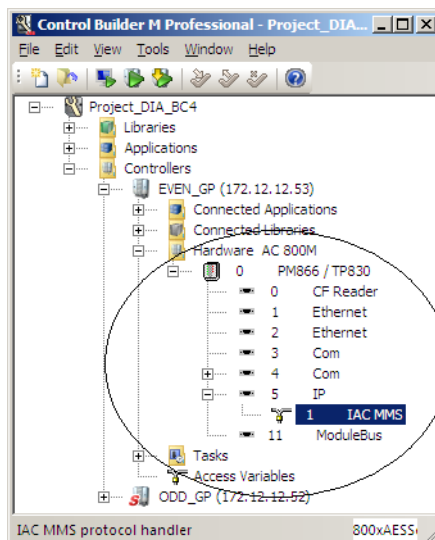
Prerequisites

The following prerequisites must be fulfilled to successfully establish cross communication:

- All Function Diagrams containing Communication Variables must be allocated to required applications.
- Generate configuration data for these diagrams.

- Applications executing these Function Diagrams must be assigned to controllers and tasks.

- For projects restored to system version 5.1, if the IAC MMS unit is not available as indicated in the diagram below, then user needs to manually insert the unit.



Cross Communication Procedure

The example in this subsection describes the procedure to create cross communication diagrams.

- Create two Function Diagrams named SenderDiagram1, and ReceiverDiagram1.
- Create logic in SenderDiagram1 and ReceiverDiagram1 as shown in [Figure 101](#).

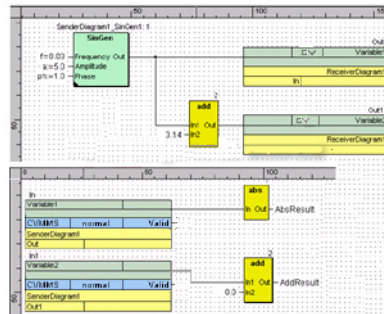


Figure 101. Diagram Variables without Cross Communication Field

3. Allocate each diagram to a different Application (see Figure 102).

The Communication Variable Variable1 is source for the diagram reference in SenderDiagram1 and sink for the diagram references in ReceiverDiagram1.

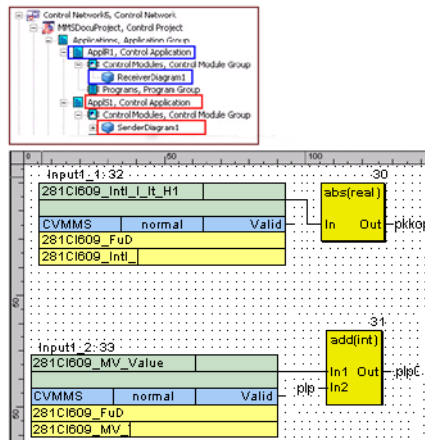


Figure 102. Diagram Variables with Cross Communication Field

4. Generate configuration data for these Function Diagrams.

Configuration data generation establishes cross communication links between the diagrams through the Communication Variables according to the engineered logic.

Additional basic information on cross communication is given in *System 800xA Control, AC 800M Configuration (3BSE035980*)* and *AC 800M, Communication Protocols (3BSE035982*)*.



From 800xA 5.1 onwards, the Inter Application Communication supports the Input Set as Predetermined (ISP) functionality. The input diagram reference is used to set the ISP value. The Diagram Variable Properties dialog passes the ISP value to the input Communication Variable.

When the inter communication fails, the input variable gets a predefined value, or a deterministic value. If a good value is not available in the previous list of values, the input variable shall be set to its defined initial value.

Deleting a Diagram Variable or Changing the Name

If the user deletes the diagram reference with variable Variable2 in source SenderDiagram1 or renames the diagram variable, then configuration data should be generated for the sender and all its receiver diagrams.

If only the application (SenderDiagram1) of the sender Function Diagram is downloaded to the controller and the applications of the receiver Function Diagrams are not yet loaded, then the variables of the ReceiverDiagram1 hold the last Retain Values. Give reference to ISP value.

Download and go Online

After downloading the project data into the controller(s) you can check the Valid ports at the MMS diagram references. Subscribe for live data and select the Valid

ports for display, see chapter [On-line Display](#).

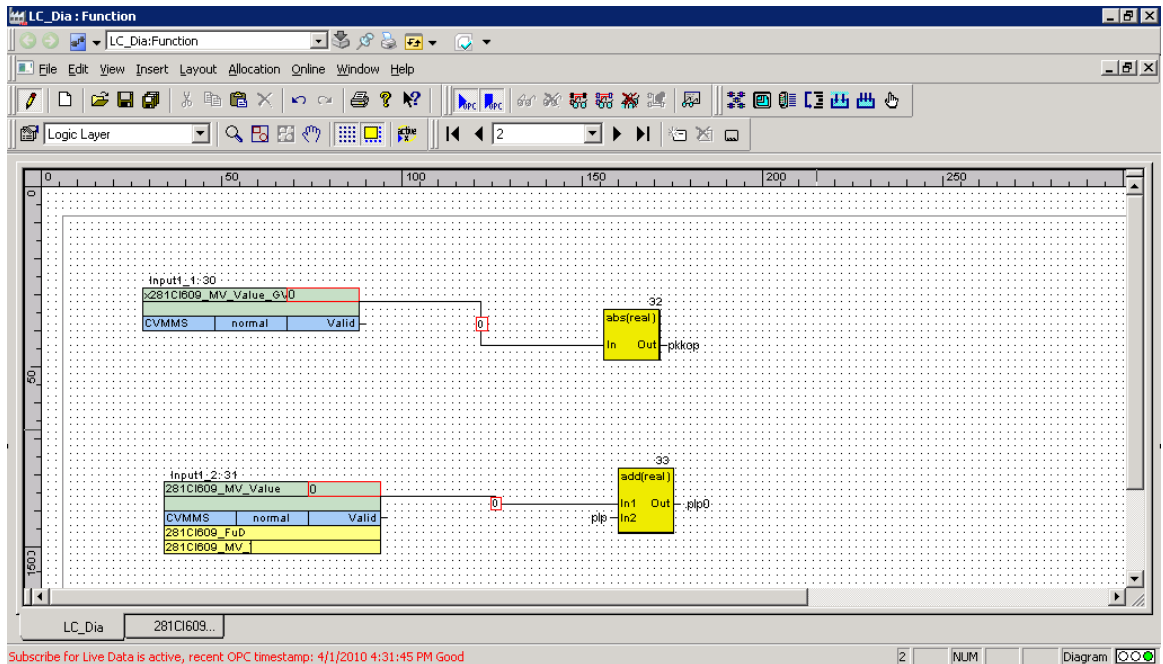


Figure 103. Live Data

User can verify the correct behavior of MMS communication through an Alarm and Event List aspect. In case of communication failures, an alarm gets generated to indicate the faulty sender or receiver.

Current Limitations

- Connection to multiple output diagram references in different applications having same diagram variable name is not supported.

On-line Display

General

Online display of values is possible for a diagram or a portion of a diagram that is in state 'Loaded'.

Display of online values is also enabled for modified diagrams if they had been generated at least once (i.e. allocatable group has a FDVersion property). A warning is given that the modified diagram does not match the one running in the controller, similar to the case that a regenerated diagram is not yet loaded.



On-line values cannot be modified directly in Function Diagram. Use Watch Window instead, see [Watch Window](#).

Display Style

Function Designer follows the 800xA system standard faceplate's style regarding the use of colors and other graphical elements for value display. However, because the faceplate do not have to distinguish between live data and configuration data, Function Designer uses an additional graphical element to indicate that a value is a 'live' value.

Table 19. Display Style of On-line Values

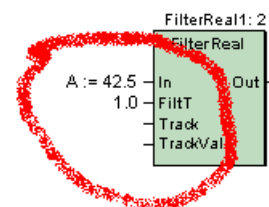
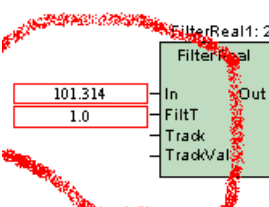
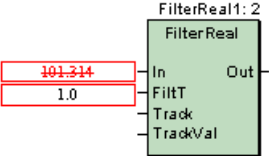

Description	Display Style
A configuration value (initial value).	 <p>The diagram shows a green rectangular block labeled 'FilterReal1: 2'. To its left, the text 'A := 42.5' is displayed. Below this text are four labels: 'In', 'FiltT', 'Track', and 'TrackVal'. The 'In' label is connected to the block. The 'FiltT' label has a value of '1.0' next to it. The 'Track' and 'TrackVal' labels are connected to the block. The 'Out' label is on the right side of the block. A red dashed line is drawn around the 'A := 42.5' text and the 'In' label.</p>
Live values are indicated by a red frame around the value.	 <p>The diagram shows a green rectangular block labeled 'FilterReal1: 2'. To its left, the text '101.314' is displayed in a red rectangular frame. Below this text are four labels: 'In', 'FiltT', 'Track', and 'TrackVal'. The 'In' label is connected to the block. The 'FiltT' label has a value of '1.0' next to it. The 'Track' and 'TrackVal' labels are connected to the block. The 'Out' label is on the right side of the block. A red dashed line is drawn around the '101.314' text and the 'In' label.</p>

Table 19. Display Style of On-line Values (Continued)

Description	Display Style
If a property indicates a bad quality (from the OPC server), the value is displayed in red color with the strikeout font.	
A forced value is indicated using a gray background color.	<p>CBM_AIS19 ==.4</p>  <p>Filter: 0 Input: 4-20mA Range: 0 - 100</p> <p>IOValue: 0 Value: 21</p>

You can select to display the actual process value for a port, a connection link, or for all ports of a component. On-line values are received from the OPC server. The values are available for the following ports, connections, and components:

Table 20. Display of On-line Values

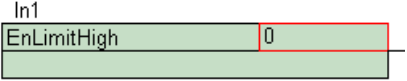
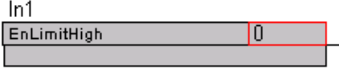
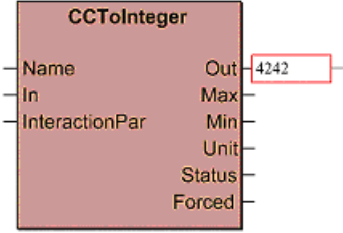
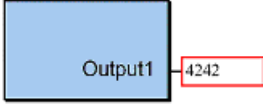
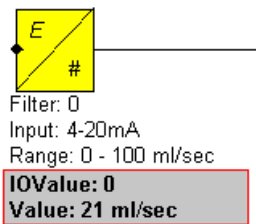
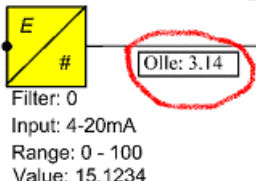
Item	Description	Example
Diagram Reference	Diagram references are implemented as global variables. The symbol for the diagram reference contains a field to display the process value.	
Diagram Parameter*	Diagram parameters are implemented as input or output variable of a single control module. The symbol for the diagram parameter contains a field to display the process value.	
Port of Control Module Port of Function Block	All terminals of all Control Modules are OPC readable.	
Port of Child Function Diagram	All input and output ports of a nested function are readable.	

Table 20. Display of On-line Values (Continued)

Item	Description	Example
Port of Function	<p>Ports of Functions are not accessible through OPC.</p> <ul style="list-style-type: none"> If the port of the Function is connected to an OPC readable variable, the value of the variable is displayed. If the port is connected to a '___link' type variable, display of on-line values depends on how configuration data was generated. In case of 'Online Optimized' code, all '___link' variables are automatically converted to 'internal_link' variables that are OPC readable. Otherwise, internal links between functions are not OPC readable. This is indicated by a '*' or a '?'.* 	
IO Signal	<p>For an IO signal, Function Designer generates a global variable of the following type:</p> <p>RealIO</p> <p>BoolIO</p> <p>Function Designer displays the "Value" part of the variable and the "Unit" string (RealIO only). If the value is forced, it is displayed in bold with grey background color.</p>	<p>CBM_AIS19 ==.4</p> 
Networks	<p>Networks correspond to variables. The variables are OPC readable if:</p> <p>a) You defined an explicit network name.</p> <p>b) You selected to generate "Online Optimized" code.</p>	<p>CBM_AIS6 \$.:Control Structure:ARD_</p> 

* - Valid only for Function Diagrams restored to 800xA 5.1.

Function Designer is aware of some frequently used data type. [Table 21](#) describes these data types, and how the Function Designer displays them in on-line display.

Other structured data types with a 'Value' record element display that value. Other structured data types without 'Value' are not displayed in the diagram directly. This is indicated by a "?" or "*". However, their on-line values can be displayed in the [Watch Window](#).

Table 21. Display of On-line Values for Some Common Data Types

Type Name	Usage	Online Display
RealIO	Connection to process IO.	RealIO.SignalPar.Unit. Example: 3.14 m/s
BoolIO	Connection to process IO.	BoolIO.Value
ControlConnection	Two-way communication between control modules.	ControlConnection.Forward.Value ControlConnection.Forward.Range.Unit Example: 100 %
SignalPar		SignalPar.Min SignalPar.Unit <= X <= SignalPar.Max SignalPar.Unit Example: 10 m/s <= X <= 12.5 m/s
Range	Value range.	Range.Min Range.Unit<=X<= Range.Max Range.Unit Example: 10 m/s <= X <= 12.5 m/s
HWStatus	IO Diagnostics	HWStatus.HwState Example: 0xADD45000
DwordIO	Connection to process IO	DwordIO.Value
DintIO	Connection to process IO	DintIO.Value
Any other structured data type with a .Value record element.		AnyType.Value

User Interface

All online commands are available from the **Online** menu, as shown in Figure below:

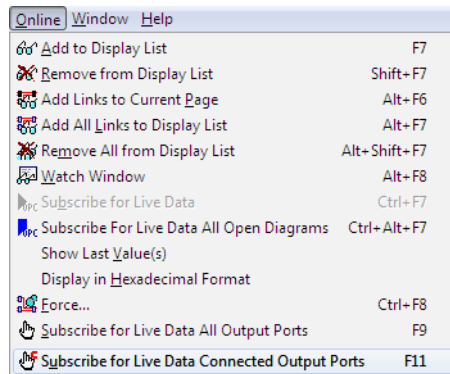


Figure 104. Online Menu

Online > Add to Display List

This command adds the selected object to the list of objects where you want to see online value. Following objects are supported:

- a single port
- a multi-selection of ports
- a component symbol (all ports will be added)
- a network link
- an IO signal

It is also possible to add a combination of these objects. The command is using the current selection in the diagram as source. The command is also available in the for the objects listed above.

- **Online > Remove from Display List**

This command reverses the action of the command [Online > Add to Display List](#).

- **Online > Add Links to Current Page**
This menu command displays the on-line values for all the links on the current page.
- **Online > Add all Links to Display List**
This command adds all connection links on the current page to the display list.
- **Online > Remove all from Display List**
This command reverses the action of the command [Online > Add all Links to Display List](#).
- **Online > Watch Window**
In Diagram view, choose this menu command to open the Watch window. The command displays the watch window. The window shows a grid where you can add the (OPC) name of variables you are interested in. See [Watch Window](#).
- **Online > Show Last Value(s)**
This command show/hides recent on-line values retrieved from the OPC server if subscription for live data has stopped.
- **Online > Display in Hexadecimal Format**
This menu command displays the online values in hexadecimal format.
- **Online > Force...**
This dialog allows to force the Value of an allocated input IO signal or the IOValue of an allocated input IO signal.

Subscribe to Live Data Functionalities

From the below mentioned Live Data functionalities user can enable only one option at a time.

- **Online > Subscribe for Live Data**
This command starts/stops on-line display of live data. The recent OPC timestamp and quality is displayed in Function Designer's status bar, see [Figure 105](#).

- **Online > Subscribe For Live Data All Open Diagrams**

In Diagram view, choose this toggle menu command to start/stop display of on-line values collected in the diagram's display list and/or collected in the Watch window for all open diagrams.

- **Online > Subscribe for Live Data All Output Ports**

The command adds labels to display live values for all the pages of function diagram and subscribes live data for the current page.

- **Online > Subscribe for Live Data Connected Output Ports**

The command adds labels to display live values for the connected output ports and subscribes live data for the current page.



- In Function Diagrams, links connected through the output port of MOVE block are not animated with color.
- Subscribe for Live Data is not supported for Extensible Parameters.

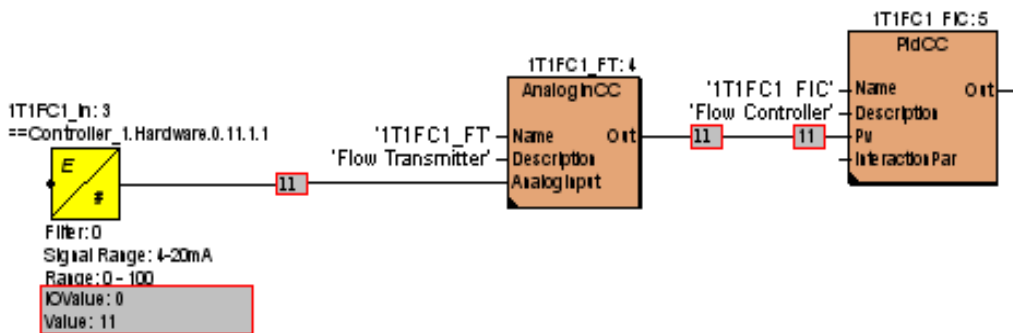


Figure 105. Diagram in Loaded State Displaying On-Line Values

Watch Window

This section describes, how to use the Watch window of Function Designer. The purpose of the Watch window is to display a user defined collection of on-line values. This function is typically used during the commissioning phase of a function.

To display the watch window, use command **Online > Watch Window**. Like the structure browsers and the output window, the Watch window can be

- docked inside Function Designer, Diagram View
- undocked as popup-window.

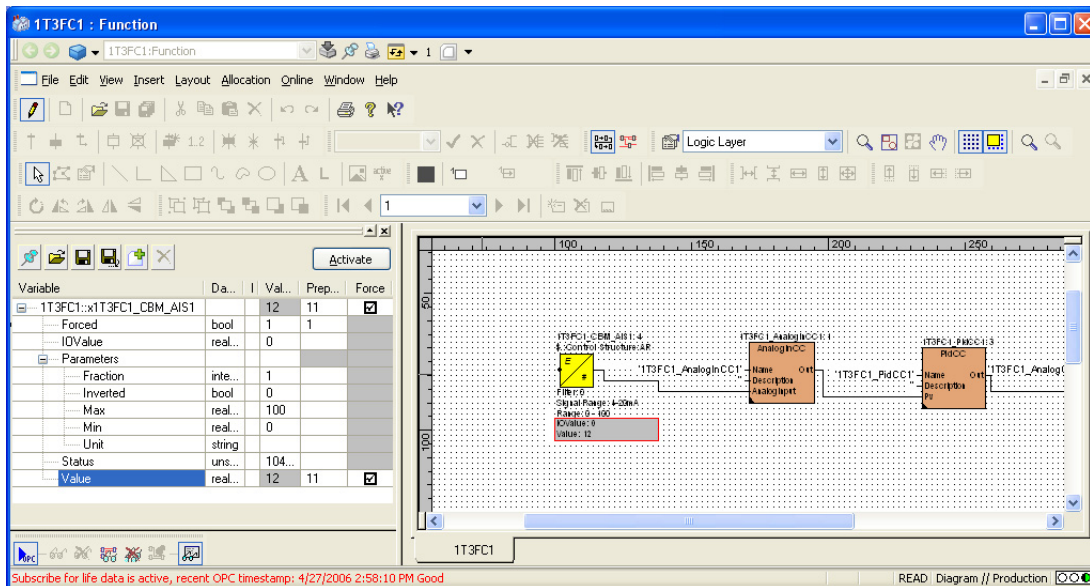


Figure 106. Watch Window for a Function Diagram Online Display

The watch-window displays online values from:

- The current function diagram that you are viewing.
- Any other diagram.
- Any other aspect, not belonging to a function diagram (a Control Builder Application for example).

If a variable is of a structured data type, the watch window displays **struct** as data type. You can click on such a structured variable to expand it and to view the individual elements it consists of.

How to add variables see [Adding Variables](#).

To active the value update, use command [Online > Subscribe for Live Data](#).

Function Designer saves the configuration of the Watch window. When opening the same Function Diagram next time, the watch window will show the same configuration. If any of the variables in the watch window does not exist anymore in you diagram, it will be marked as invalid.

Saving the Watch Window can be done implicitly during exit of the Function Diagram or explicitly using the **Save / Save As** commands, see [Watch Window Commands](#).

Watch Window data are stored in an aspect of the aspect category Function Online Data.

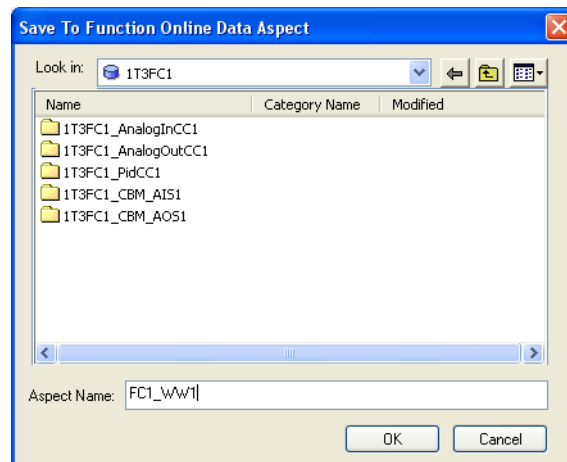


Figure 107. Save Watch Window Configuration to Function Online Data Aspect

Aspects of this aspect category are copied. An aspect object (typically the one representing a Function Diagram or a Function Diagram Type) may carry 0...n aspects of the category Function Online Data.

When a diagram or a diagram type is opened the Watch Window is filled with the data used last time. This is ensured by loading the Function Online Data aspect with

the latest modification date. When closing the Watch Window the aspect modification date is always updated (even if no data is saved).

Aspects of the aspect category Function Online Data do not have an aspect view.

Watch Window Grid

The Watch Window grid displays the variables to be watched. For each variable the following properties are displayed:

- **Variable Name**
- **Data Type**
- **OPC Value**
- **Prepared Value**
- a **Force** check box determining whether to force (if applicable) or to set the prepared value

The values in the cells of columns **Variable Name**, **Data Type** and **OPC Value** are read-only in the grid. The cells of columns **Prepared Value** and the **Force** check box can be edited. Like in online display in diagrams together with the **OPC Value**, its quality (bad quality is indicated by red text color and strike through font), and the information whether it is forced is displayed (in this case the value is displayed on a gray background).

The **Force** check box shows up only for variables which can be forced at all (i.e. if the according data type has got a component called **Forced**) in order to distinguish between forcing and setting the prepared value. This applies to the following data types:

- RealIO
- BoolIO
- DintIO
- DWordIO
- ControlConnection (component Forward.Forced)
- ControlConnectionForward

For an alternative to force CBM_Signals see [Force Dialog](#).

Data types which have a **Value** component but no **Forced** component can only be set (i.e. the set value could be overwritten by the controller in the next cycle).

For data types which don't have a **Value** component the entries in the **Value** and **Prepared Value** columns are missing (exception: for the data type HWStatus the component **HWState** is taken instead).



Communication Variables are read only and do not support the entry of a **Prepared Value**.

If the OPC property is not writable the according **Prepared Value** edit field is disabled (dimmed).

If for a structured variable the **Value** and/or the **Forced** component exist as separate entries (besides the variable itself) in the Watch Window grid the according **Prepared Value** and **Force** entries are synchronized.

If editing a Prepared Value is finished by pressing <CR> or <Enter> the input focus changes to the **Activate** button in order to allow the user to activate the prepared value quickly.

Operations on variable entries in the grid:

- **Expand** - expands the next level of a variable of structured data type
Expanded components can be added as separate entries to the Watch Window with a item "Add".
- **Collapse** - collapses the level of a variable of structured data type
- **Move** - A Watch Window entry can be moved to another position by means of drag & drop.
- **Select All** - Using the Ctrl/A accelerator all entries can be selected before performing commands like **Activate**, **Deactivate**, **Remove**, etc.

Watch Window Commands

- **Pin / Unpin** - If pinned, the Watch Window contents is kept even if another diagram gets active in the Function Designer's client area (e.g. when opening a child diagram or when navigating via Diagram References).
If unpinned, the Watch Window contents is cleared and reloaded according to

the related Watch Window data of the activated diagram (see also "Function Online Data" aspect). Refilling is done when the diagram gets active.

- **Open** - Offers the possibility to chose a Function Online Data aspect (of any object) and to fill the Watch Window with the according data.
- **Save** - Stores the Watch Window data in the Function Online Data aspect from which it was loaded. In case the Watch Window is "dirty", i.e. was changed after Open, the user is queried to save its contents.
- **Save As** - Offers the possibility to store the Watch Window data in a (new) Function Online Data aspect (of any object), see [Figure 107](#).
- **Add** - Pops up the Add Variable to Watch Window dialog which allows adding a new variable to the Watch Window. See [Adding Variables](#)
- **Remove** - Removes the selected variables are removed from the Watch Window (using the Ctrl/A accelerator all variables can be selected to remove them). Besides the toolbar the Remove command is available in a popping up on selected variables.
- **Activate** - Activates the selected prepared values in the controller (using the Ctrl/A accelerator all prepared values can be activated in the controller). The activation has to be acknowledged by the user. If the according **Forced** check box is checked the prepared value is forced in the controller (i.e. held permanently). If the according **Forced** check box is unchecked the prepared value is set in the controller (i.e. may be overwritten by the controller in the next cycle). Besides the toolbar the **Activate** command is available in a popping up on selected variables.
- **Deactivate** - Deactivates the selected values in the controller (using the Ctrl/A accelerator before all values can be deactivated in the controller). The deactivation has to be acknowledged by the user. The prepared values are reset to "" and the according **Force** check boxes get unchecked. Besides the toolbar the **Deactivate** command is available in a popping up on selected variables.

Adding Variables

To add a new variable to the watch window, click the **Add...** button. Function Designer displays the following dialog:

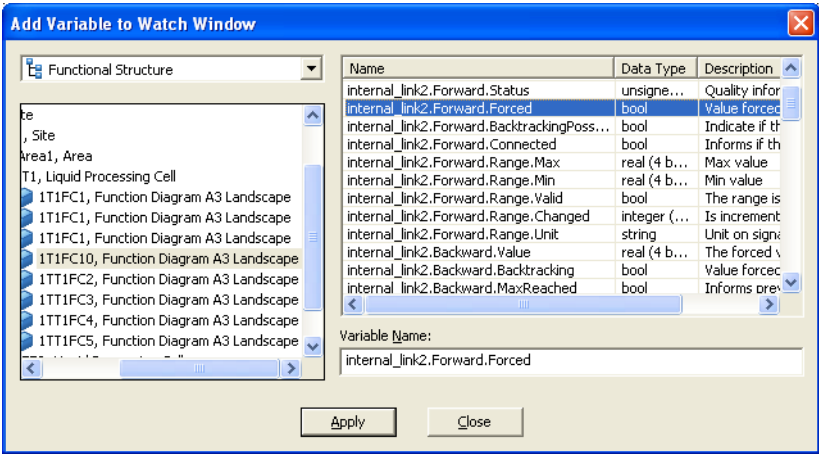


Figure 108. Add Watch Variable

Select one or more variables from the list in the right part of the window and click **Apply** to add the selected variable to the watch window.

You can also type the name of a variable in the edit field.



To display a variable of a structured data type, type the name of the variable up to the first dot into the edit field and press **Apply**. Example x1T3FC1_CBM_AIS1.



The Watch Window settings are not stored for a specific user but they are shared for all users of this diagram.

Alternative for adding variables to the Watch Window: The **Online** menu and the s which pop up on ports, links and symbols provide a menu item Add to Watch Window. When performed on symbols (the variables related to) all visible ports of the symbol get added to the Watch Window.

Force Dialog

An alternative to the Watch Window functionality to activate a force value is the entry **Force ...** or **Online > Force...** on ports and links. The command is enabled for the Value (data types RealIO, and BoolIO) of an allocated input IO signal (object type CBM_AIS, CBM_DIS) and the IOValue (datatype RealIO, BoolIO) of an allocated input IO signal (object type CBM_AOS, CBM_DOS). This command displays the dialog:

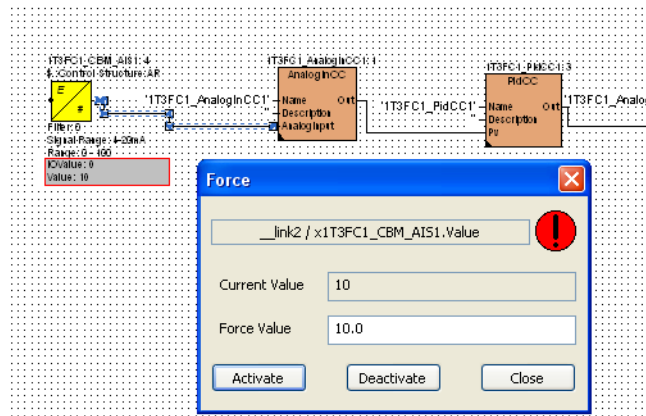


Figure 109. Force Dialog, Value of a CBM_AIS Forced

Field Device Integration

Introduction

The Fieldbus Builder PROFIBUS/HART Integration of *Function Designer* supports a basic workflow of functional planning with HART device (object) types represented as CBM_Signals. See [HART Devices](#). HART device types by default provide a function aspect with a function component and can be instantiated in the Functional Structure.

From 800xA 5.0 SP2 onwards a basic workflow of functional planning with CBM_Signals to be allocated to channels of modules of PROFIBUS devices is also supported. New CBM_Signal objects (CBM_DINTIS, CBM_DINTOS, CBM_DWIS, CBM_DWOS, and CBM_HWStatus) have been introduced in Function Designer to support PROFIBUS devices. These signal objects support DInt and DWord data types. See [PROFIBUS Devices](#).

For HART, PROFIBUS and FOUNDATION Fieldbus devices it is supported to reference and navigate to the IO variables representing these devices in a control application. This support bases on the general functionality of Function Designer's AC 800M integration to reference (and even create) application global IO variables via diagram references. After the variable is connected to the hardware the diagram reference shows the control designation of the device. The context menu on the reference can be used to navigate to the corresponding hardware editor in Control Builder M Professional, see [Referencing Devices](#).

PROFIBUS and FOUNDATION Fieldbus device types by default do not provide a function aspect with a function component and therefore cannot be inserted into function diagrams. Alternatively you can add an own Function aspect containing a Function Component view to an instance of such a device type, see [Function Components and Diagrams](#). You then can insert the instances¹ from the Control Structure into the Functional Structure below corresponding Function Diagram objects. They show up in the function diagram and can serve to fulfill your documentation and, via context menu, your aspect navigation needs. If you copy such a function diagram object the device representations contained in the diagram are removed. You have to insert other devices in the new diagram.

1. While PROFIBUS device types can only be instantiated in the Control Structure, FOUNDATION Fieldbus device type instances can only be uploaded into the Control Structure.

HART Devices

Prerequisites

The system extension ***Function Designer for Fieldbus Builder PROFIBUS/HART*** is based on the system extensions ***Function Designer*** and on ***HART Device Integration Library*** and ***Fieldbus Builder PROFIBUS/HART***. All tools have to be installed, all system extensions have to be loaded.

To load the system extension ***Function Designer for FB P/H***, use the Configuration Wizard via **System Administration > (System Name) > System Extension Load** and select first ***HART Device Integration Library - Basics*** followed by ***Fieldbus Builder PROFIBUS/HART*** and then ***Function Designer for Fieldbus Builder PROFIBUS/HART***.

Object Types for HART Devices

Following object types for HART devices (HART device types) are provided by default in Object Type Structure, Object Types\Field Devices, see [Figure 110](#):

- HART Actuators
 - ABB Generic HART Actuator
- HART Transmitter
 - ABB Generic HART Transmitter

Additional specific HART device types can be installed from the Device Library DVDs via the Device Library Wizard below the object type groups HART Actuators and HART Transmitter.

All standard HART device types should already include the aspects needed to be treated as signal object types, see HART device support in section IO Allocation of *System 800xA Engineering, Engineering Studio (3BDS011223*)*.

If this is not the case you can add these aspects using the HWDProcessor utility as described in this referenced instruction book.



In earlier system versions it was supported to have a separate CBM_Signal object for a HART device object and to have both objects represented in a function Diagram. Now the HART device represents the CBM_Signal object for a HART device object also.

Projects already using two separate objects can continue with the same objects. In this case the Fieldbus Management view on the board object is not updated by IO Allocation.



Configuration Data Generation of restored Function Diagrams using HART devices connected to CBM_Signals results in errors. Hence, user needs to ensure that the HART device object types have not been preprocessed with HWDProcessor.

Follow the steps below to configure the HART devices manually after IO Allocation of CBM_Signals:

1. Assign the HART device object to the IO board object (for example AI895 or AO895) in Control Structure, using either SHIFT+CTRL Drag & Drop from Function Diagram or Insert Object in Plant Explorer.
2. In the Main View of the Fieldbus Management aspect on the IO board object in Control Structure use Drag & Drop to assign the HART device object to the same channel as the corresponding CBM_Signal object.

System extension ***Function Designer for Fieldbus Builder PROFIBUS/HART*** adds the following aspects to all these device (object) types:

- Function
The HART Actuators object type group object and the HART Transmitter object type group object each have one Function aspect for the whole corresponding group of HART device types.

- Function Parameters

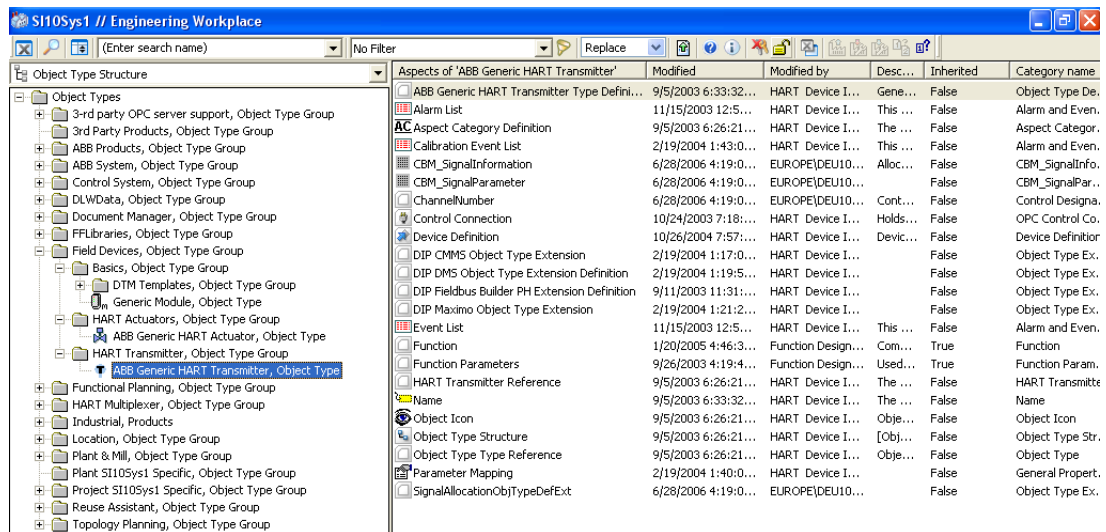


Figure 110. Generic HART Device Type Objects

For a specific HART device that is not supported you can prepare a copy of the generic device type e.g. Generic HART Actuator or Transmitter. Check that the device (object) types include the aspects Function and Function Parameters.

Building up the Functional Structure

You can build up the Functional Structure by inserting HART device types in a function diagram of *Function Designer*. You connect them to other function components on the diagram as you do it with CBM signal objects. DTM (Device Management) aspects, Fieldbus Management aspects, and other aspects of the

device object can be selected inside the function diagram opening the component by right mouse click.

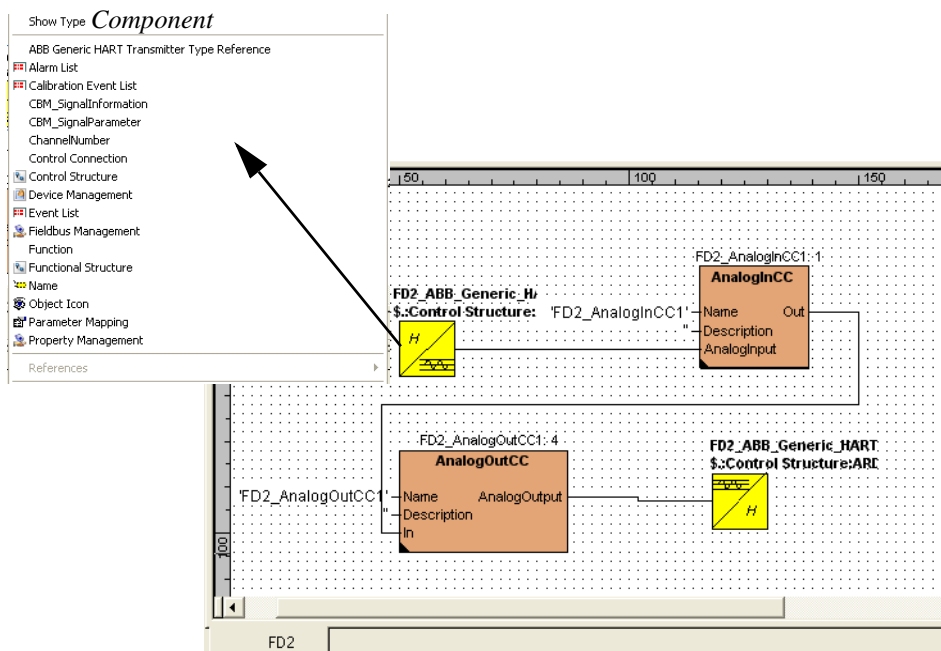


Figure 111. HART Device Objects in Function Diagram

Allocating Devices to the Control Structure

Instances of the HART device types can be allocated to channels of HART-IO boards, for example AI895/AO895, with the IO Allocation tool:

1. In Function Designer with the opened diagram click **Allocation > IO Allocation** to open the IO Allocation tool in the context of this diagram. The right pane shows the HART devices in the context of this diagram.
2. On the Boards object in the left pane of the IO Allocation tool click right to open the structure browser.

3. Click **Insert Elements** and add the needed boards from the Control Structure of the structure browser opened.
4. Drag and drop the HART devices from the right pane to the left pane, for example onto the Boards object to automatically do the allocation.

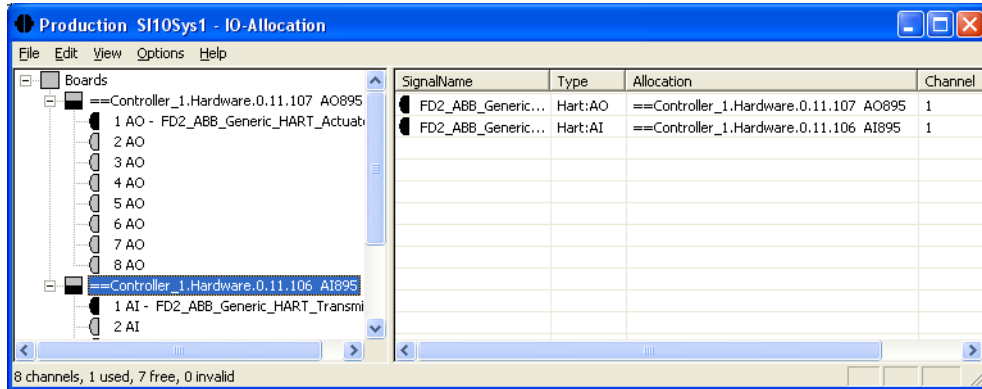


Figure 112. IO Allocation of HART Devices

In Function Designer generate configuration data to write the allocation information to Control Builder M Professional (or use **Write to CBM** item on the Boards object).

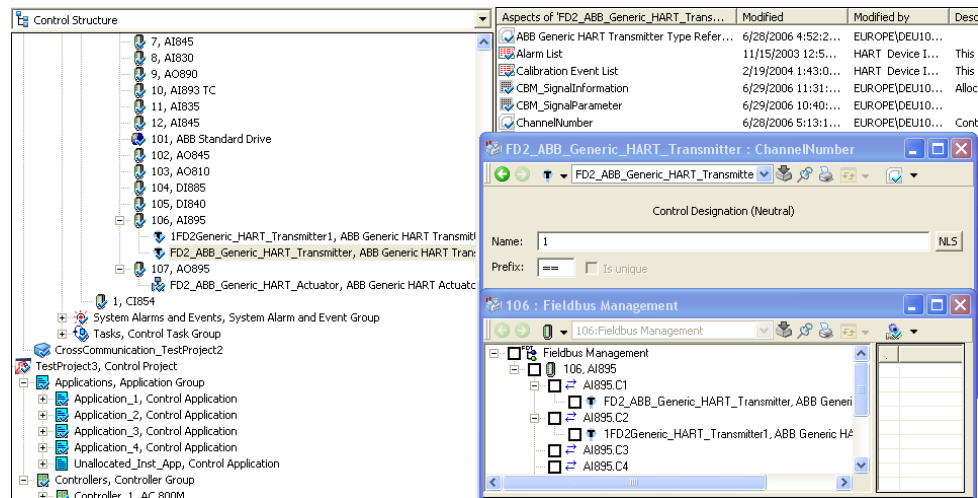


Figure 113. HART Devices Allocated in Control Structure

For further information on IO Allocation see *System 800xA Engineering, Engineering Studio (3BDS011223*)*.

Referencing of HART devices is possible only for Function Diagrams created on system versions prior to 5.1. For details on referencing of such Function Diagrams, refer to [Referencing Devices](#).

PROFIBUS Devices

CBM Signals for PROFIBUS

The following signal object types are introduced to support PROFIBUS device integration. They also can be used to read / write channels of I/O boards of corresponding data types. As well as the signal object types CBM_AIS,

CBM_AOS, CBM_DIS, CBM_DOS can be used to read / write channels of corresponding data types of PROFIBUS modules.

Signal Object Type Name	Description	Data Type
CBM_DINTIS	Double Integer Input Signal (input)	DInitIO
CBM_DINTOS	Double Integer Output Signal (output)	DInitIO
CBM_DWIS	Double Word Input Signal (input)	DWordIO
CBM_DWOS	Double Word Output Signal (output)	DWordIO
CBM_HWStatus	Hardware Status (input)	HWStatus

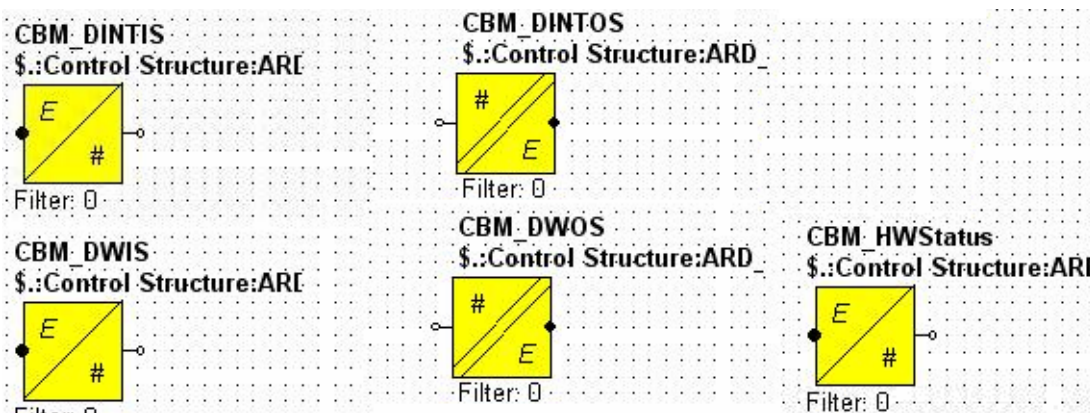


Figure 114. Symbols of CBM Signals for PROFIBUS Device Integration

Prerequisites

The system extension **Function Designer for Fieldbus Builder PROFIBUS/HART** is based on the system extensions **Function Designer** and on **PROFIBUS Device Integration Library** and **Fieldbus Builder PROFIBUS/HART**. All tools have to be installed, all system extensions have to be loaded.

To load the system extension Function Designer for **FB P/H**, use the Configuration Wizard via **System Administration > (System Name) > System Extension Load** and select first **PROFIBUS Device Integration Library - Basics** followed by **Fieldbus Builder PROFIBUS/HART** and then **Function Designer for Fieldbus Builder PROFIBUS/HART**.

After installing the software PROFIBUS devices need to be loaded on the system using Device Library Wizard (DLW).

Preparation of PROFIBUS Devices.

PROFIBUS specific device libraries can be installed through Device Library Wizard. User has to prepare these libraries using HWDPprocessor to use them in Engineering Studio for allocating or de-allocating signals to the channels of PROFIBUS modules.

For further information about preparation of PROFIBUS devices see *System 800xA Engineering, Engineering Studio (3BDS011223*)*.

Building up the Functional Structure

Functional Structure can be built by inserting PROFIBUS related signals in a Function Diagram. Connect them to other function components on the diagram as you do it with all other CBM signal objects.

Allocate the Function Diagram to an Application and Generate Configuration Data as explained in [Allocating Function Diagrams](#).

1. Select the signal object in Function Diagram.
2. Right click to open object context menu.
3. Then Device Functions verb can be used to execute the different device functions.

4. The aspects Device Management, Fieldbus Management or Property Management can be opened from context menu also.

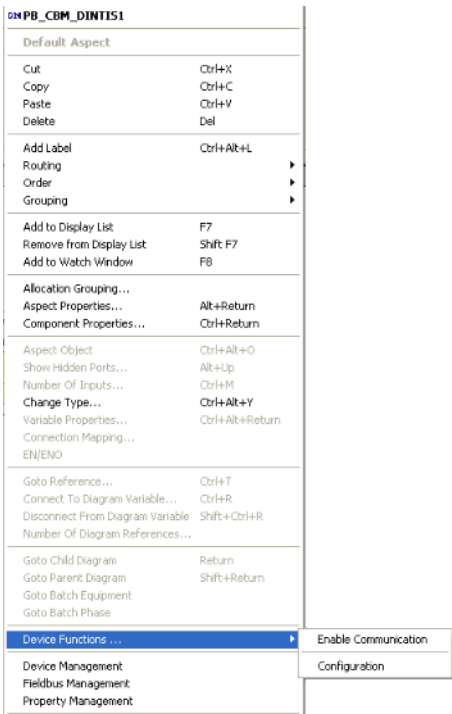


Figure 115. Object Context Menu with Device Functions Verb

Allocating Signals

After allocating the logic of the Function Diagrams to an Application in the Control Structure, the IO signals (CBM_Signals) need to be allocated to the channels of PROFIBUS modules using IO Allocation tool. Follow the steps below to allocate the IO signals.

1. In Function Designer with an opened diagram click Allocation > IO Allocation from the menu bar to open the IO Allocation tool in the context of the opened diagram.

- 2. Right click on the Boards object in the left pane of the IO Allocation tool and select Insert Element.
- 3. Add required PROFIBUS modules from the Control Structure of the structure browser opened.
- 4. Drag the signals from the right pane and drop them on to the Boards object in the left pane to allocate the signals automatically.

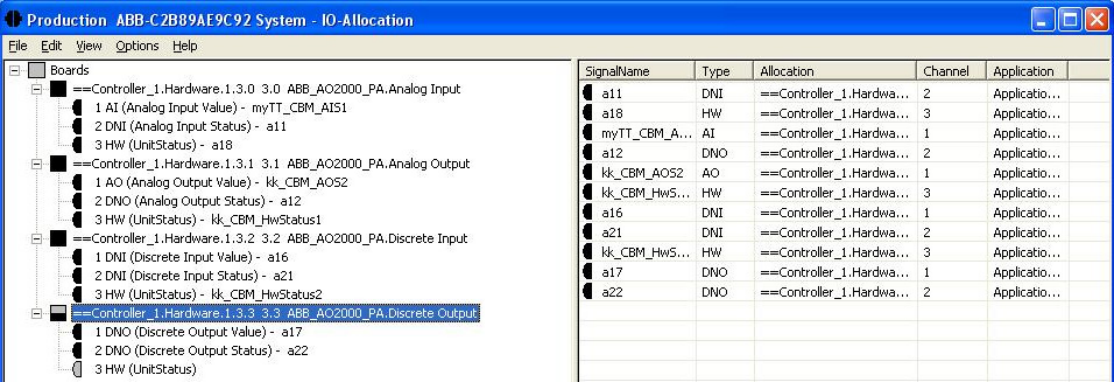


Figure 116. IO Allocation to Channels of PROFIBUS Modules

To write allocation information to Control Builder M Professional either generate configuration data in Function Designer or right click on the Boards object in IO Allocation tool and select Write Allocation into CBM.

User can write allocation information to CBM through Plant Explorer also. Follow the steps below to write to CBM through plant explorer.

- 1. Open Plant Explorer and select Control Structure in the structure browser.
- 2. Select required object, right click and select Advanced > Write Allocation into CBM.

FOUNDATION Fieldbus

Advanced IO for Foundation Fieldbus helps the user to use the Fieldbus Application Diagram (FBAD) in the Function Diagram Editor by creating the FF Proxy Object. Engineering Studio supports the Foundation Fieldbus (FF) diagrams for IO configuration. During the **Configuration Data Generation** the FF signals used in the FBAD diagram are updated to the control builder CI860 board.

The FBAD diagram used in the Function Diagram is published by FF Builder to the control builder CI860 board. This is done before allocating the FBAD diagram to the FF Proxy Object.

Loading the System Extension

To use the FF Proxy Objects in the Function diagrams, load the extension object created for the **ABB FF Proxy Extension for Function Designer**. As this extension has dependency with the Fieldbus extension, user has to load the required Fieldbus extensions before or along with the IO for FF.

Following image shows the extension for FF Proxy:

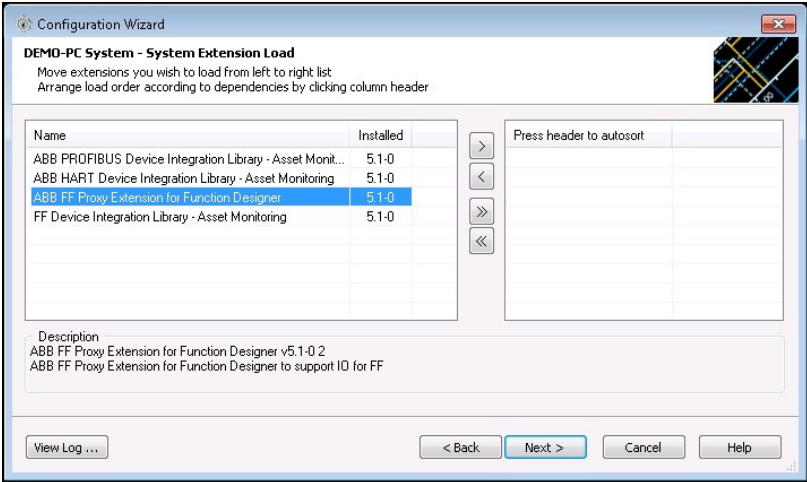


Figure 117. FF Proxy Extension

Once the extension is loaded successfully, the FF Proxy Object is available in the Object Type Structure and the Application Engineer can start using it in the Function Diagram.

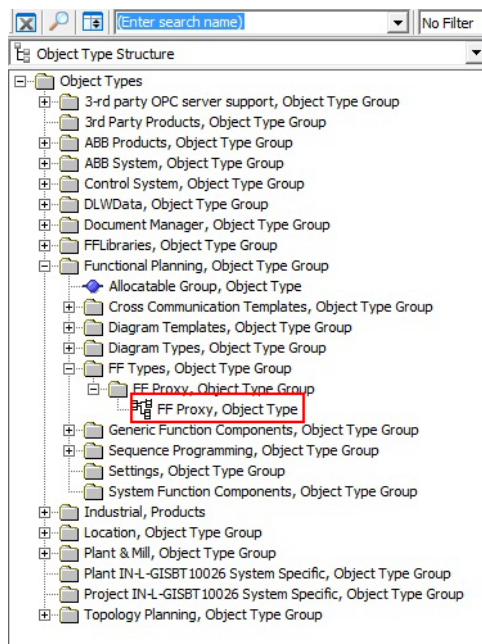


Figure 118. FF Proxy in Object Type Structure

Inserting FF Proxy Object to Function Diagram

Application Engineer can use the FBAD in Function Diagram through the FF Proxy Object available in the Object Type Structure. After inserting the FF Proxy Object in the Function Diagram, the new object inserted is defined as a generic object without any ports, as this proxy object does not know the FBAD diagram it is representing. Hence allocate the FBAD diagram to this proxy object.



If user generates the diagram without allocating the FBAD diagram, following warning message is displayed:

FF Proxy 'Proxy name' is not Allocated to FBAD

Procedure to add FBAD to the Function Diagram:

1. Drag and drop the FF Proxy Object from the Object Type Structure.

2. Select the valid FBAD to allocate the proxy object and click **OK** to create ports and save the allocation details.

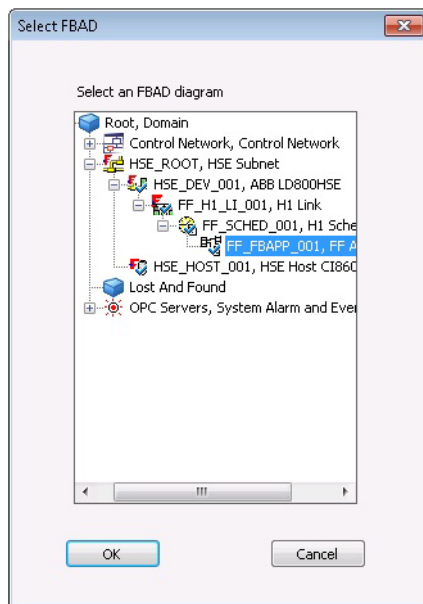


Figure 119. Select FBAD

Once the FBAD is allocated, ports are created to the FF Proxy object. Based on requirement, users can connect to the ports later.



Click **Cancel** to continue without allocating the FBAD.
The FBAD diagram can be allocated in later stage.

Allocating FBAD to FF Proxy Object

Once the proxy object is inserted to the Function Diagram, FBAD diagram is allocated /reallocated to the proxy object.

Application Engineer can allocate the FF Proxy from the context menu of the FF Proxy Object.

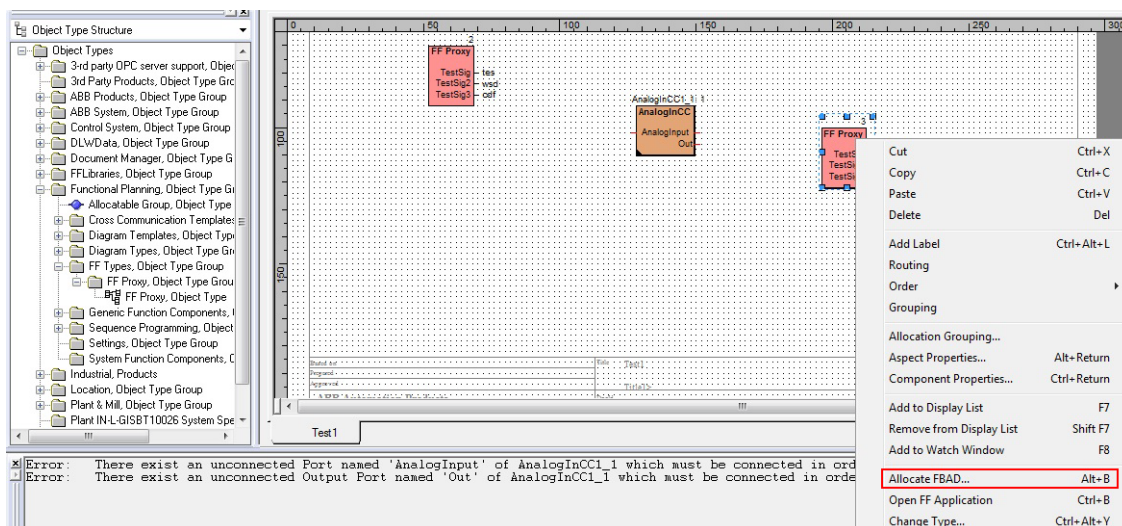


Figure 120. Allocate FBAD Context Menu



User has to select one proxy object at a time for the allocation.

To allocate FBAD:

1. Right-click on FF Proxy Object and click **Allocate FBAD** menu. **Select FBAD** dialog is displayed, for information on Select FBAD, refer [p. 213, Inserting FF Proxy Object to Function Diagram](#).

- 2. Select the FBAD Application for allocation. It validates the FBAD diagram published by the FF builder by checking the port information.



If the FBAD diagram is not published, following error message is displayed:
Selected FF Application is not consistent: Please select a valid FF Application

To avoid this error user has to publish the corresponding diagram from the FF builder.

After the allocation user can insert glue logic such as communication variables, global variables, and page connectors based on user specific requirement.

Configuration Data Generation on FF Proxy Object

Configuration Data Generation creates the corresponding variable in CBM, based on the signal type for each port.

	Name	Data Type	Attributes	Initial Value	Description
1	FDVERSION	date_and_time	constant	2012-02-15	Variable required by Function Designer for
2	TestVar1	FFBoolConnection			
3	TestVar2	FFRealConnection			
4	TestVar3	FFRealConnection			
5					
6					
7					
8					

Variables Communication Variables External Variables

Figure 121. FF Variable Generation

Created variables are allocated to the respective channels CI860 board's in CBM.

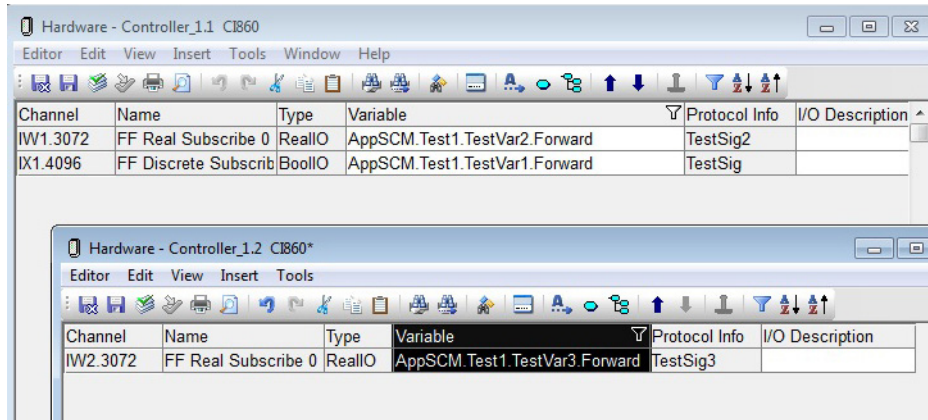


Figure 122. Hardware Controller CI860

If the FF Proxy Object is not allocated to any FBAD diagram then configuration data generation will continue with following warning message:

Warning: FF Proxy 'Proxy name' is not Allocated to FBAD



Configuration Data Generation displays an error message, if the same FBAD diagram is allocated to multiple FF proxy and if the allocated channel is assigned to some other diagram variable.



As the Proxy object is not supported in CBM, online values are not displayed in the ports of FF proxy object.

Handling Cascaded Loop Back Signals

IO for FF handles the cascaded loop back FF signals in Function Diagrams as a single pin for the bidirectional connection if the signal name has the predefined suffixes **_FRWD** and **_BKWD** or **_frwd** and **_bkwd**.

Example

The cascaded signal **SIGNCASCA** has to follow the naming convention as **SIGNCASCA_FRWD** and **SIGNCASCA_BKWD** in FF Builder as shown in

Figure 123. While allocating the same FBAD to FF Proxy object in Function Diagram, it creates a single port with merged signal **SIGNCASCA** for the backward and forward as shown in Figure 124. If the naming convention is handled properly, then during the configuration data generation, the forward and backward signals is written to the corresponding channels of CI860.

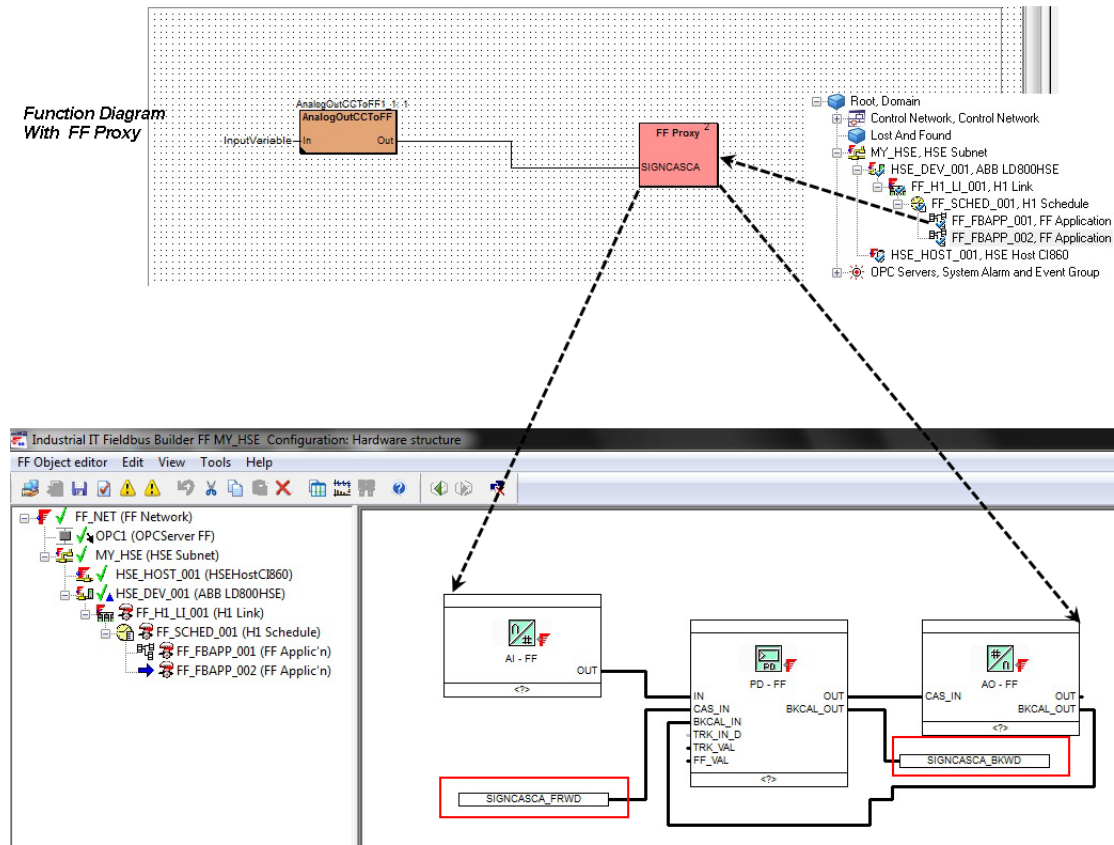


Figure 123. Handling Cascaded Loop Back Signals

If one signal is named with the suffix **_FRWD**, then it must have the counter signal name with the suffix **_BKWD**. Else, it will treat it as a separate signal instead of the merged signal.



If FBAD is modified in FF Builder, the FF Proxy Object allocated to the same FBAD needs to be reallocated and generated.

Copy/Paste on FF Proxy Object

Performing copy/paste deletes all the previous port information, and a new proxy template object instance is created. User has to reallocate the FBAD diagram for the new pasted object.

Open FF project in FF Builder

To open FF project in FF builder, right-click on FF Proxy Object and click **Open FF Application** menu. This opens the FF Builder with allocated FBAD diagram. After allocating the FF Proxy Object to the FBAD diagram, the FBAD diagram is opened in FF Builder through the **Open FF Application** option.

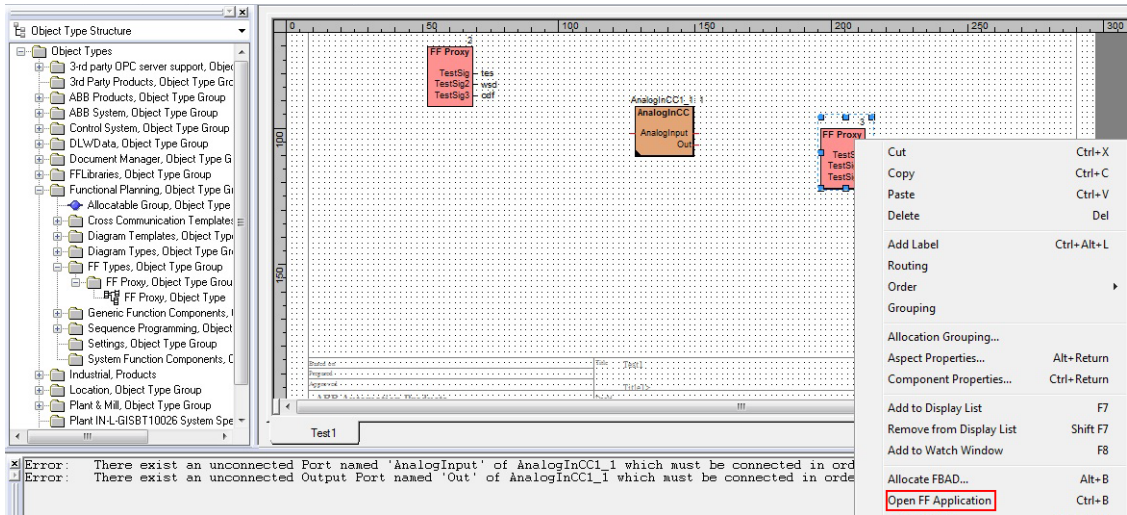


Figure 124. Open FF Application

Communication Interface

FOUNDATION Fieldbus devices can be referenced for Function Diagrams created prior to system version 5.1. The procedure is similar as described for HART devices in [Referencing Devices](#).

Differences to take care of when creating the references:

- The data types and IO variables are defined by Fieldbus Builder FF.

Error Messages

Function Diagram checks the Channel/HSE HOST/Direction/Data Type attributes of the FF Signal. These attributes are mandatory for Function Diagram to create the ports and generate the code. If any one of these attributes are missed during FF engineering, then FuD displays error while allocating the FBAD to the FF Proxy object.

[Table 22](#) shows the list of error messages displayed by FuD while allocating the FBAD to the FF Proxy object with possible solution.

Table 22. Errors with solution

Serial No.	Error Description	Solution
1	If the Channel is not assigned to the signal in FF Builder, FuD displays following error message: Channel is not allocated for some signals	Correct the FBAD Diagram and reallocate the FBAD.
2	If the HSE HOST/ Direction/ Data Type is not assigned to the signal in FF Builder, FuD displays following error message: Channel information is not found for some signals	Correct the FBAD diagram and reallocate the FBAD.

Table 22. Errors with solution

Serial No.	Error Description	Solution
3	If the allocated FBAD do not have any FF signal, FuD displays following error message: Signal Information is not Found This may happen if the FBAD diagram is not saved properly after modification.	Correct the FBAD and reallocate the FBAD to the FF Proxy object.
4	If the XML received from the FBAD is not proper, FuD displays following error message: XML Parsing failed	Re save the FBAD diagram properly.
5	If any fault occurred during the processing of FBAD XML, FuD displays following error message: XML Error	Check if all the signal names and their attributes are saved properly.
6	If the selected FBAD diagram do not have any XML data, FuD displays following error message: Failed to get the FF information	Save the selected FBAD in FF builder properly.

PROFINET Devices

IO allocation for PROFINET allows the user to use IO allocation tool for the signal allocation of PROFINET devices.

Prerequisites

- 800xA system should be created and running.
- GSDML files for PROFINET devices required for installation should be available.



Install PROFINET devices manually, as Device Library Wizard does not support PROFINET devices.

Installing PROFINET devices manually

Steps:

1. Right-click on Hardware in **Control Builder M** to create a New Hardware Library.

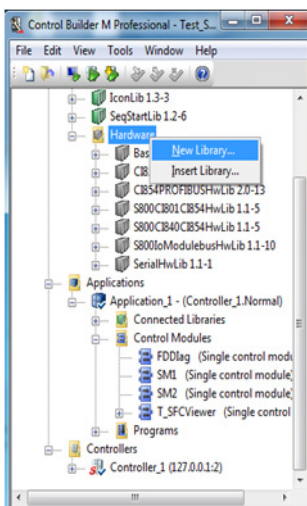


Figure 125. Create New Hardware Library

2. Assign a valid name to the library and insert the hardware types.

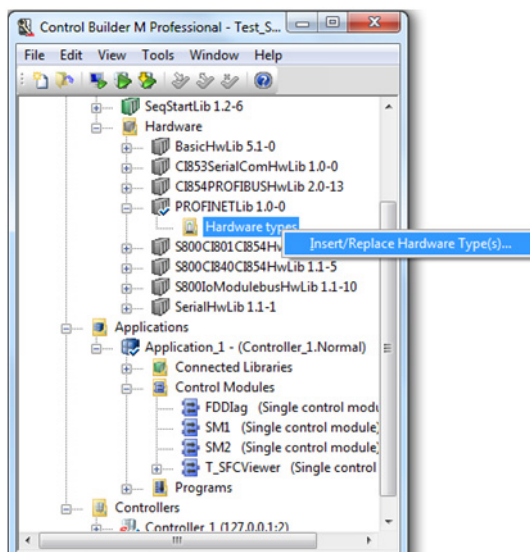


Figure 126. Insert Replace Hardware Type(s)

3. Browse and select the required PROFINET hardware definition file (*GSDML.xml*) and click **Open**.

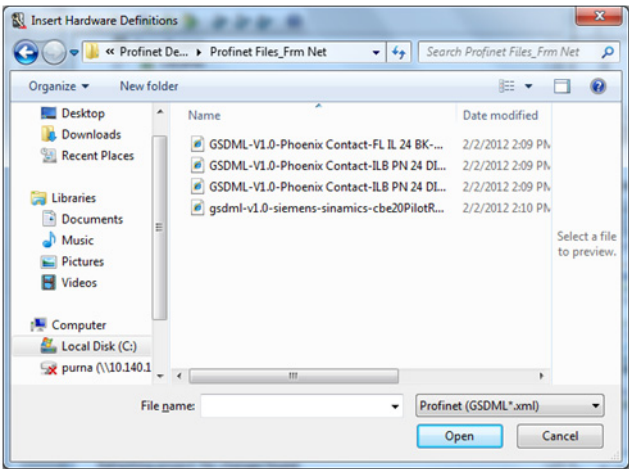


Figure 127. Selecting Profinet file

4. After installation, the device is visible under the library.

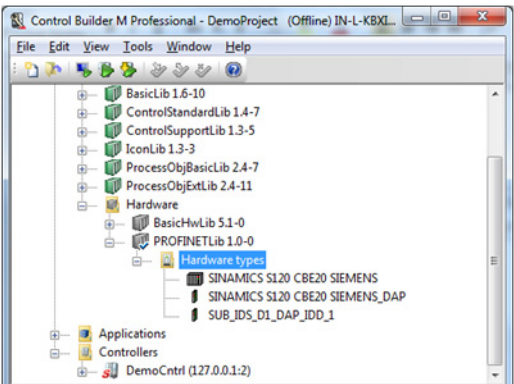


Figure 128. Device in Library

Preparing PROFINET IO Module

PROFINET specific device libraries cannot be installed through Device Library Wizard, refer [Installing PROFINET devices manually](#) on page 222 to extract the libraries in to the engineering workplace.

User has to prepare PROFINET specific device libraries using **HWDProcessor** to use them in IO allocation tool as well as Function Diagram for allocating or de-allocating signals to the channels of PROFINET IO modules.

For further information about preparation of PROFINET IO Modules see *System 800xA Engineering, Engineering Studio (3BDS011223*)*.

Building Functional Structure

Functional Structure is built by inserting the CBM signals in a Function Diagram. Connect them to other function components on the diagram as done to other CBM signal objects.

Allocate the Function Diagram to an application and Generate Configuration Data as explained in [Allocating Function Diagrams](#) on page 126.

Allocating Signals

After allocating the logic of the Function Diagrams to an application in the Control Structure, the IO signals (CBM_Signals) need to be allocated to the channels of PROFINET modules using IO Allocation tool. Follow the steps below to allocate the IO signals:

1. In Function Designer with an opened diagram click **Allocation > IO Allocation** from the menu bar.
The IO Allocation tool in the context of the opened diagram opens.
2. Right-click on the Boards object in the left pane of the IO Allocation tool and select **Insert Element**.
3. Add required PROFINET modules from the Control Structure of the opened structure browser.
4. Drag the signals from the right pane and drop them on to the boards object in the left pane to allocate the signals automatically.

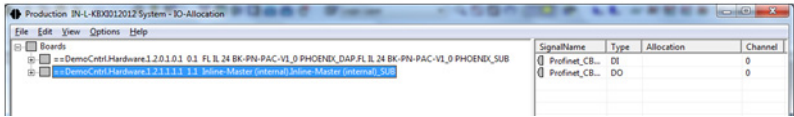


Figure 129. IO Allocation to Channels of PROFINET Modules

To write allocation information to Control Builder M Professional either perform Configuration Data Generation in Function Designer or right click on the Boards object in IO Allocation tool and select Write Allocation into CBM.

User can write allocation information to CBM through Plant Explorer. Follow the steps below to write to CBM through plant explorer:

1. Open the Plant Explorer and select **Control Structure** in the structure browser.
2. Select the required object, right-click and select **Advanced > Write Allocation** in to CBM.

IEC61850 Devices

Input output allocation for IEC61850 allocates the signals of the hardware type **CI868IEC61850HwLib**.

Prerequisites

- 800xA system should be created and running.
- Engineering Studio build should be installed.
- **ABB AC800M Signal Extension for IEC61850** should be loaded.

Create System Extension for IEC61850

ABB AC800M Signal Extension for IEC61850 is available in the **System Extension Load** of Configuration Wizard.

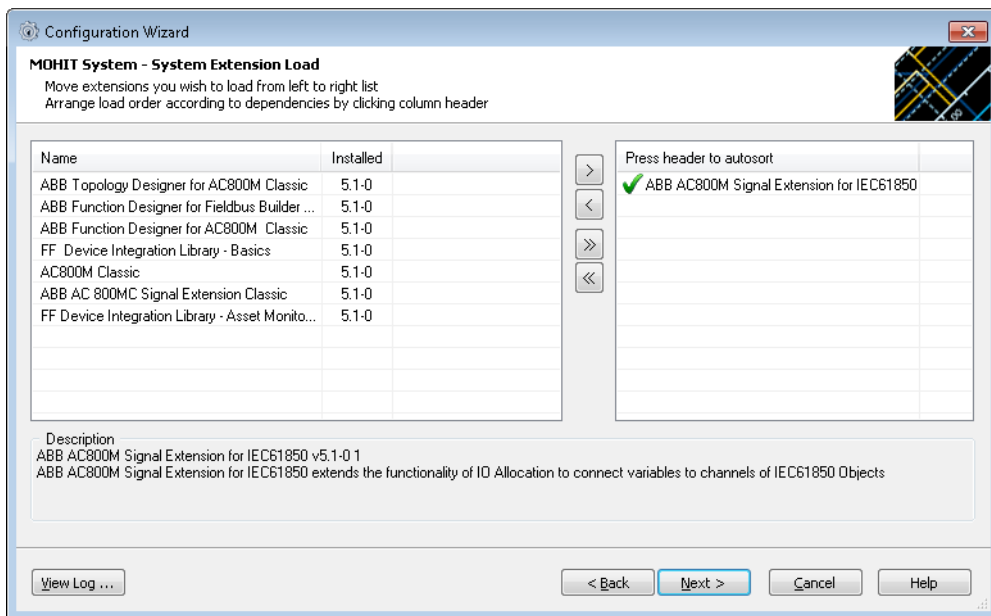


Figure 130. Loading System Extension



System extension has dependency on ABB Signal Extension for AC800M Connect.

Installing IEC61850 devices

Steps:

1. In Control Builder M, insert **CI868IEC61850Hwlib** at project level under the hardware folder and then insert **CI868IEC61850Hwlib** at controller level under the connected library folder.
2. Right-click on the hardware and click **Insert Unit**,

Insert Unit dialog box is displayed.

3. In Insert Unit dialog box, select CI868 and click **Insert**.

After insertion, CI868 is visible under the Control Structure.



CI868 can also be added from control structure.

Steps to insert objects

1. In control structure, right-click on Hardware and select **New Object**.

New Object dialog box is displayed.

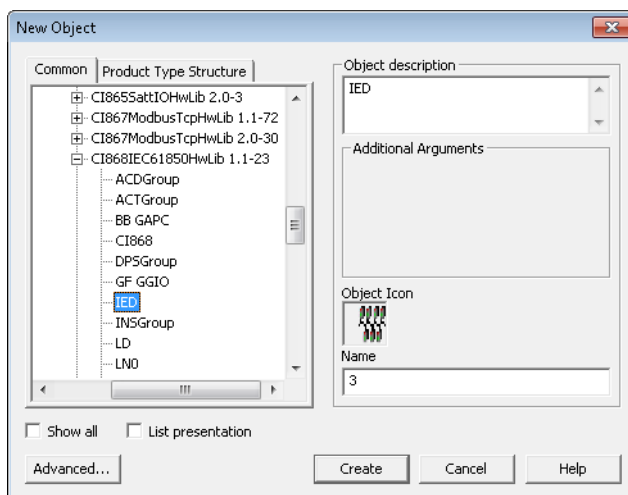


Figure 131. New Object

- Common | Product Type Structure
- Object Types
 - Control System
 - AC 800M/C Connect
 - CBM_Signals
 - Libraries
 - Hardware
 - ABBDrvNpbaC18
 - ABBDrvNpbaC18
 - ABBDrvRetaC18
 - ABBDrvRpbaC18
 - ABBDrvRpbaC18
 - ABBMNS5CI871
 - ABBPnI800C1851
 - ABBPnI800C1854

3. Under IED or MyIED add new modules.

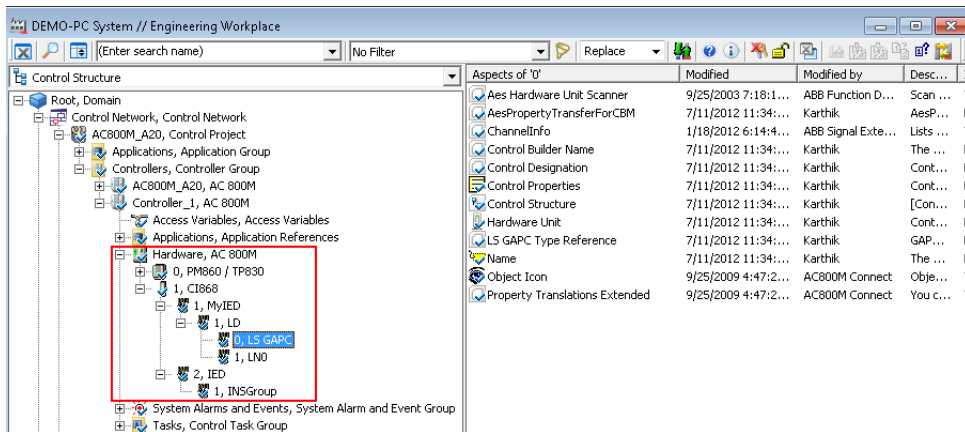


Figure 133. Modules

Allocating Signals

After allocating the Function Diagrams to an Application in the Control Structure, the IO signals (CBM_Signals) configured in Function Diagram is to be allocated to the channels of IEC61850 modules using IO Allocation tool

To allocate the IO signals:

1. In Function Designer with an opened diagram click **Allocation > IO Allocation** from the menu bar to open the IO Allocation tool in the context of the opened diagram.
2. Right-click on the Board's object in the left pane of the IO Allocation tool and select Insert Element.
3. Add required IEC61850 modules from the Control Structure of the opened structure browser.
4. Drag the signals from the right pane and drop them on to the Board's object in the left pane to allocate the signals.

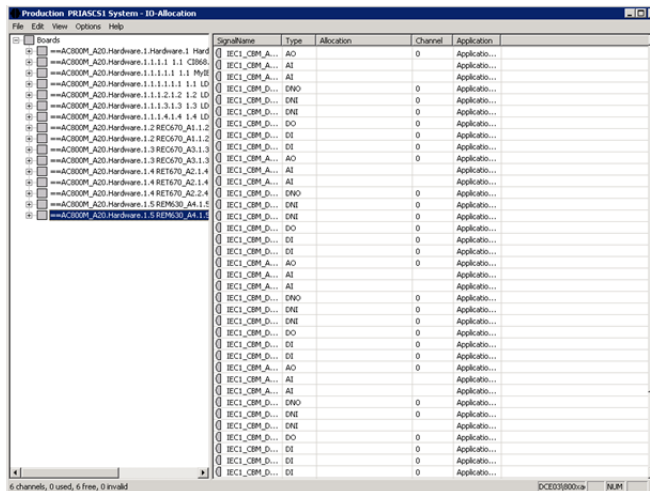


Figure 134. Signal Allocation



Avoid allocating non relevant signals for IEC1850 modules.

To write allocation information to Control Builder M Professional either generate configuration data in Function Designer or right-click on the Boards object in IO Allocation tool and select Write Allocation into CBM.

User can write allocation information to CBM through Plant Explorer also.

To write to CBM through plant explorer.

1. Open Plant Explorer and select Control Structure in the structure browser.
 2. Right-click the desired object and select Advanced > Write Allocation into CBM.
-

Bulk Data Manager Support and Bulk Operations

Bulk Data Manager is described in detail in *System 800xA Engineering, Engineering Studio (3BDS011223*)*. This chapter addresses some enhanced functionality useful when copying function diagrams or function components, or when setting property values via Bulk Data Manager.

Function Designer publishes its internal properties via the Function aspect, the Function Parameter aspect, and the Allocatable Group aspect. These include

- Internal properties like DiagramSaveTime, ComponentSaveTime, ObjectTypeSaveTime that are used internally to subscribe to changes and to update open diagrams.
- Diagram properties like DiagramIsBad and DiagramSize.
- Allocatable group (configuration data generation) properties like FDVERSIONGEN, GENERATED, ALLOCATED, IOALLOCATED, S_STATUS, S_DESCRIPTION, ModifiedBy, ModifiedTime. See [Diagram Status in Bulk Data Manager and in System Status Viewer](#).
- Component name properties like ObjectName, ObjectDescription, ControlBuilderName, ControlBuilderDescription.
- Component properties like AspectObject and NumberOfInputs.
- Component positioning properties like Position.Page or Position.X. See [Create Diagram\(s\) through Bulk Data Manager](#).
- Port parameter properties like In1, In1.Direction, In1.DataType, In1.initVal, In1.Inversion, and In1.Visible.
- Variable properties like VariableDataType and VariableAttributes.

All copied and inherited diagram properties are published via the diagram's Function aspect. All copied and inherited component and port properties including connect strings are published via the component's Function aspect - even if the data

is physically stored inside the Function Parameter aspect, see [Table 23](#).

Table 23. Properties get published via...

Property	... gets published via
Copied diagram property	<ul style="list-style-type: none"> Diagram's Function aspect Diagram's Function Parameter aspect
Inherited diagram property	<ul style="list-style-type: none"> Diagram's Function aspect
Copied component/port property	<ul style="list-style-type: none"> Component's Function aspect Component's Function Parameter aspect
Copied component/port property marked as "Publish as"	<ul style="list-style-type: none"> Parent diagram's Function aspect Parent diagram's Function Parameter aspect Component's Function aspect Component's Function Parameter aspect
Inherited component/port property	<ul style="list-style-type: none"> Component's Function aspect
Inherited component/port property marked as "Publish as"	<ul style="list-style-type: none"> Parent Diagram's Function aspect Component's Function aspect

The number and kind of published properties vary from Function aspect to Function aspect. A simple CBM_AIS e.g. publishes Value, Value.DataType, Value.Direction, while a control module publishes all its control module parameters.

Preparation of bulk data sheets to modify or copy complete Function diagrams can be a time consuming task. It is necessary to add Function aspects from different components, and to filter out relevant properties which should be modified during copy, e.g. initial values for set points.

This is improved by the **Publish via Parent Diagram** setting, see [Figure 10](#). All component and port properties marked as "**Publish as**" are additionally published through the parent diagram's Function aspect. Thus user can quickly define the most important properties/parameter of a diagram (template), add the diagram's Function

aspect to a bulk data sheet, and copy them through Bulk Data Manager while setting individual properties/parameter, see also the following example:



Copy/paste or move Function Diagrams between different projects is not supported through Bulk Data Manger.



When data is populated in the BDM sheet from 800xA System, if data in any cell is not in the required format, change the format accordingly.

Copy Diagram(s) via Bulk Data Manager

Assume you configure a simple diagram **MyTank** that measures some level and indicates that a specific level has reached. The parameters **Level** and **Start Delay** of function block **LevelDetection1** are marked as **Publish via Parent Diagram**, see [Figure 135](#). These values shall be set individually while copying the diagram using Bulk Data Manager (BDM).

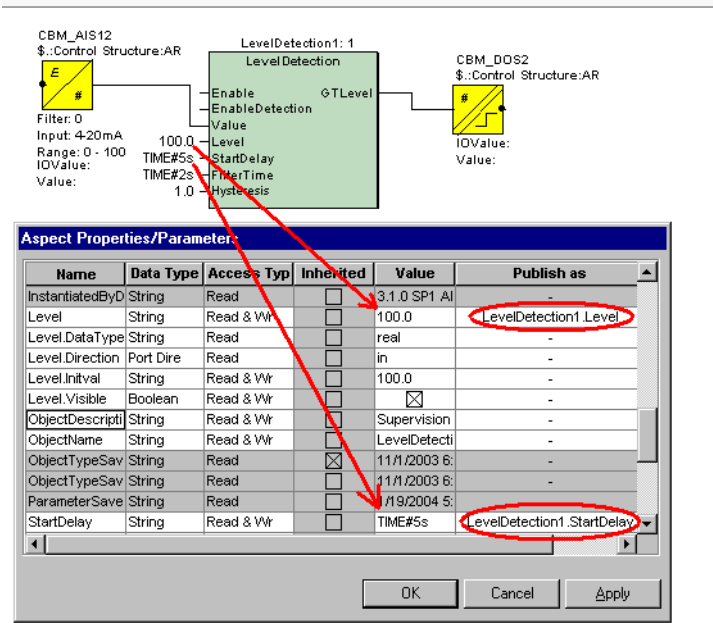


Figure 135. Example Diagram MyTank to be Copied via Bulk Data Manager

1. Right-click on any background inside Function diagram **MyTank**, and invoke the Engineering Workplace command **Advanced > Bulk Data Manager**.
2. Use BDM options 'Subtree enabled' and 'Transaction by object' as in [Figure 139](#).
3. Drag&Drop the Function aspect of MyTank into an empty bulk data sheet.
4. BDM displays the Configure Properties dialog. As the **LevelDetection1** parameter **Level** and **StartDelay** were marked as **Publish via Parent Diagram**, they are listed as diagram properties **LevelDetection1.Level** and **LevelDetection1.StartDelay**, see [Figure 136](#). Check them in order to modify them while copying diagram(s). They get displayed in bulk data sheet row 1.
5. Drag&Drop **MyTank** into the bulk data sheet. The current values are echoed in bulk data sheet row 2.

- 6. In BDM copy row 2 into row 3(4,5...), set **MyTank** as source object, set individual parameter values, and invoke BDM command **Save (all Objects)**.

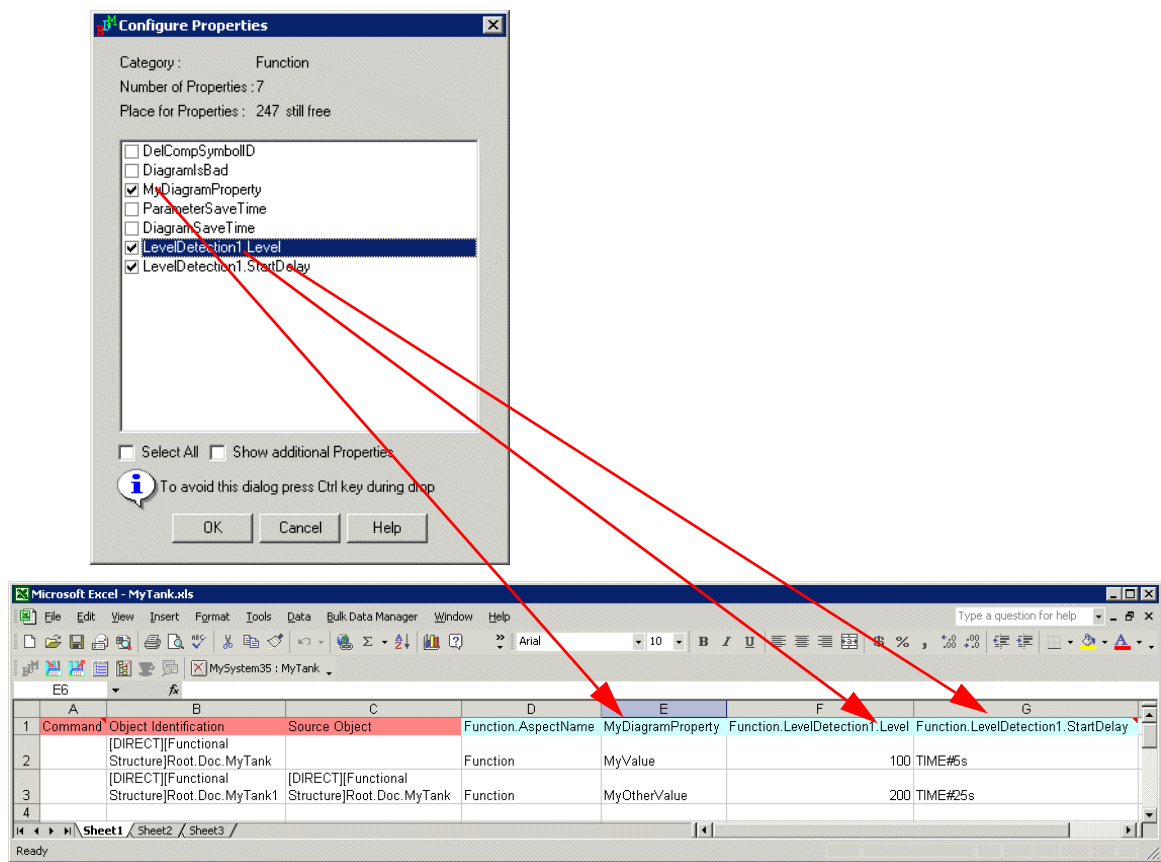


Figure 136. Bulk Copy Diagram(s) with Properties Published via Parent Diagram

Copy and Rename Loop Diagram(s) via Bulk Data Manager

Assume you need to bulk copy and rename a loop control, see example [Figure 137](#). Mind that components and the ‘Name’ parameter of control modules take the diagram name ‘Typ1’ as prefix.

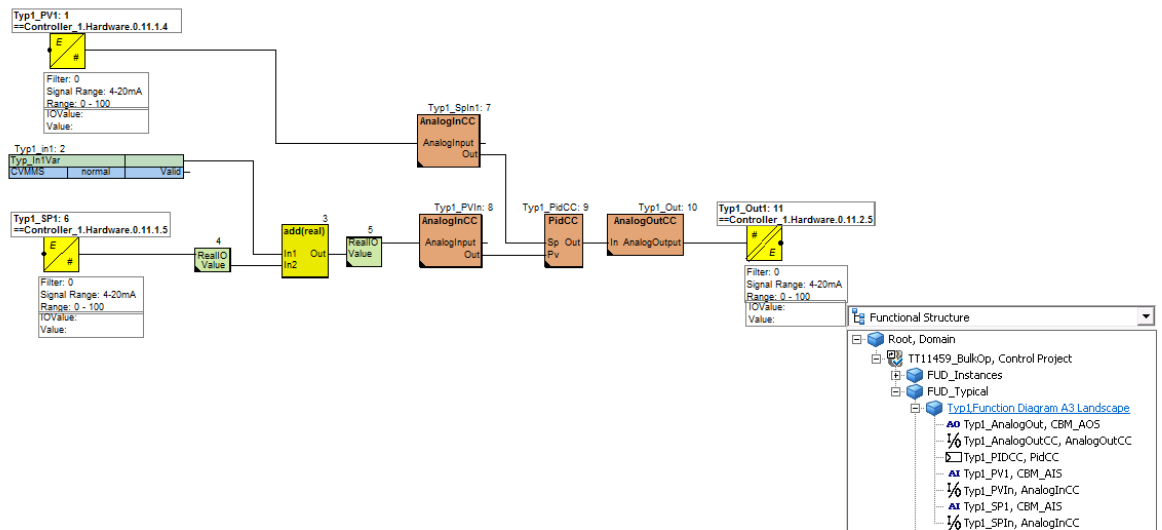


Figure 137. Example Loop Control

The BDM sheet in [Figure 138](#) copies and renames the loops F001\LC1...LC5 and F002\LC6...LC10 in a single row. F001 and F002 are two folder objects in Functional Structure that are used later for allocation. The BDM sheet consists of two columns only:

- Object Identification (F001\LC1...LC5 and F002\LC6...LC10)
- Source Object (Typ1)

The following BDM options are used, see [Figure 139](#):

- Subtree enabled' (Otherwise diagram components will not be copied.)

- Transaction ‘By Object’.



Make sure **NOT** to use Transaction ‘Data Area’. ‘Data Area’ is actually not supported by CB/Function Designer. If you do so, you might get inconsistencies in your data in Functional Structure and in Control Structure.

In the copied loop diagrams ‘LCx’ the components and the ‘Name’ parameter of control modules are automatically renamed, see [Figure 140](#). Same holds for the Control Builder Name used inside Control Builder, see [Figure 141](#).

	A	B	C
1	Command	Object Identification	Source Object
2		[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Instances.F001.LC1	[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Typical.Type1
3		[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Instances.F001.LC2	[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Typical.Type1
4		[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Instances.F001.LC3	[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Typical.Type1
5		[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Instances.F001.LC4	[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Typical.Type1
6		[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Instances.F001.LC5	[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Typical.Type1
7		[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Instances.F002.LC6	[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Typical.Type1
8		[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Instances.F002.LC7	[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Typical.Type1
9		[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Instances.F002.LC8	[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Typical.Type1
10		[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Instances.F002.LC9	[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Typical.Type1
11		[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Instances.F002.LC10	[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Typical.Type1
12			

Figure 138. Bulk Copy and Rename Loop Diagram(s)

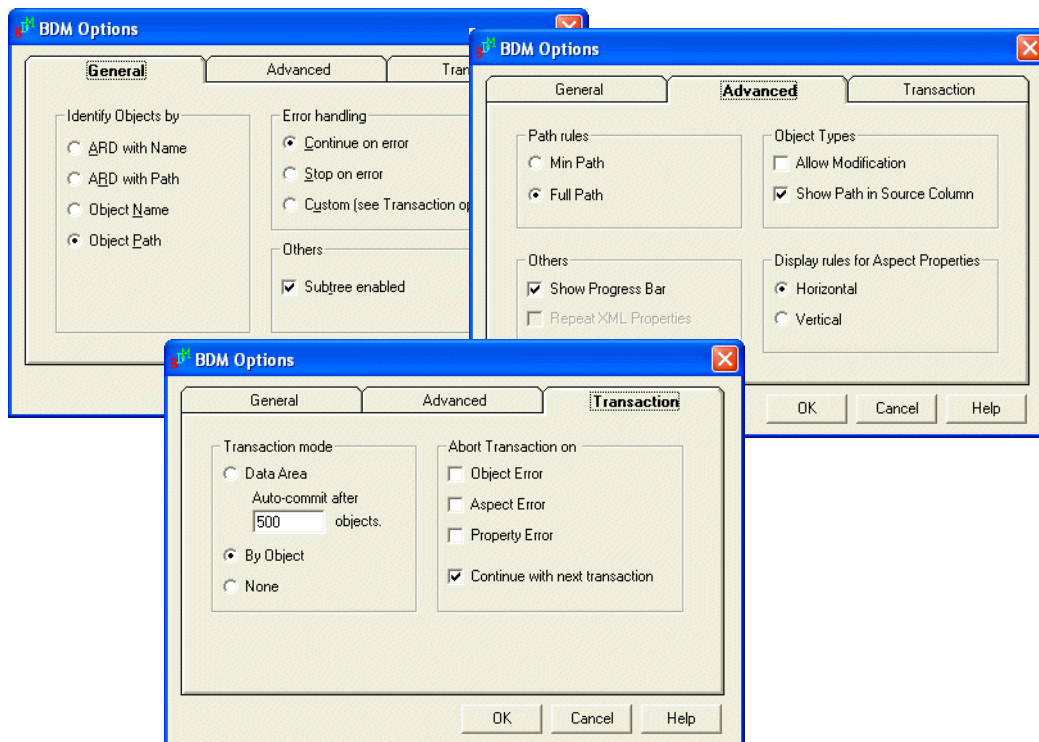


Figure 139. BDM Options to Copy Diagram(s)

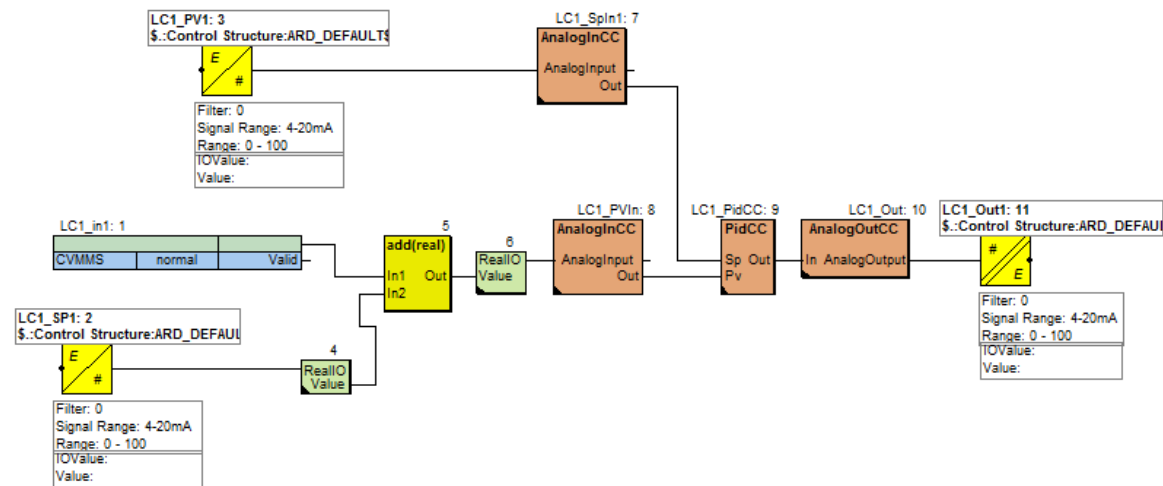


Figure 140. Copied Loop Diagram LC1

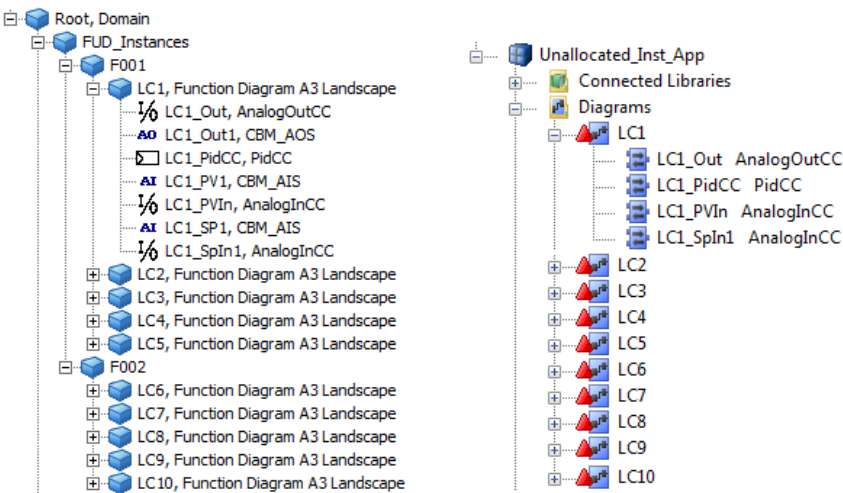


Figure 141. Copied Loop Diagram LC1 in Functional Structure and in Control Builder

Working with Diagram References (Symbol Objects) Using Bulk Data Manager

Diagram references and diagram variables are supported as symbol objects.

Diagram references created through Bulk Data Manager must be converted from aspect object to symbol object. Upgrading aspect object to symbol object is described in *System 800xA, Upgrade (3BSE036342*)* manual.

Two engineering templates named **BDM_DiagramRef_Var_Basic** and **BDM_DiagramRef_Var_Advanced** are used to modify diagram references and diagram variables represented as symbol objects.

These templates display the details of each diagram reference such as reference path, reference type (input / output), reference name, reference description, variable name, variable description, variable data type, variable init value, variable attributes, can create communication variable, on copy diagram reference, source count, sink count, and MMS count in a single row.

User can perform the following operations using these templates by editing the respective fields:

- Connecting and disconnecting diagram variables.
- Modification of description of diagram variables.
- Modification of name and description of diagram references.
- Modification of datatype, initial value, and attributes of diagram variables (possible only through **BDM_DiagramRef_Var_Advanced** template).
- Modification of Communication Variable.
- Modification of On Copy Diagram Reference.

The above operations can also be performed when the user drags and drops the Function Diagram References aspect on to an ordinary BDM activated excel sheet. Select the required attributes from the navigation tree that is displayed.

Follow the steps to connect the diagram references in bulk:

1. Open **BDM_DiagramRef_Var_Advanced** engineering template.
2. Drag and drop the required object(s) having the details of diagram references and variables that need to be modified, into the template (see [Figure 142](#)).



Do not modify the values in the read only columns (columns with pink color header) of the template.

3. Enter / edit the reference name, reference description, variable name, variable description, variable datatype, variable init value, variable attributes, communication variable, and on copy diagram reference as required (see [Figure 143](#)).



An error message is displayed, when the user provides invalid entries in the variable datatype (e.g. bool, real, BoolIO, RealIO, int) and variable attribute (e.g. retain, coldretain, nosort, constant, hidden, state) columns.

4. Click **Save All Objects**.

Read only area

Col	Object Id	Ref Obj Path	Ref Type	Reference Name	Ref Desc	Variable Na	Var De	Var Dataty	Var Init Vali	Var Attribut
Filter:										
	[DIRECT]	[Fun Root/MyTank/F0	Input	LC11_In1	Typ2 In1 reference					
XML E	[DIRECT]	[Fun Root/MyTank/F0	Input	LC12_In1	Typ2 In1 reference					
XML E	[DIRECT]	[Fun Root/MyTank/F0	Input	LC13_In1	Typ2 In1 reference					
XML E	[DIRECT]	[Fun Root/MyTank/F0	Output	Out2_Ref1	Out2_Var description					
XML E	[DIRECT]	[Fun Root/MyTank/F0	Output	Out3_Ref1	Out3_Var description					

Figure 142. Diagram References Without Connected Diagram Variables

Col	Object Id	Ref Obj Path	Ref Type	Reference Name	Ref Desc	Variable Name	Var Desc	Var Datatype	Var Init Value	Var Attribute
Filter:										
	[DIRECT]	[Fun Root/MyTank/F0	Input	LC11_In1	Typ2 In1 refer	Out2_Var1		bool		retain
XML E	[DIRECT]	[Fun Root/MyTank/F0	Input	LC12_In1	Typ2 In1 refer	Out3_Var1		bool		retain
XML E	[DIRECT]	[Fun Root/MyTank/F0	Input	LC13_In1	Typ2 In1 refer	Out3_Var1		bool		retain
XML E	[DIRECT]	[Fun Root/MyTank/F0	Output	Out2_Ref1	Out2_Var des	Out2_Var1		bool		retain
XML E	[DIRECT]	[Fun Root/MyTank/F0	Output	Out3_Ref1	Out3_Var des	Out3_Var1		bool		retain

Figure 143. Diagram References with Connected Diagram Variables

In the entered diagram variable names, the existing diagram variables get connected to the respective diagram references. The new diagram variables are created and get connected to the respective diagram references.

All the changes made in the descriptions / value / data type / attributes are also incorporated.

Bulk Allocate Diagrams

The diagrams copied and connected in the bulk examples above are placed in Control Structure in the Unallocated Application, see [Figure 144](#). In order to allocate them to different control applications, they could be moved individually in Control Structure.

Because we have located them in folders/groups F001, F002 in Functional Structure, another approach is taken here: Allocate the folders/groups including the child diagrams in a single step:

- F001 and its child diagrams LC1...LC5 shall be allocated to Application_1
- F002 and its child diagrams LC6...LC10 shall be allocated to Application_2.

The bulk data sheets in [Figure 145](#) and [Figure 147](#) first add a Function, an Allocatable Group, and a Diagram aspect to the folders/groups F001 and F002. In a 2nd step the folders/groups are moved to the corresponding applications.

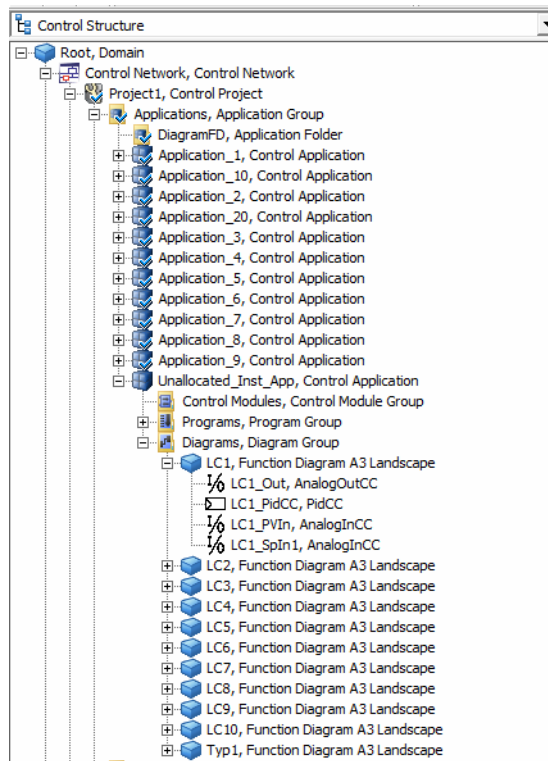


Figure 144. Unallocated Diagrams LC1...LC10

A	B	C	D	E	F
1	Command	Object Identification	Source Object	Function/Function Group	Allocatable Group/Single Control Module/Single Control Module AspectName
2		[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Instances.F001	Function	Allocatable Group	Single Control Module
3		[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Instances.F002	Function	Allocatable Group	Single Control Module
4					
5					
6					
7					
8					

Ready

Figure 145. Bulk Allocate, Step1: Create Allocation Aspects

After step1, the folders/groups F001, F002 are also placed in the unallocated application.

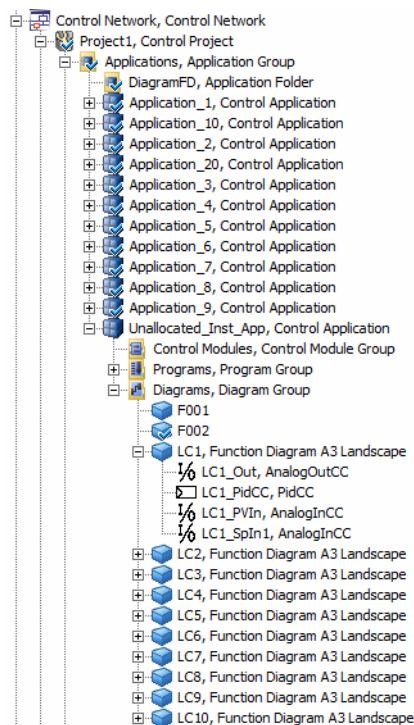


Figure 146. Unallocated Diagrams F001, F002, LC1...LC10 After Step1

	A	B	C	D	E	F	G	H
1	Command	Object Identification	Source Object	Basic Property Structure/Control Structure.ParentObject				
2		[DIRECT][Control Structure]Root.Control Network.TT11459_BulkOp.Applications. Unallocated_Inst_App.Control Modules.F001		[DIRECT][Control Structure]Root.Control Network.TT11459_BulkOp.Applications. Application_1.Control Modules				
3		[DIRECT][Control Structure]Root.Control Network.TT11459_BulkOp.Applications. Unallocated_Inst_App.Control Modules.F002		[DIRECT][Control Structure]Root.Control Network.TT11459_BulkOp.Applications. Application_2.Control Modules				
4								

Figure 147. Bulk Allocate, Step2: Move Folders/Groups to Applications.

After step2, the folder/group F001 is allocated to Application_1, and F002 is allocated to Application_2. The child diagrams LC1...LC10 are still located in the unallocated application, they get moved to the correct application when generating configuration data for the parent folders/groups F001, F002, see [Bulk Generate Configuration Data](#).

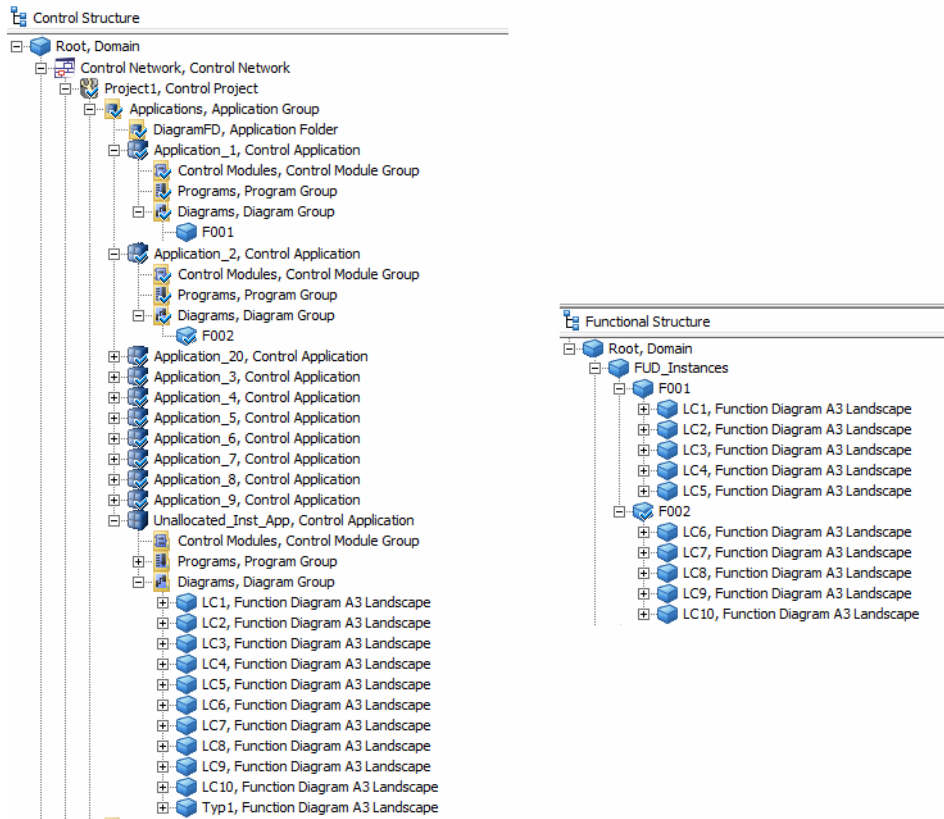


Figure 148. Allocated Diagrams F001, F002 After Step2

Bulk Generate Configuration Data

Configuration data for the diagrams in the bulk examples above can be bulk generated by the bulk data sheet in [Figure 149](#). Multi select the folders/groups and invoke allocatable group aspect command **Generate Configuration Data**. Alternatively, aspect command **Generate Configuration Data (Full Build)** can be

invoked. Full build always generates all diagrams / groups including their children. Without full build, only modified diagrams/groups get generated.



It is recommended not to generate more than 2500 function diagrams at one shot through Bulk Data Manager.

These aspect commands can also be invoked from the Allocatable Group aspect in the Plant Explorer. E.g. select **FuD_Instances\F002.Allocatable Group** aspect and invoke command **Generate Configuration Data (Full Build)**.

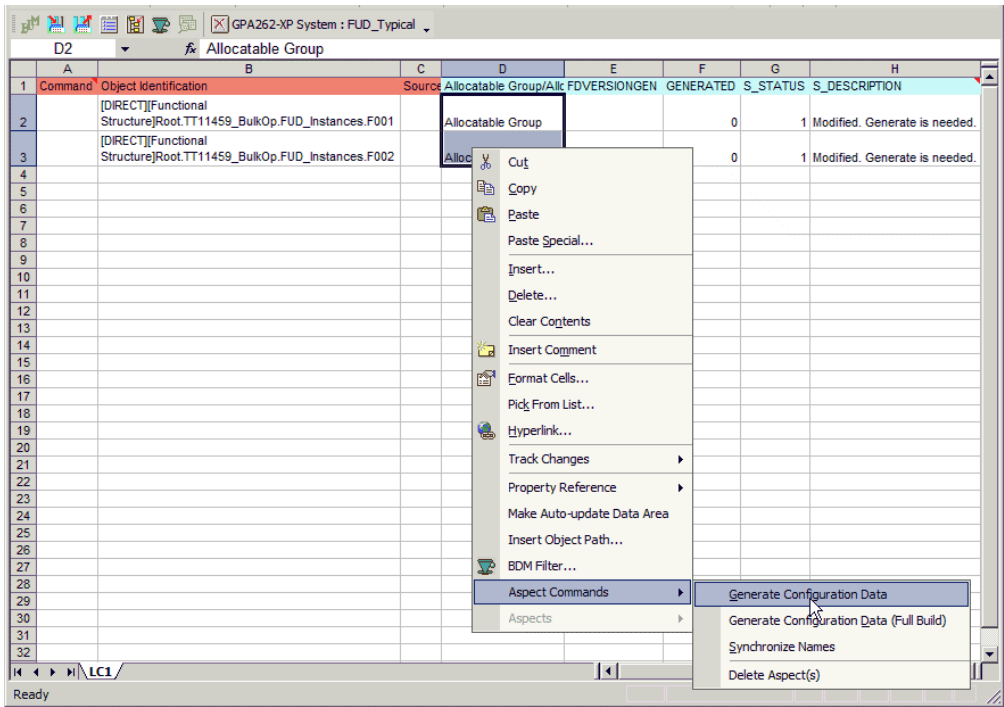


Figure 149. Bulk Generate Configuration Data

In either case, a working dialog indicating the progress and a Cancel button is displayed. For each generated diagram/group, errors and warnings are written to the AES Error Context aspect. Successful configuration data generation can also be verified by checking the diagram status, see [Diagram Status in Bulk Data Manager](#)

and in System Status Viewer.

During configuration data generation, the child diagrams LC1...LC10 and Out1, Out2 are to the correct application, see [Figure 150](#).

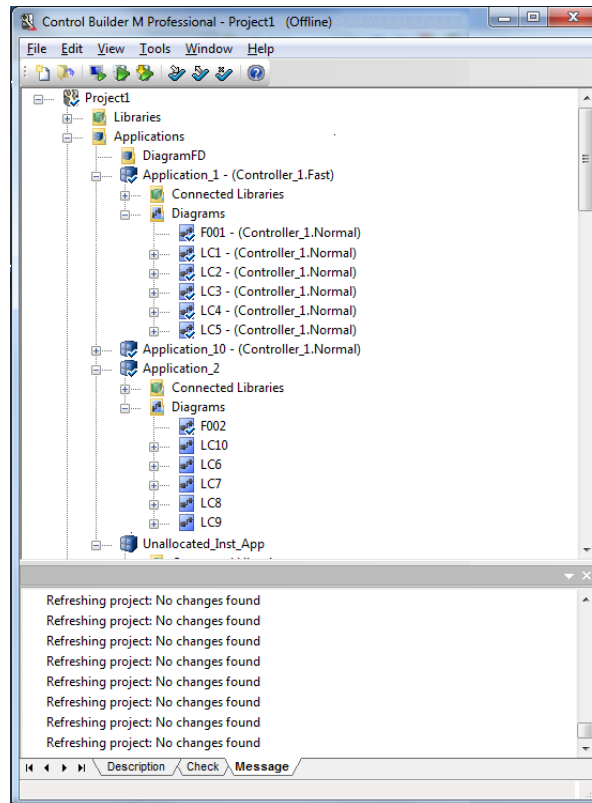


Figure 150. Allocated and Generated Diagrams

Diagram Status in Bulk Data Manager and in System Status Viewer

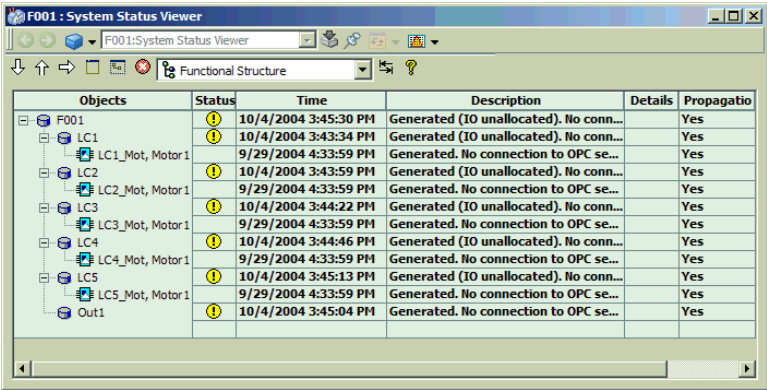
The diagram status can be checked

- by the traffic light status bar indication for each diagram, see [Instance Specific Initial Values](#).

- by the bulk data sheet in [Figure 151](#). The allocatable group aspect properties
 - **GENERATED**
 - **S_STATUS**
 - **S_DESCRIPTION**indicate successful configuration data generation.
- in System Status Viewer, see [Figure 152](#) and [Instance Specific Initial Values](#). Add a System Status Viewer aspect to a folder/group or to any parent object in Functional Structure or in Control Structure. The System Status Viewer displays the status of all child diagrams/groups.

	A	B	C	D	E	F	G	H
	Command	Object Identification	Source	Allocatable Group/Alt	FDVERSION/GEN	GENERATED	S_STATUS	S_DESCRIPTION
1		[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Instances.F001		Allocatable Group	2004-10-04-15:45:30	1	2	Generated (IO unallocated). No connection to OPC server.
2		[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Instances.F001.LC1		Allocatable Group	2004-10-04-15:43:34	1	2	Generated (IO unallocated). No connection to OPC server.
3		[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Instances.F001.LC2		Allocatable Group	2004-10-04-15:43:59	1	2	Generated (IO unallocated). No connection to OPC server.
4		[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Instances.F001.LC3		Allocatable Group	2004-10-04-15:44:22	1	2	Generated (IO unallocated). No connection to OPC server.
5		[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Instances.F001.LC4		Allocatable Group	2004-10-04-15:44:46	1	2	Generated (IO unallocated). No connection to OPC server.
6		[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Instances.F001.LC5		Allocatable Group	2004-10-04-15:45:13	1	2	Generated (IO unallocated). No connection to OPC server.
7		[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Instances.F001.Out1		Allocatable Group	2004-10-04-15:45:04	1	2	Generated. No connection to OPC server.
8								
9								

Figure 151. Check Diagram Status in BDM



The screenshot shows the 'F001: System Status Viewer' window. On the left is a tree view of the functional structure with objects F001, LC1, LC2, LC3, LC4, LC5, and Out1. On the right is a table with columns: Objects, Status, Time, Description, Details, and Propagation. The table lists the status of each object, with most showing a yellow warning icon and a message about unallocated IO or no connection to OPC.

Objects	Status	Time	Description	Details	Propagation
F001	⚠	10/4/2004 3:45:30 PM	Generated (IO unallocated). No conn...		Yes
LC1	⚠	10/4/2004 3:43:34 PM	Generated (IO unallocated). No conn...		Yes
LC1_Mot, Motor1	⚠	9/29/2004 4:33:59 PM	Generated. No connection to OPC se...		Yes
LC2	⚠	10/4/2004 3:43:59 PM	Generated (IO unallocated). No conn...		Yes
LC2_Mot, Motor1	⚠	9/29/2004 4:33:59 PM	Generated. No connection to OPC se...		Yes
LC3	⚠	10/4/2004 3:44:22 PM	Generated (IO unallocated). No conn...		Yes
LC3_Mot, Motor1	⚠	9/29/2004 4:33:59 PM	Generated. No connection to OPC se...		Yes
LC4	⚠	10/4/2004 3:44:46 PM	Generated (IO unallocated). No conn...		Yes
LC4_Mot, Motor1	⚠	9/29/2004 4:33:59 PM	Generated. No connection to OPC se...		Yes
LC5	⚠	10/4/2004 3:45:13 PM	Generated (IO unallocated). No conn...		Yes
LC5_Mot, Motor1	⚠	9/29/2004 4:33:59 PM	Generated. No connection to OPC se...		Yes
Out1	⚠	10/4/2004 3:45:04 PM	Generated. No connection to OPC se...		Yes

Figure 152. Check Diagram Status in System Status Viewer

Bulk Documentation and Contents

User can bulk print and create a contents diagram for each substructure in Functional Structure. E.g. to print and create a contents chapter for all diagrams, add a Function aspect to the Root object in Functional Structure and follow the steps below.

Following the bulk examples above, we print and create a contents chapter for the diagrams in folder/group F001:

- Open the Function aspect of F001.
- Invoke command **File > Create Contents** to create a contents diagram.
- In the **Create Contents dialog** select a template and check complete or overview contents, see [Figure 153](#).
 - Complete contents create detailed contents for each diagram and all its pages including functional designation (if any), name, description, page comment and page number. See [Figure 154](#).
 - Overview contents create contents for all diagrams according to the selected number of levels. The overview includes functional designation, name, description, and number of pages. Child diagrams are indented according to their level in Functional Structure.

- On OK, the contents diagram gets created according to the selected template and settings, see [Figure 154](#). The diagrams get listed according to their level in alphabetical order. In this example, the contents diagram lists the diagram name, its description, and page numbers.
- You can modify and save the created contents diagram, e.g. add some binder information and simple graphical components like lines and texts.
 - **File > Open Contents** command opens an already created contents diagram.
 - **File > Create Contents** command overwrites an already created one.

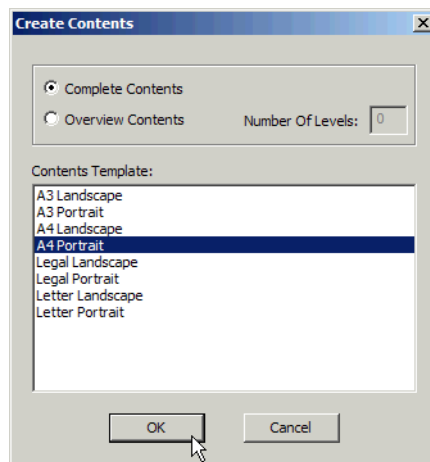


Figure 153. Create Contents Dialog

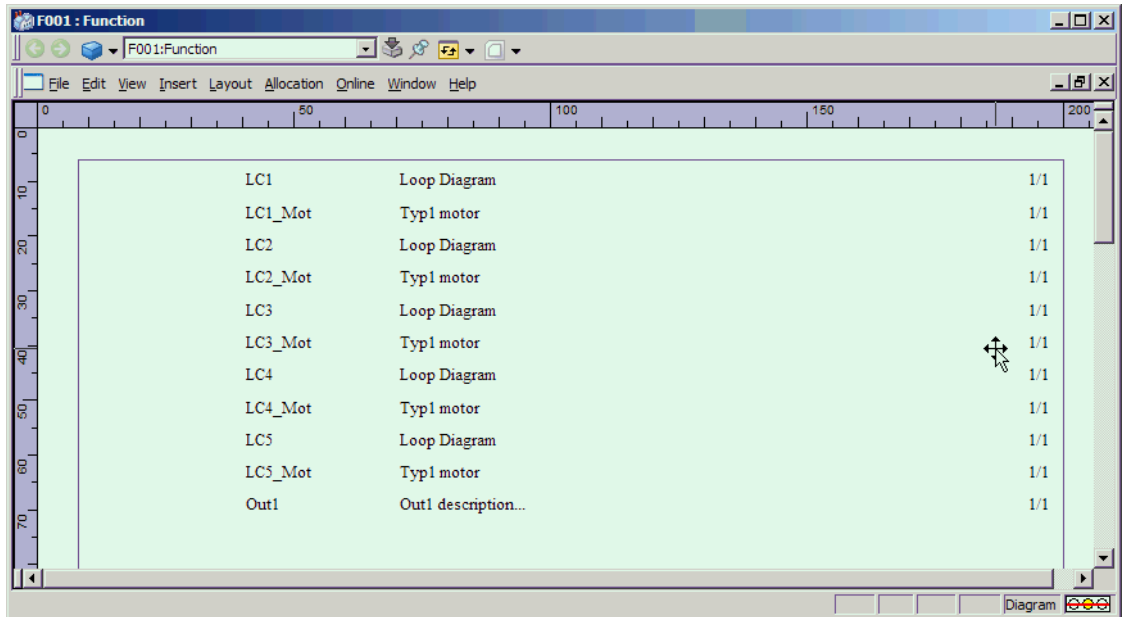



Figure 154. Complete Contents Diagram for Folder/Group F001

User can add individual page comments to the diagrams. E.g. open diagram F001\LC1 and invoke the page comment button  on the **Page** toolbar, or the **View > Pages > Page Comment** command.

- In the **Page Comment dialog** enter a page comment and select a range of pages to apply the comment to. See [Figure 155](#).

- Recreate the F001 contents diagram with page comments by the **File > Create Contents** command. See [Figure 156](#).

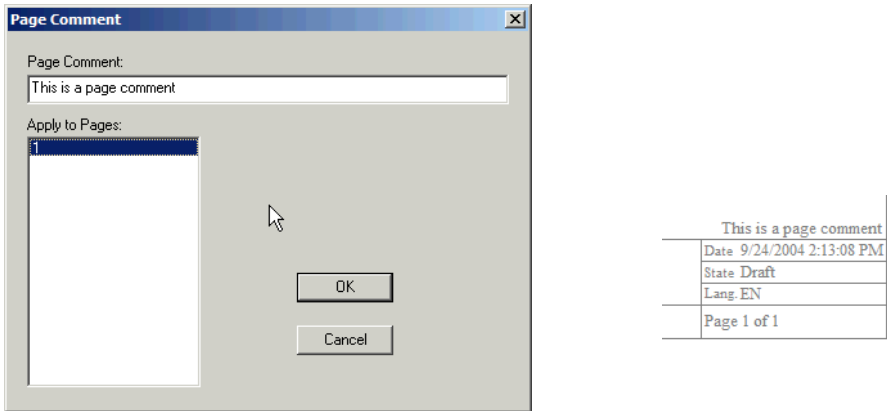


Figure 155. Page Comment Dialog and Page Comment in Diagram

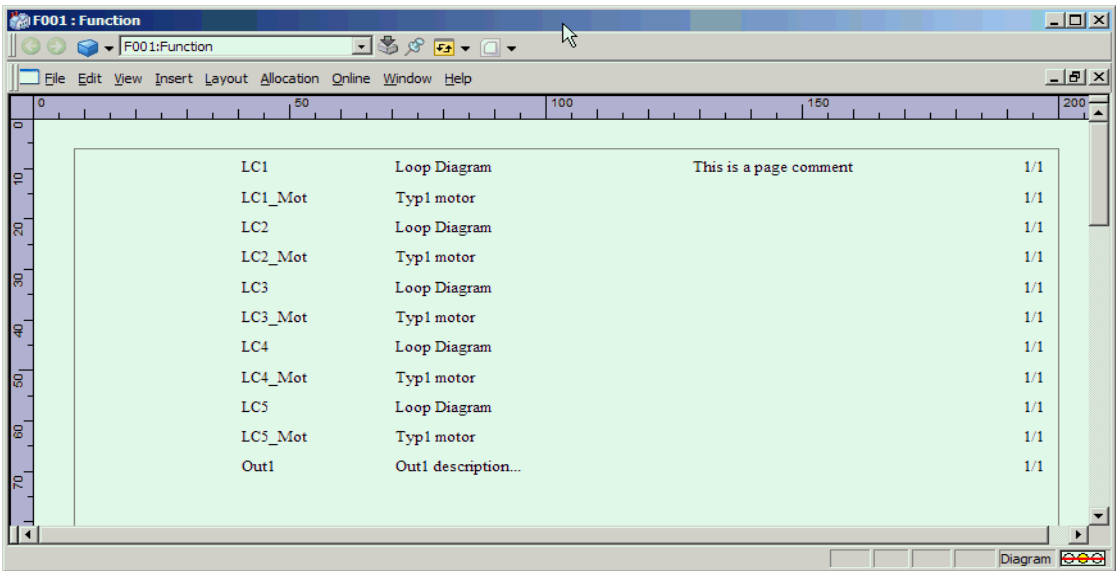


Figure 156. Complete Contents Diagram with Page Comment

User can also modify the contents template and adapt them to your needs. E.g. change fonts, distances, and references displayed. In **[Object Type Structure]Functional Planning\Diagram Templates\ContentsTemplates** select a corresponding contents template and open its **Function Diagram Contents Template** aspect, see [Figure 157](#). The templates installed by the Function Designer system extension define the following aspect property and document references:

- **Detailed contents:**
 - \$.:Functional Designation:Name\$
 - \$.:Name:Name\$
 - \$.:Name:Description\$
 - \$PgComment
 - \$Page/\$LastPage
- **Overview contents**
 - \$.:Functional Designation:Name\$
 - \$.:Name:Name\$
 - \$.:Name:Description\$
 - \$LastPage



Check the **component properties** of selected text elements in the contents templates. Component names prefixed by **DetailedContents...** define the complete/detailed contents, and component names prefixed by **OverviewContents...** define the overview contents.

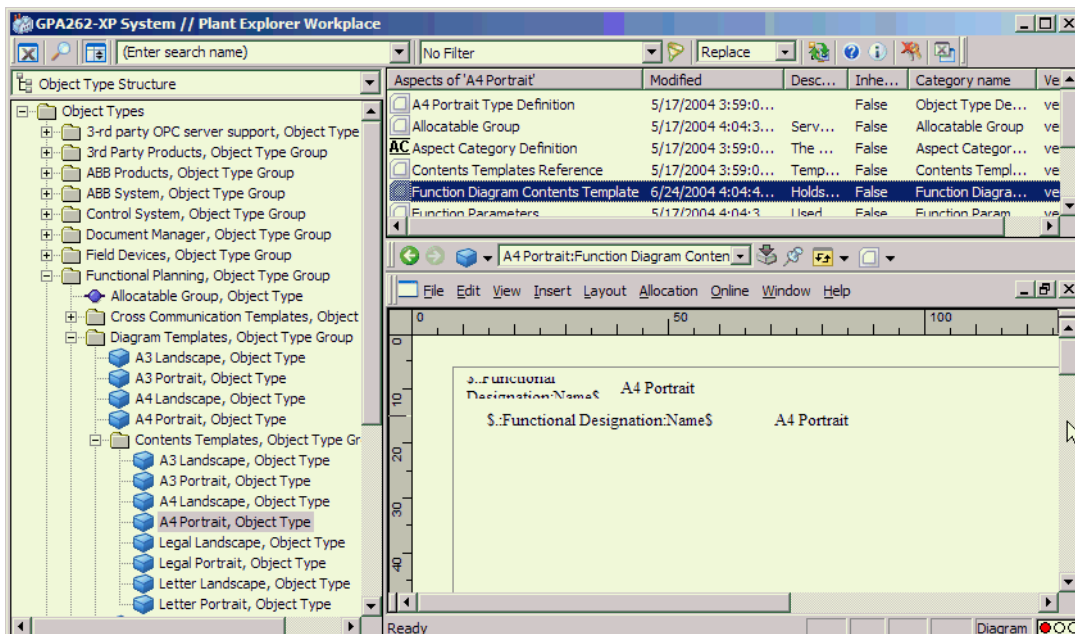


Figure 157. Contents Templates in Object Type Structure

In order to bulk print diagrams or a complete substructure of diagrams in Functional Structure, select the parent Function aspect.

1. Invoke the **File > Page Setup** command to open a standard Windows Page Setup Dialog. Enter a suitable printer page size. Typically, this printer page equals the diagram page size selected by the diagram template.
 - a. Invoke the **File > Template** or **Edit > Measurements and Size** command in order to check the diagram page size, see Figure 158.

- b. In cases that the diagram page size (e.g. A3 landscape) differs from the printer page size (e.g. A4 landscape), select Fit to sheet(s) in [Figure 158](#).

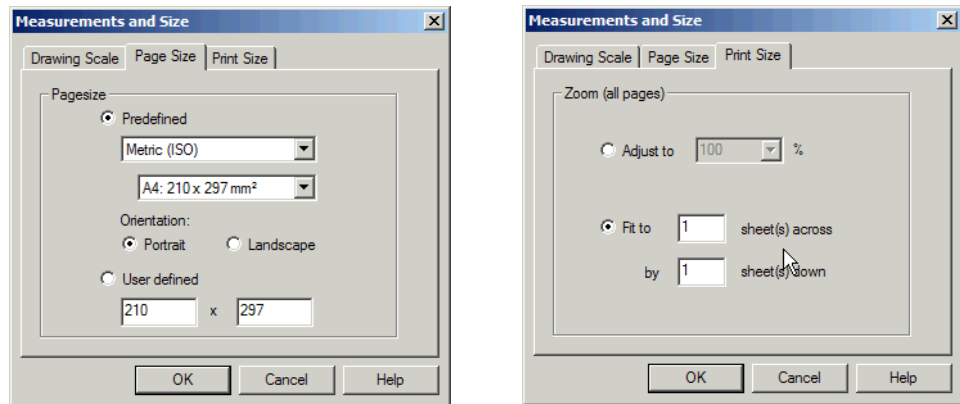


Figure 158. Edit > Measurements and Size Dialog

2. Invoke the **File > Print** command in order to run the bulk print job.
 - a. Check **Contents** to print the contents diagram.
 - b. Check **Diagrams [Child Diagrams]** to print diagrams including children.
 - c. Check **Instances of Diagram Types** to print diagrams of object types with their instance specific data.
 - d. Check required options for additional Port Documentation:
 - Hidden Ports with Modified Initial Values**
 - Connected Hidden Ports**
 - Visible Ports with Modified Initial Value**
 This additional documentation is generated in list form.
 - e. Check the **Printer and Printer Page Settings** to print the contents and diagrams with individual printer and printer page settings, or you can inherit and even save the settings from the parent Function aspect. In the latter case you can verify or change these settings by pressing the **Page Setup** button.
 - f. Check the **Print Components on Layers** options to print the diagrams as laid out on the screen, or in their original color. In the first case,

components on different layers get grayed or not printed at all according to the layer properties.

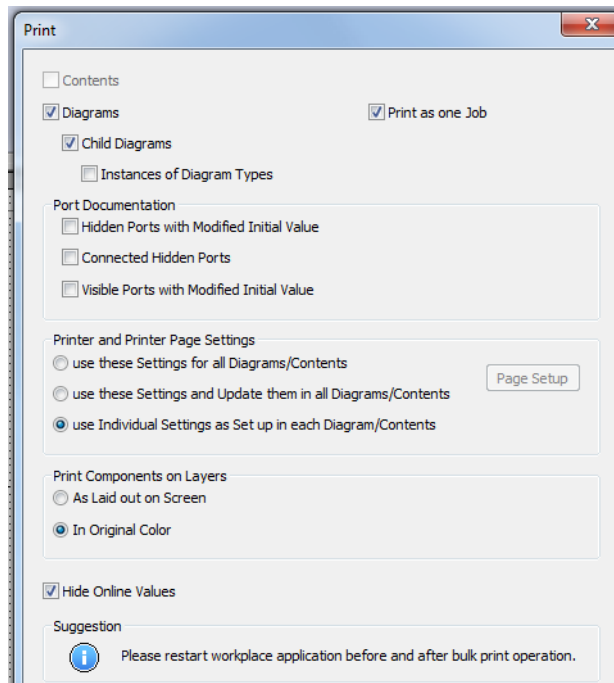


Figure 159. Function Designer Print Dialog (File > Print)

- g. Check the **Hide Online Values** option as required
3. Click OK. The standard Windows Print dialog gets displayed, see [Figure 242](#). Confirm or change the selected printer driver, number of copies, and option Print To File, and press OK again.
4. A print working dialog indicating the current printed diagram gets displayed. You can cancel the print job by pressing its **Cancel** button.
5. The printed diagrams are printed according to their level in alphabetical order. And they get listed in the output window of the parent Function aspect.

Create Diagram(s) through Bulk Data Manager

You can also bulk create diagrams via BDM from scratch. E.g. the bulk sheet in [Figure 160](#)

- creates a diagram F010\D1 from a diagram template type **Function Diagram A4 portrait**,
- adds components **D1_PV1, D1_PVIn, D1_PID**
- and positions them to the given positions and page numbers:
 - Position.Page sets the page number
 - Position.Y and Position.X set the top-left position of a component
 - in 0.01 mm for metric (ISO) diagram templates like A4, A3, and
 - in 0.001 inch for english (US) templates like letter, legal.



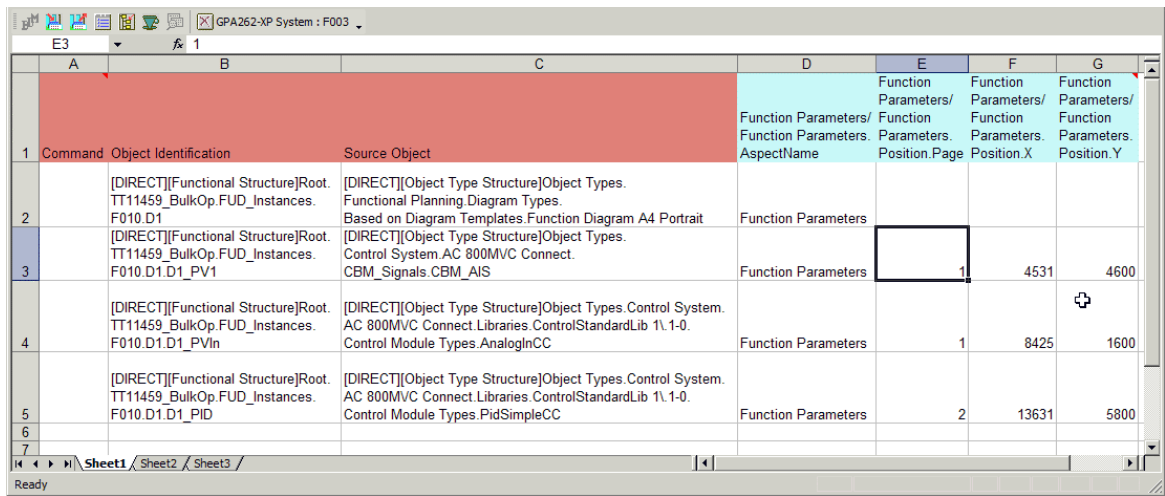
Perform the following while inserting Function Blocks or Control Modules in several pages of a Function Diagram using Bulk Data Manager:

- The Bulk Data Manager sheet must be sorted in ascending order with respect to the **Position.Page** values (shown in figure), if these values are provided in the Bulk Data Manager sheet.

Command	Object Identification	Source Object	Function/Function.AspectName	Function/Function.Position.Page
	[DIRECT]Functional Structure Root.testteset.asf asd100	[DIRECT]Object Type Structure Object Ty Function		1
	[DIRECT]Functional Structure Root.testteset.asf asd101	[DIRECT]Object Type Structure Object Types.Control System.AC 800MVC		1
	[DIRECT]Functional Structure Root.testteset.asf asd102	[DIRECT]Object Type Structure Object Types.Control System.AC 800MVC		2
	[DIRECT]Functional Structure Root.testteset.asf asd103	[DIRECT]Object Type Structure Object Types.Control System.AC 800MVC		2
	[DIRECT]Functional Structure Root.testteset.asf asd104	[DIRECT]Object Type Structure Object Types.Control System.AC 800MVC		3
	[DIRECT]Functional Structure Root.testteset.asf asd105	[DIRECT]Object Type Structure Object Types.Control System.AC 800MVC		3
	[DIRECT]Functional Structure Root.testteset.asf asd106	[DIRECT]Object Type Structure Object Types.Control System.AC 800MVC		3
	[DIRECT]Functional Structure Root.testteset.asf asd107	[DIRECT]Object Type Structure Object Types.Control System.AC 800MVC		4
	[DIRECT]Functional Structure Root.testteset.asf asd108	[DIRECT]Object Type Structure Object Types.Control System.AC 800MVC		4
	[DIRECT]Functional Structure Root.testteset.asf asd109	[DIRECT]Object Type Structure Object Types.Control System.AC 800MVC		4
	[DIRECT]Functional Structure Root.testteset.asf asd110	[DIRECT]Object Type Structure Object Types.Control System.AC 800MVC		5
	[DIRECT]Functional Structure Root.testteset.asf asd111	[DIRECT]Object Type Structure Object Types.Control System.AC 800MVC		5
	[DIRECT]Functional Structure Root.testteset.asf asd112	[DIRECT]Object Type Structure Object Types.Control System.AC 800MVC		5

- These components get instantiated as aspect object. By setting the aspect property **Function Parameters.Aspect Object** to 'false' user can bulk convert

them into symbol objects, see [Aspect Objects and Symbol Objects](#).



	A	B	C	D	E	F	G
1	Command	Object Identification	Source Object	Function Parameters/ Function Parameters. AspectName	Function Parameters/ Function Parameters. Position.Page	Function Parameters/ Function Parameters. Position.X	Function Parameters/ Function Parameters. Position.Y
2		[DIRECT][Functional Structure]Root. TT11459_BulkOp.FUD_Instances. F010.D1	[DIRECT][Object Type Structure]Object Types. Functional Planning Diagram Types. Based on Diagram Templates.Function Diagram A4 Portrait	Function Parameters			
3		[DIRECT][Functional Structure]Root. TT11459_BulkOp.FUD_Instances. F010.D1.D1_PV1	[DIRECT][Object Type Structure]Object Types. Control System.AC 800MVC Connect. CBM_Signals.CBM_AIS	Function Parameters	1	4531	4600
4		[DIRECT][Functional Structure]Root. TT11459_BulkOp.FUD_Instances. F010.D1.D1_PVIn	[DIRECT][Object Type Structure]Object Types.Control System. AC 800MVC Connect Libraries.ControlStandardLib 1\ 1-0. Control Module Types.AnalogInCC	Function Parameters	1	8425	1600
5		[DIRECT][Functional Structure]Root. TT11459_BulkOp.FUD_Instances. F010.D1.D1_PID	[DIRECT][Object Type Structure]Object Types.Control System. AC 800MVC Connect Libraries.ControlStandardLib 1\ 1-0. Control Module Types.PidSimpleCC	Function Parameters	2	13631	5800
6							
7							

Figure 160. Bulk Create Diagrams From Scratch

- User can also bulk connect components via BDM. E.g. bulk sheet [Figure 161](#)
 - connects D1_PV1.Value with D1_PVIn.AnalogInput via link1

- connects D1_PVIn.Out with D1_PID through link2.

	A	B	C	D	E	F	G
1	Command	Object Identification	Source Object	Function Parameters/Value	Function Parameters/AnalogInput	Function Parameters/Out	Function Parameters/PV
2		[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Instances.F010.D1	[DIRECT][Object Type Structure]Object Types.Functional Planning.Diagram Types.Based on Diagram Templates.Function Diagram A4 Portrait				
3		[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Instances.F010.D1.D1_PV1	[DIRECT][Object Type Structure]Object Types.Control System.AC 800MVC Connect.CBM_Signals.CBM_AIS	link1			
4		[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Instances.F010.D1.D1_PVIn	[DIRECT][Object Type Structure]Object Types.Control System.AC 800MVC Connect Libraries.ControlStandardLib 1\1-0.Control Module Types AnalogInCC		link1	link2	
5		[DIRECT][Functional Structure]Root.TT11459_BulkOp.FUD_Instances.F010.D1.D1_PID	[DIRECT][Object Type Structure]Object Types.Control System.AC 800MVC Connect Libraries.ControlStandardLib 1\1-0.Control Module Types PidSimpleCC				link2
6							
7							

Figure 161. Bulk Connect Components

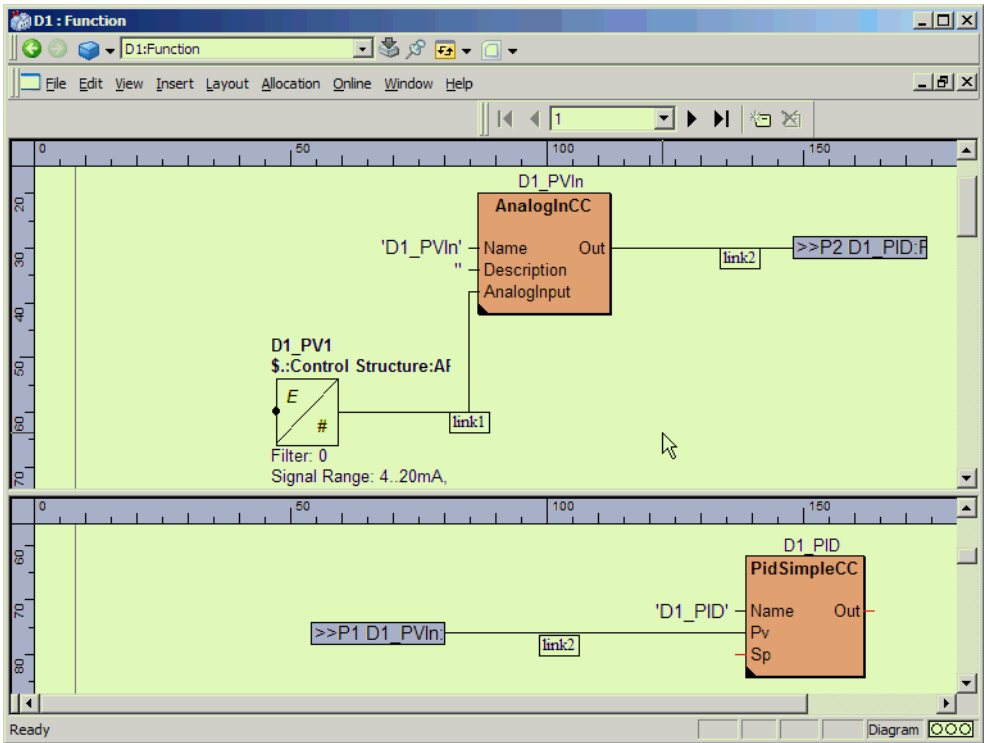


Figure 162. Diagram Created from Scratch with BDM

Import / Export (AFW-Files)

A Function Diagram defines the allocatable group which implement the entities for a Function Diagram object in the Functional Structure. They are a sub-entities to a Control Application entity. Importing / exporting a Function Diagram object is performed by the Import / Export tool.

A Function Diagram type object is implemented as a Control Module Type object and therefore also is an entity.

The Import / Export tool is described in *System 800xA, Administration and Security (3BSE037410*)*.

Import/export for control projects, control applications, control libraries, control module types etc. is described in *System 800xA Control, AC 800M Configuration (3BSE035980*)*.

Export / Import Function Diagram

A Function Diagram can be exported /imported either allocated or un-allocated. If the control application does not exist in the target system you are queried for a corresponding new placement. If the signals of the diagram are allocated they will loose the allocation to boards during export and have to be re-allocated in the target system.

The following procedure assumes that the dependencies of the Function Diagram are already fulfilled except for new placements to be done on the target system: All required system extensions are loaded, all user defined libraries and object types are available, the target Control Project is available and the needed libraries are connected to the target Control Application.

To export a single Function Diagram:

1. Launch the Import / Export tool from the toolbar of Plant Explorer.
2. Drag and drop the Function Diagram object into the left pane of the Contents tab of the Import / Export tool window.
3. Accept the check-marked contents shown in the Add Items dialog and press **OK**. By default export is done without dependencies.
4. In the toolbar press **Save**.

5. In the Save import/export-file dialog navigate to an appropriate folder, enter a file name and click **Save**.

The Function Diagram is now available as .afw archive file. Neither the parent objects nor the dependencies of the Function Diagram have been added.

To import a single, not yet existing Function Diagram:

1. Transfer (or give access to) the stored .afw file to the workstation where you want to do the import.
2. Make sure that the required system extensions and user-defined libraries are loaded.
3. Launch the Import / Export tool from toolbar of Plant Explorer.
4. In the toolbar press **Open**.
5. In the Open import/export-file dialog navigate to the stored .afw archive file and press **Open**.
6. If the parent object hierarchy given in the Functional Structure and in the Control Structure of the source system is not available in the Functional Structure and in the Control Structure of the target system: Provide corresponding substitutions by editing new placements in the Placements tab. Or rely on the Import Export tool querying you for new parent objects during import.
This includes placements in Control Structure that assign the Function Diagram to a new Control Project and a new Control Application.
7. In the toolbar press **Import All**.
8. A progress dialog pops up. Wait until an end message like "Import succeeded" is shown in the grid area of this dialog.



A warning message is given: "Restore Single Control Module <SCM name> succeeded with warnings. Reason: A single control module may only be imported as part of an entity."

This warning can be accepted as Generate Configuration Data command (see below) will incorporate the single control module representing the Function Diagram into the control application.

9. If no further error messages are shown in the grid area of the progress dialog: Press **Done** to exit.
10. To complete the imported diagram:
 - a. Re-connect the diagram references to appropriate diagram variables.
 - b. If not allocated: Allocate the diagram to an application.
 - c. (Re-)Allocate the signals to I/O boards.
 - d. Perform Write Allocation into CBM.
 - e. Perform Generate Configuration Data. This is required in any case.
 - f. Download the corresponding application to make this new diagram work in the controller.

If the Function Diagram object already exists in the target system you are queried for replacement of this object. If you press **Yes** the diagram will be replaced completely by the imported one.



Replacing the Function Diagram overwrites changes of the same diagram the in target system done in parallel to the changes in the engineering system.

Export / Import Application

A control application can be exported / imported with all Function Diagrams allocated to it.

The following procedure assumes that the dependencies of the control application and Function Diagrams are already fulfilled except for new placements to be done on the target system:

To export the control application:

1. In the Control Structure navigate to the control application and drag and drop it onto the Contents tab of the Import / Export tool.
Alternative:
In the Function Structure navigate to a Function Diagram allocated to the control application and drag and drop it onto the Contents tab of the Import / Export tool. Additionally add the dependency of this Function Diagram to the control application via the Add Items context menu to the contents to export.

2. Save the afw-file.

To import the control application:

1. Open the .afw file with the Import / Export tool
2. Edit placements in the Placements tab or rely on editing placements when query boxes on missing placements during import show.
If the project does not exist on the target system the application can be placed into another project.
3. In the toolbar press **Import All** and wait until an end message like “Import succeeded” is shown in the progress dialog.
4. If no error messages are shown in the grid area of the progress dialog: Press **Done** to exit.
5. To complete the diagrams of the imported application:
 - a. If needed: Re-connect the diagram references to appropriate diagram variables.
 - b. Allocate the signals to I/O boards.
 - c. Perform Write Allocation into CBM.
 - d. Perform Generate Configuration Data for all Function Diagrams. This is required in any case.
 - e. If needed: Change task connection of the imported application.
 - f. Download the application to make it work in the controller.

If the control application object already exists in the target system you are queried for replacement of this object. If you press **Yes** the application will be replaced completely by the imported one.



Replacing the control application overwrites changes of the same control application in the target system done in parallel to the changes in the engineering system.

Export / Import IO Allocation

A control application with several Function Diagrams can be exported / imported with keeping the allocation of signals to I/O boards. This requires to export / import the controller, which includes the corresponding I/O boards, too.

To export:

1. In the Control Structure navigate to the control application and drag and drop it onto the Contents tab of the Import / Export tool.
2. In the Control Structure navigate to the controller and drag and drop it onto the Contents tab of the Import / Export tool.
3. Save the afw-file.

You can also split the required contents into two files: One for the application and one for the controller.

To import:



Replacing the controller overwrites changes of the same controller in the target system done in parallel to the changes in the engineering system.



Replacing the control application overwrites changes of the same control application in the target system done in parallel to the changes in the engineering system.

1. Open the .afw file with the Import / Export tool.
2. If the control project does not exist: Edit placements in the Placements tab or rely on editing placements when query boxes on missing placements during import show.
3. In the entity view navigate to and right-click on the controller.
4. Click **Import** to import the controller.
5. If controller is already existing: Answer the replace queries with **Yes**.



A warning message is given: “Restore Application <application name> succeeded with warnings. Reason: An application may only be imported as part of an entity.”

This warning can be accepted as the import of the control application that follows will correct the situation.

6. If no error messages are shown in the grid area of the progress dialog: Press **Done** to exit.
7. In the entity view navigate to and right click on the control application.
8. Click **Import** to import the control application.
9. If the control application is already existing: Answer the replace queries with **Yes**.
10. If no error messages are shown in the grid area of the progress dialog: Press **Done** to exit.
11. To complete the diagrams of the imported control application:
 - a. If needed: Re-connect the diagram references to appropriate diagram variables and perform Generate Configuration Data for the affected Function Diagrams.
 - b. Download the application to make it work in the controller.



Always disable function **Automatic Write Allocation into CBM** in IO Allocation tool before performing import operation.

Export / Import Project

A control project can be exported with all its control applications, its Function Diagrams and controllers.

To export:

1. In the Control Structure navigate to the control project and drag and drop it onto the Contents tab of the Import / Export tool.
2. Depending on the target system add the needed dependencies.
3. Save the afw-file.

To import:



Replacing the control project overwrites changes of the same control project in the target system done in parallel to the changes in the engineering system.

1. Open the .afw file with the Import / Export tool.

2. In the toolbar press **Import All** and wait until an end message like “Import succeeded” is shown in the progress dialog.
3. If no error messages are shown in the grid area of the progress dialog: Press **Done** to exit.
4. All Function Diagrams keep their allocation to control applications and their I/O allocation. Do adaptations to the project contents as required.

Detailed Difference Report

Difference report provides the differences between Function Diagrams in the system and Diagrams present in the afw file to be imported. If there are any modifications between the Function Diagrams, the difference report provides a report for the Function aspect indicating the additions, deletions, or modifications between the Function Diagrams. For more details, refer to the *System 800xA, Maintenance (3BSE046784*)* manual.

Segmentation of AFW-Files

To limit memory consumption during export or import of an afw-file a segmentation of the exported contents into several afw-files can be necessary. Note that memory consumption is determined by the size of the aspects included into an afw-file. The size of an aspect depends on many circumstances, like for example added bitmaps to Function Diagrams or Graphic Displays or added documents to Document aspect, etc.

A control project to export can be segmented for example into:

- One file for the Control Project, in the Add items dialog un-check the check box **Include children**.
- One file for each Control Application.
- One file for each Controller.

A control application can be segmented for example into:

- One file for the Control application, in the Add items dialog un-check the check box **Include children**.

- One file for each Function Diagram.



CBM shows error messages “Error reading sourceblock: <application name>...” during import of the application and during import of the diagrams until import of the last but one. Further, Import shows a warning “Restore Single Control Module <SCM name> succeeded with warnings....” for each diagrams single control module. These messages can be accepted as mandatory generation of configuration data for the diagrams will correct the data of the application.

The files have to be imported in the same sequence as they were exported.

Engineering Repository

The 800xA System Engineering Repository set up is described in *System 800xA, Maintenance (3BSE046784*)*.

Standard entities of the control structure can be used to store and move Function Diagrams between different systems using the Engineering Repository. These entities can be directly exported to the Engineering Repository by selecting the Engineering Repository context menu available from the Control Project, Control Applications, and Controller.



If the target system does not contain the same functional structure, define a parent object as custom entity in the source system and then export the functional structure.

Configure the source and target systems to move engineering solutions including the functional structure elements from a source system to a target system using Engineering Repository:

In Source System:

1. Define a custom entity of the functional parent object with dependencies including children in the Repository Browser.
2. Export the custom entity of the source system to Engineering Repository.
3. In the Control Structure, export the desired standard entity of the Source System to the Engineering Repository.

In Target System:

1. Import the custom entity of the Functional Structure from Repository.
2. Import the Control Structure entity from Repository.

For more details on Engineering Repository, refer to *System 800xA 5.1 Configuration (3BDS011222*)*.

Section 3 Graphic Editor Reference



This section describes the graphic editor user interface of both Function Designer and Topology Designer.

The texts denominate designer / designer aspect / designer structure / designer component where

- Function Designer / Function aspect / Functional Structure / Function component / Function diagram
- and Topology Designer / Topology aspect / Control Structure / Topology component / Topology diagram apply.

Figures are taken from Function Designer. Exceptions for Topology Designer are mentioned explicitly.

Topology Designer / Topology Status Viewer functions are described further in *System 800xA Engineering, Engineering Studio Topology Designer (3BDS011225*)*.

Exploring the Windows User Interface

Designer can be used with full functionality in a preview window as part of the Plant Explorer window of the **Engineering Workplace** or Plant Explorer Workplace, or in a separate popup (overlap) window, see *System 800xA Engineering, Engineering Studio Function Designer Getting Started (3BDS100968*)*.

You can open several windows and order the windows cascading, tiled or as icons. It is also possible to split a window horizontally and/or vertically.

Main Window Handling

On top, the main window displays a navigation toolbar containing useful commands for navigating between aspects, views, and menus.



Figure 163. Navigation Toolbar of the Designer’s Main Window

In detail, the following commands and controls are offered:

Icon	Description
	Goes back to previous Aspect in display history
	Goes forward to next Aspect in display history
	Opens a context menu with object verbs of the selected Aspect Object. The icon displayed is the one of the selected Object.
	Clicking on the pointer lists all Aspects in display list.
	Drop Target; only active if the window is unpinned.
	Pins/Unpins the current window
	Opens a context menu for changing between aspect views, e.g. between Diagram and the Component view
	Opens a context menu with aspect verbs of the selected Aspect.

Diagram/Component View

Designer offers three different views on the designer aspect:

- Component view:
Use Component view to create symbols or templates for designer components.

- **Diagram view:**
Use Diagram view to create designer diagrams and to insert designer components.
- **Parent Diagram view:**
Use Parent Diagram view to navigate to the parent diagram. This view only exists if there is a diagram in an object in the designer structure above the object containing the designer aspect.

To open one of these views right click on the designer aspect and select the corresponding context menu item.




On choosing an object from designer structure or Object Type Structure, designer recognizes whether a (parent) diagram and/or component data is existing and automatically opens a corresponding default view.

The Component view is default if there exists component data, but no diagram nor a parent diagram.

The Parent Diagram view is default if there exists component data and a parent diagram, but no diagram data.

If both component data and diagram data are existing in a designer aspect, the Diagram view is opened.

Component View

To open designer's Component view, click the pointer on icon  in the navigation toolbar, and choose **Component** from the context menu.


Component view displays component document windows with optional split panes.

Diagram View

To open designer's Diagram view, click the pointer on icon  in the navigation toolbar, and choose **Diagram** from the context menu.

Diagram view displays a diagram document window with optional split panes.

Parent Diagram View

To open designer's Parent Diagram view, click the pointer on icon  in the navigation toolbar, and choose **Parent Diagram** from the context menu.

Parent Diagram view displays the parent diagram in designer structure, opens a diagram document window with optional split panes, and automatically navigates to the diagram page and scrolls to the component selected in Plant Explorer.

Diagram/Component Document Windows

Open Diagram/Component Documents

Diagram/Component document windows can be opened in several ways:

- To create a new diagram / component document on a function aspect, choose the **File > New** menu command.
- To open an existing Diagram / Component document, choose the **File > Open** menu command and use the **Open Aspect** dialog.
- To open a child (nested) diagram, containing a detailed diagram, choose the **View > Goto Child Diagram** menu command or the **Goto Child Diagram** context menu command (**Designer Component Context Menu**).

Open New Windows

The designer offers the possibility to open simultaneously several diagram/component documents. Each diagram/component document is opened in a separate window, so you can have several windows open at the same time.

The designer also offers the possibility to open the same diagram/component document in several windows. That can be useful, for example, to display different parts or pages of a diagram/component simultaneously in different windows.

To open a new window for an already diagram/component, choose the **Window > New Window** menu command. Each window displaying the same diagram/component document automatically gets a unique number as suffix in the window's title separated by a colon from the original title.

Arranging Windows

If you have opened simultaneously several diagram/component document windows you can arrange them by using **Window Menu** items **Cascade**, **Tile**, **Arrange Icons**, and **Workbook Mode**.

Splitting Windows

It is possible to split a window either in horizontal or in vertical direction or in both. These so-called splitter windows can be used to display different parts or pages of the same diagram which normally could not be simultaneously displayed on the screen.

To split a window, do as follows:

- To split the window horizontally, point to the corresponding area on top of the vertical scroll bar, and drag and drop the split bar.
- To split the window vertically, point to the corresponding area on the left side of the horizontal scroll bar, and drag and drop the split bar.

The following figure illustrates the procedure respectively the result of splitting a window horizontally and vertically.

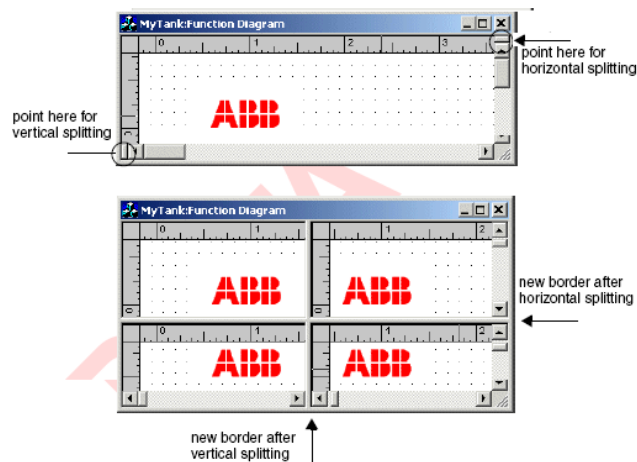


Figure 164. Splitting Windows

Dockable Windows/Toolbars

In designer, the menu bar, tool bars, Structure Browsers, Output Window, and Watch Window are dockable. That means, they can be

- docked on top of the main window
- docked on bottom of the main window
- docked at the left border of the main window
- docked at the right border of the main window
- or undocked, i.e. floating as popup window independent of the main window.

Figure 165 displays an example with undocked windows/toolbars.

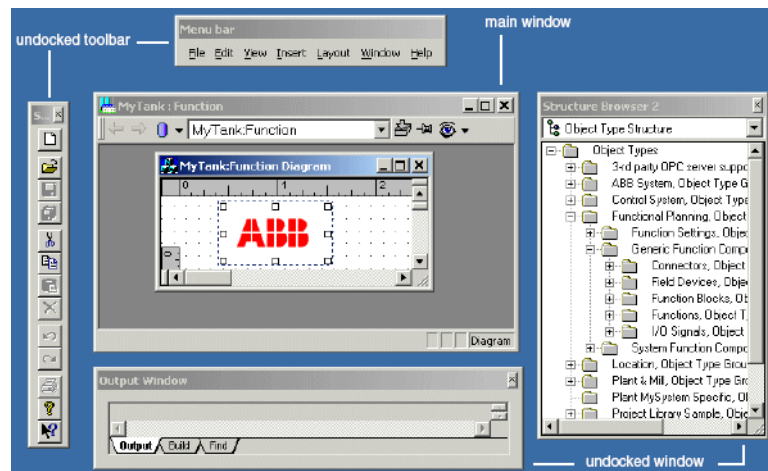


Figure 165. Undocked Windows/Toolbars

Figure 166 displays an example with docked windows/toolbars.

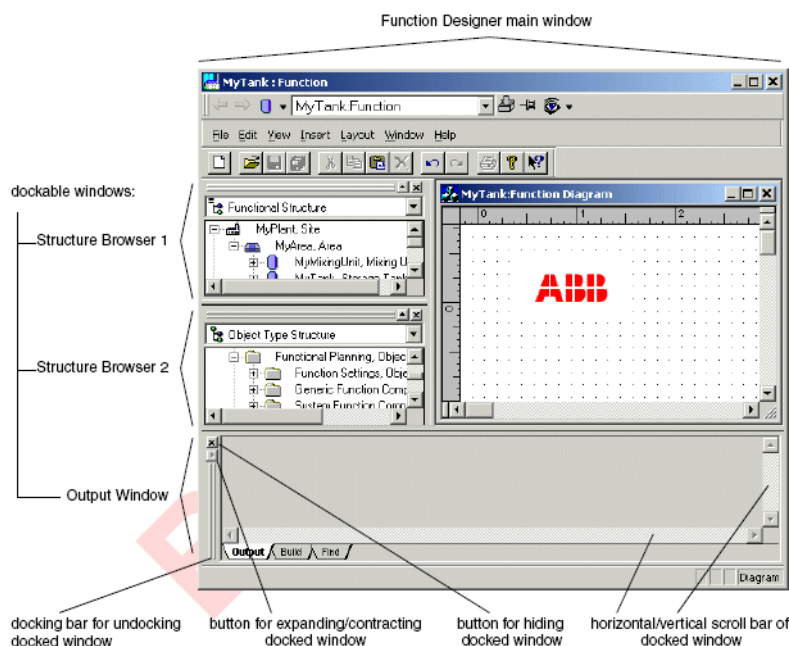


Figure 166. Docked Windows/Toolbars

Changing the Location of Docked Windows/Toolbars

You can dock a window/toolbar at the four border areas of the main window. To dock a window/toolbar at another location, drag-and-drop it to the wanted location.

Figure 167 displays an example with different dock locations.

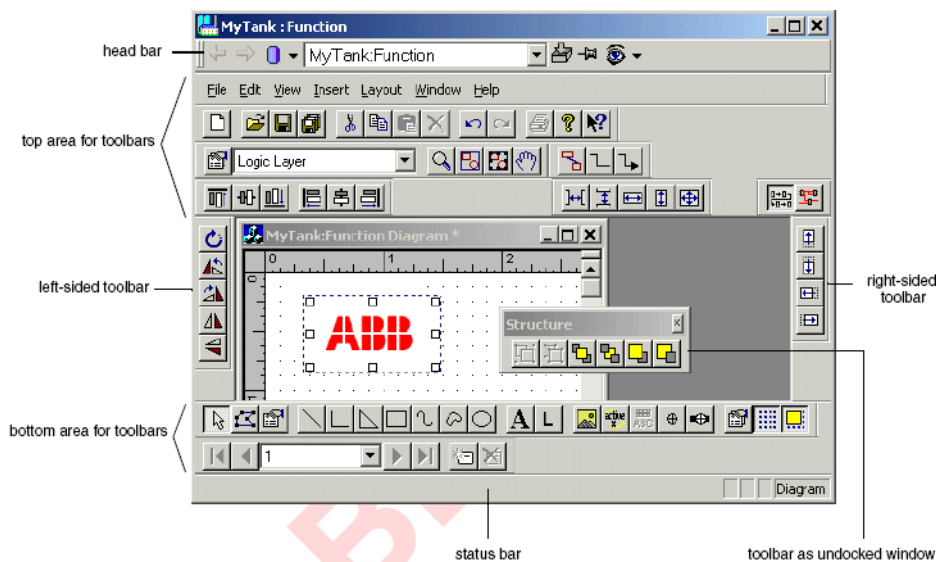


Figure 167. Toolbar Locations for Docking



The menu bar cannot be docked back by drag and drop action. Double-click the menu bar to dock it back.

Working with Menus

Designer's menu bar is switched according to

- the active view, Component View or (Parent) Diagram View
- the existence of component / diagram data.

The complete Menu bar contains the following menus:

- File Menu
- Edit Menu
- View Menu
- Insert Menu
- Layout Menu
- Allocation Menu (not available in Topology Designer)
- Online Menu (not available in Topology Designer)
- Window Menu
- Help Menu



The entry in the D/C column of tables listing menu commands indicates in which designer view (**D**ialog/**C**omponent) the menu command is available.

File Menu

Figure 168 and Figure 169 provides the different File menu options available in the Plant Explorer for the diagram and component views.

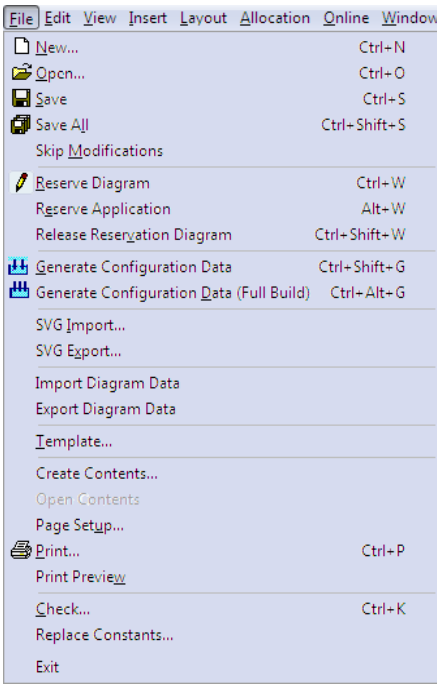


Figure 168. File Menu - Diagram View

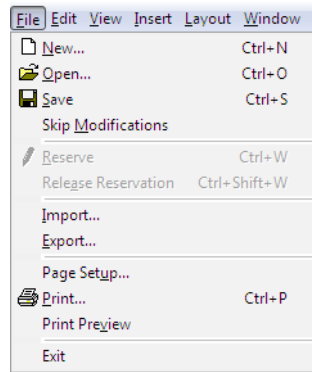


Figure 169. File Menu - Component View

Table 24. Description for File Menu Commands





Menu Command	Shortcut		Description	D/C	Toolbar
 File > New...	ALT > F > N	CTRL+N	Opens the dialog New dialog to create a new component, component template, or XML type definition. Topology Designer: No dialog.	C	Standard
 File > New...	ALT > F > N	CTRL+N	Creates a new diagram. Opens the Select Master Page Template dialog to copy or reference a template with header and footer like A3, A4 portrait/landscape. Topology Designer: Templates and Skeletons	D	Standard
 File > Open...	ALT > F > O	CTRL+O	Opens the Open Aspect dialog to open any component/diagram document, or any other aspect.	D/C	Standard
 File > Save	ALT > F > S	CTRL+S	Saves the current component/diagram document into the function aspect.	D/C	Standard

Table 24. Description for File Menu Commands (Continued)





Menu Command	Shortcut		Description	D/C	Toolbar
 File > Save All	ALT > F > L	CTRL+SHIFT+S	Saves all open component/diagram documents.	D	Standard
File > Skip Modifications	ALT > F > M		Undo all changes in current component/diagram document since last saving.	D/C	
  Topology Designer: File > Reserve Project or File > Reserve Application or File > Reserve Controller	ALT > F > R	CTRL+W	Reserves the diagram or control module type for exclusive modify access. Reserves the project or application or controller for exclusive modify access.	D/C	Standard
File > Reserve Application Not available in Topology Designer	ALT > F > E	ALT+W	Reserves the application (including all diagrams) or library for exclusive modify access.	D	
File > Release Reservation Diagram Function Designer (Component View) File > Release Reservation Diagram Topology Designer: File > Release Reservation	ALT > F > V ALT > F > A ALT > F > A	CTRL+SHIFT+W	Releases the diagram / the application or the control module type / the library from exclusive modify access. Releases the controller from exclusive modify access.	D/C	
 File > Generate Configuration Data Not available in Topology Designer	ALT > F > G	CTRL+SHIFT+G	Saves the diagram and (re)generates configuration data, e.g. for AC 800M. This command builds only the modified diagrams.	D	

Table 24. Description for File Menu Commands (Continued)



Menu Command	Shortcut		Description	D/C	Toolbar
 File > Generate Configuration Data (Full Build) (Full Build) Not available in Topology Designer	ALT > F > D	CTRL+ALT+G	Saves the diagram and (re)generates configuration data, e.g. for AC 800M. This commands builds all diagram.	D	
File > SVG Import... Topology Designer & Function Diagram (Component View): File > Import	ALT > F > I		Opens the File Import dialog to import component or background graphics.	D/C	
File > SVG Export... Topology Designer & Function Diagram (Component View): File > Export	ALT > F > X ALT > F > E		Opens the File Export dialog to export the current component/ diagram document into a different file format.	D/C	
File > Import Diagram Data			Opens the Open dialog to import the .xml file.	D	
File > Export Diagram Data			Opens the Save As dialog to save the required file in .xml format.	D	
File > Template...	ALT > F > T		Opens the Select Master Page Template dialog to copy or reference a drawing template with header and footer like A3, A4 portrait/landscape.	D	
File > Create Contents...			Creates or updates a contents diagram for a function diagram or a complete substructure.	D	
File > Open Contents			Opens already create contents diagrams.	D	
File > Page Setup...	ALT > F > U		Opens the Page Setup dialog to change page layout settings.	D/C	

Table 24. Description for File Menu Commands (Continued)

Menu Command	Shortcut		Description	D/C	Toolbar
 File > Print...	ALT > F > P	CTRL+P	Prints the active component/ diagram document, and optionally child diagrams.	D/C	Standard
File > Print Preview Topology Designer & Function Designer (Component View) File > Print Preview	ALT > F > W ALT > F > V		Displays a print preview of the active document.	D/C	
File > Check... Not available in Topology Designer	ALT > F > C	CTRL+K	Check diagram for valid identifiers and connections.	D	
File > Replace Constants... Not available in Topology Designer			Remove constant connections that equal the initial value.	D	
File > Exit			Allows the user to close the Function Diagram.	D/C	

- **File > New**

In Component view, choose this menu command to create a new function component, or to overwrite an existing one. For Function Designer, the command opens the New dialog. It allows to select between the three options:

- **Component:** To create or overwrite a complete designer component definition including fixed graphical symbol with ports, and a fixed parameter set.
- **Component Template:** To create or overwrite designer component template, i.e. a graphical symbol template. On instantiation in a function diagram, a complete graphical symbol and ports are automatically created on base of this template and on the actual parameter set.
- **XML Type Definition:** To create or overwrite a XML based type definition including a fixed and extensible set of parameters. On instantiation in a designer diagram, parameters are automatically created on base of this type definition. Typically used together with a designer component template to define object types.

For Topology Designer no dialog is shown.

In Diagram view, choose this menu command to create a new designer diagram, or to overwrite an existing one. The command opens the [Select Master Page Template](#) dialog for *Function Designer* and the [Templates and Skeletons](#) for Topology Designer:

- to copy or reference a template for the master page layer with header and footer like A3, A4 portrait/landscape
- or to leave the initial diagram blank
- or to copy a skeleton (Topology Designer only).
See also [File > Template](#).

- **File > Open**

Choose this menu command to open an existing designer component, designer diagram, or any other aspect. See also [Open Aspect](#).

- **File > Save**

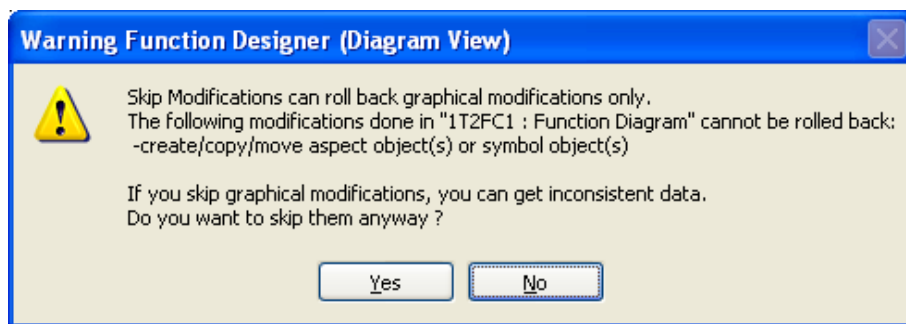
Choose this menu command to save your current designer component or designer diagram.

- **File > Save All**

Choose this menu command to save all open designer diagrams.

- **File > Skip Modifications**

Choose this menu command to undo all changes done in the current designer component or designer diagram since last save operation. User will get a warning as shown below:



Click **Yes** to confirm skipping the modification. Click **No** to cancel it, if you want to continue working your designer component or designer diagram.



If in Diagram view, you did create / delete aspect objects, modify aspect properties or modify connections these modifications are not undone on File > Skip Modifications, or on Close Diagram.

File > Skip Modifications restores graphics data stored in the designer aspect only, but does not restore aspect objects nor aspect property values.

- **File > Reserve Diagram**

In Functional Structure, it reserves the opened Function Diagram to gain exclusive modify access. This is a toggle menu item, and can be used to release a reserved Function Diagram.

In Object Type Structure, it reserves the library that is allocated to the control module object type of the opened Function Diagram to gain exclusive modify access.

- **File > Reserve Application**

It reserves the application to which the Function Diagram is allocated. This is a toggle menu item, and can be used to release a reserved application.

- **File > Release Reservation Diagram**

In Functional Structure, it releases the Function Diagram.

In Object Type Structure, it releases the Function Diagram.

- **File > Generate Configuration Data (Full Build)**

Note: This command is not available in Topology Designer.

Choose this menu command to your current function diagram and to (re)generate configuration data for an already allocated solution in Control Structure. In detail, code is regenerated for all allocatable groups belonging to this function diagram, which have a Control Structure aspect. For AC 800M objects this means that *Function Designer* generated diagrams / single control modules are generated again. The command generates the open diagram and all child (nested) diagrams, independent of their modification state.

- **File > Generate Configuration Data**

Note: This command is not available in Topology Designer.

For Function Diagrams created in 800xA 5.1:

This command does the same as command **File > Generate Configuration Data (Full Build)**.

For Function Diagrams restored to 800xA 5.1:

This command does the same as command **File > Generate Configuration Data (Full Build)**, but is generates only those diagrams which are in state 'modified' (red traffic light indication in the status bar).

- **File > SVG Import**

Opens the [File Import](#) dialog where you can select files to be imported.

See also [Import/Export](#).

- **File > SVG Export**

Opens the [File Export](#) dialog where you can define name, location and format to export graphics data of

- a designer component
- a single layer/page of a designer diagram
- a complete designer component.

See also [Import/Export](#).

- **File > Import Diagram Data**

Opens the **Open File** dialog to select the required .xml file.

See also [Export Diagram Data / Import Diagram Data](#).

- **File > Export Diagram Data**

Opens the **Save As** dialog to save the required file in .xml format.

See also [Export Diagram Data / Import Diagram Data](#).

- **File > Template**

For Function Designer it opens the [Select Master Page Template](#) dialog where you can select a predefined template with header/footers for the function diagram.

For Topology Designer it opens the [Templates and Skeletons](#) dialog where you can select a predefined template or skeleton with header/footers for the topology diagram.



You can add additional templates or customize installed ones in Object Type Structure:

For Function Designer under Object Types\Functional Planning\ Diagram Templates

For Topology Designer under Object Types\Topology Planning\ Diagram Templates.

See also the description in the Name aspect of corresponding object type group.



If you reference a template, the diagram's master page layer gets read-only. That means:

- Header/footer entries like author, document number, cannot be edited inside the diagram, but are referenced as aspect properties from a Function (or Topology) Diagram Document aspect of type Document.
- Diagram instances with referenced templates get automatically updated when the template changes.

If you copy a template, the diagram's master page layer gets writable. That means:

- Header/footer entries like author, document number, can be edited directly inside the diagram, but can also be referenced as aspect properties from a Function (or Topology) Diagram Document aspect of type Document.
- Diagram instances with copied templates do NOT get automatically updated when the template changes.
- **File > Create Contents**
Opens the [Contents](#) dialog to create a contents diagram
- **File > Open Contents**
Open already generated contents diagrams, see [Contents](#).

- **File > Page Setup**

Choose this menu command to open the standard [Page Setup](#) dialog where you can set page layout settings like printer paper format, page orientation, and margins.

- **File > Print**

Choose this menu command to open the standard [Print](#) dialog. In case there exist child diagrams on lower level, you can optionally print all sub-diagrams.

- **File > Print Preview**

Choose this menu command to open the [Print Preview](#) where you can check how your component or diagram would be printed. Via the following buttons, you can navigate through different pages of your designer diagram, zoom the previews, display two pages simultaneously and start printing:

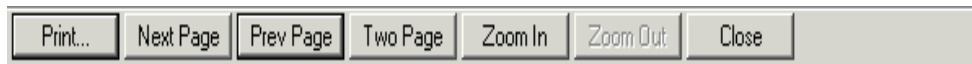


Figure 170. Print Preview buttons

- **File > Check**

Note: This command is not available in Topology Designer.

Choose this menu command to check the function diagram for valid identifiers connections, and inversions.



Save the modified Function Diagrams before performing a **File > Check** operation. Otherwise the latest graphical modifications (moving the existing Function Blocks / Control Modules, Field Devices, CBM_Signal objects and their signal connections) after last save will be lost.

In case there exist child diagrams on lower level, you can optionally check all sub-diagrams. The results of this command are displayed

- in the function diagram itself: Invalid identifiers, connections, or inversions are drawn in red.

- as warnings/errors in the output window in textual form. A double-click on such a warning/error navigates to the corresponding symbol, label, port, or connection link in the function diagram.
- **File > Complete Check**

In 800xA 5.1 and systems restored to 800xA 5.1, the **File > Complete Check** generates a Diagram check and the compilation errors or warnings appear in the Build pane of the Function Designer.
- **File > Replace Constants**

Note: This command is not available in Topology Designer.

Choose this menu command to remove constant connections if they equal the initial value. By that it is possible to tune the value after download without the need to regenerate configuration data.

This command can be optionally applied to child diagrams as well. You can invoke this command in order to bulk update diagrams created with 800xA 3.1 where initial values of the type were connected as constants.
- **File > Exit**

Choose this menu command in Diagram view or Component view to close the Function Diagram. It prompts the user to save the data before closing the Function Diagram.

Edit Menu

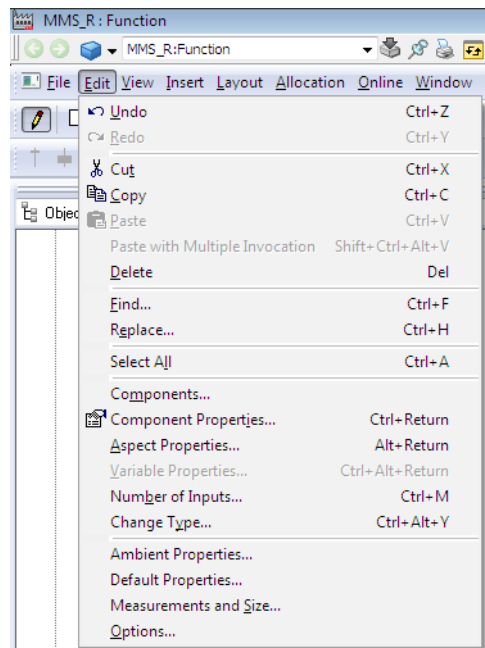


Figure 171. Edit Menu

Table 25. Description for Edit Menu Commands






Menu Command	Shortcut		Description	D/C	Toolbar
 Edit > Undo	ALT > E > U	CTRL+Z	Undoes the last action.	D/C	Standard
 Edit > Redo	ALT > E > R	CTRL+Y	Redoes the previously undone action.	D/C	Standard
 Edit > Cut	ALT > E > T	CTRL+X	Cuts the selection to the clipboard.	D/C	Standard
 Edit > Copy	ALT > E > C	CTRL+C	Copies the selection to the clipboard.	D/C	Standard
 Edit > Paste	ALT > E > P	CTRL+V	Inserts the clipboards contents.	D/C	Standard

Table 25. Description for Edit Menu Commands (Continued)


Menu Command	Shortcut		Description	D/C	Toolbar
Edit > Paste with Multiple Invocation Not available for Topology Designer		SHIFT+ CTRL+ ALT+V	Inserts Diagram References and Variables with the same name as the source.	D/C	
Edit > Delete	ALT > E > D	DEL	Deletes the selected item(s).	D/C	
Edit > Find Not available for Topology Designer.	ALT > E > F	CTRL+F	Finds a string according to selected Options.	D	
Edit > Replace Not available for Topology Designer.	ALT > E > E	CTRL+H	Finds and replaces a string according to selected Options.	D	
Edit > Select All	ALT > E > L	CTRL+A	Selects all components of the active layer and active page.	D	
Edit > Components...	ALT > E > M		Opens the Components dialog to navigate to individual graphical components.	D/C	
 Edit > Component Properties...	ALT > E > I	CTRL+ Return	Opens the Component Properties dialog to set graphical properties.	D/C	Drawing
Edit > Aspect Properties...	ALT > E > A	ALT+ RETURN	Opens the Aspect Properties dialog (Ports property page) dialog.	D	
Edit > Variable Properties... Not available for Topology Designer	ALT > E > V	CTRL+ ALT+ RETURN	Opens the Variable Properties dialog .	D	
Edit > Number of Inputs... Not available for Topology Designer	ALT > E > B	CTRL+ M	Opens the Number of Inputs dialog to configure number of inputs for system functions like and / or	D	
Edit > Change Type... Not available for Topology Designer	ALT > E > Y	CTRL+ ALT+ Y	Opens the Change Type dialog to change the type of a system function, function block or control module.	D	
Edit > Ambient Properties			Opens the Ambient Properties dialog to set ambient properties valid for the whole component/diagram.	D/C	
Edit > Default Properties			Opens the Default Properties dialog to set default graphical properties for new components.	D/C	
Edit > Measurements and Size	ALT > E > S		Opens the Measurements and Size dialog to set properties of the drawing page,	D/C	

Table 25. Description for Edit Menu Commands (Continued)

Menu Command	Shortcut		Description	D/C	Toolbar
Edit > Options	ALT > E > O		Opens the Options dialog	D	

- **Edit > Undo**
Choose this menu command to undo the last action. See also [Undo](#).
- **Edit > Redo**
Choose this menu command to redo the last undone action. See also [Redo](#).
- **Edit > Cut**
Choose this menu command to cut selected components to the clipboard.
- **Edit > Copy**
Choose this menu command to copy selected components to the clipboard
- **Edit > Paste**
Choose this menu commands to paste components (cut or copied via the **Cut** or **Copy** command) from the clipboard into the component/diagram document.
- **Edit > Paste with Multiple Invocation**
Note: Not available for Topology Designer.
Choose this menu command to paste diagram references or variable components with the same name as the source. For Function Diagrams restored to 800xA 5.1, choose this menu command to paste function blocks or variable components with the same name as the source.
- **Edit > Delete**
Choose this menu command to delete selected component(s) from the component/diagram document.
- **Edit > Select All**
Choose this menu command to select all components of the active layer and active page.
- **Edit > Find**

Note: Not available for Topology Designer.

Choose this menu command to open the **Find** tab of the Find dialog and then to find a string according to selected **Options**.

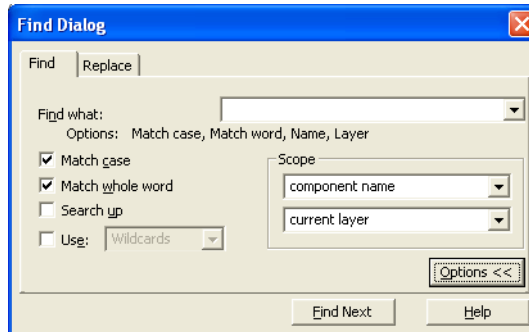


Figure 172. Find Dialog, Find Tab (Options Shown)

Press **Options>>** to show, press **Options<<** to hide the options part of the dialog:

- **Match case:** Mark this check box if the case of the Find what: string shall be matched.
- **Match whole word:** Mark this check box if the whole Find what: string shall be matched.
- **Search up:** Mark this check box if the search direction shall be upwards.
- **Use:** Mark this check box if the wildcards * (for a sequence of arbitrary characters) or ? (for one arbitrary character) shall be taken care during search. Disables the check boxes for case and whole word.
- **Scope:**
 In the first combo box you can choose between: Component name, component type name, control builder name, connection string, unconnected mandatory ports, unconnected ports, unconnected inputs, unconnected outputs, port name, any string.
 Default is component name.
 In the second combo box you can choose between: Current layer, current diagram (all layers), current page, current selection.

- **Edit > Replace**

Note: Not available for Topology Designer.

Choose this menu command to open the **Replace** tab of the Find dialog and then to find and replace a string according to selected **Options**.

Options see **Edit > Find** above.

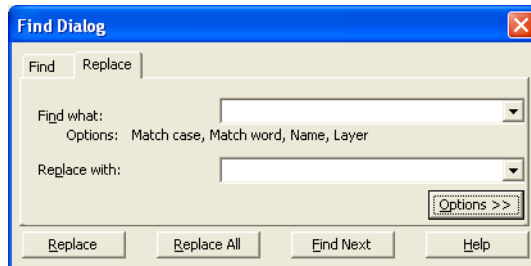


Figure 173. Find Dialog, Replace Tab (Options Not Shown)

- **Edit > Components**

Choose this menu command to open the [Components](#) dialog

- to navigate to individual graphical components in the component/diagram document
- to open the [Component Properties](#) dialog.

- **Edit > Component Properties**

Choose this menu command to open the [Component Properties](#) dialog to set graphical properties of the selected graphic component(s) like line width, fill color, font, etc.

- **Edit > Aspect Properties**

Choose this menu command to open the Aspect Properties dialog to view or set **Aspect Properties** of the selected designer diagram, designer component, or port.

- **Edit > Variable Properties**

Note: Not available for Topology Designer.

Choose this menu command to open the Variable Properties dialog to change variable properties.

- **Edit > Number of Inputs**

Note: Not available for Topology Designer.

Choose this menu command to change the number of input ports of applicable components.

- **Edit > Change Type**

Note: Not available for Topology Designer.

Choose this menu command to change the type of applicable components.

- **Edit > Ambient Properties**

Choose this menu command to open the [Ambient Properties](#) dialog to set ambient properties that are valid for the while component/diagram document like background and selection colors.

- **Edit > Default Properties**

Choose this menu command to open the [Default Properties](#) dialog to set default graphical properties valid for newly created/inserted graphic components (primitives).

- **Edit > Measurements and Size**

Choose this menu command to open the [Measurements and Size](#) dialog to adjust units of measure, drawing scale, module, and the drawing page size for the active function component / function diagram.

- **Edit > Options**

Choose this menu command to open the [Options](#) dialog to set some options concerning printing, pasting components, and auto routing.

View Menu

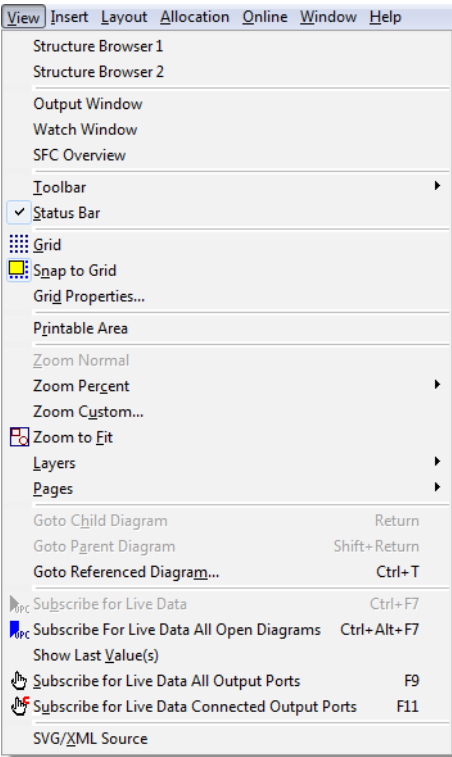


Figure 174. View Menu

Table 26. Description for View Menu Commands

Menu Command	Shortcut		Short Description	D/C	Toolbar
View > Structure Browser 1			Choose this menu option to access the Structure Browser 1.	D	
View > Structure Browser 2			Choose this menu option to access the Structure Browser 2.	D	

Table 26. Description for View Menu Commands (Continued)







Menu Command	Shortcut		Short Description	D/C	Toolbar
View > Output Window			Choose this menu option to access the Output window.	D	
View > Watch Window			Choose this menu option to access the Watch window.	D	
View > SFC Overview			Choose this menu option to open a new dockable window that contains the sequence tree structure.	D	
View > Toolbar > Customize	ALT > V > T > C		Opens the Customize dialog.	D/C	
View > Status Bar	ALT > V > S		Turns on or off the display of the status bar	D/C	
 View > Grid	ALT > V > G		Shows or hides the grid.	D/C	Drawing
 View > Snap to Grid	ALT > V > N		Snaps symbols to grid.	D/C	Drawing
View > Grid Properties	ALT > V > D		Opens the Grid Properties dialog.	D/C	
View > Printable Area	ALT > V > R		Check to outline printable area in print preview according to the selected printer and print paper size.	D/C	
View > Zoom Normal	ALT > V > Z		Zooms the canvas display to the normal default size (100%)	D/C	
View > Zoom Percent	ALT > V > C > 5 ALT > V > C > 7 ALT > V > C > 1 ALT > V > C > 2		Zooms the canvas display to 50% of the normal size. Zooms the canvas display to 75% of the normal size. Zooms the canvas display to the normal default size (100%). Zooms the canvas display to 200% of the normal size.	D/C	
View > Zoom Custom	ALT > V > U		Opens the Zoom dialog.	D/C	

Table 26. Description for View Menu Commands (Continued)

Menu Command	Shortcut		Short Description	D/C	Toolbar
 View > Zoom to Fit	ALT > V > F		Zooms so that all components are visible.	D/C	View
 View > Layers > Properties	ALT > V > L > P		Sets Layer Properties .	D	View, Drawing
View > Layers > L1 Master Page Layer	ALT > V > L > 1		Sets active layer to Master Page Layer.	D	
View > Layers > L2 Background Layer	ALT > V > L > 2		Sets active layer to Background Layer.	D	
View > Layers > L3 Logic Layer	ALT > V > L > 3		Sets active layer to Logic Layer.	D	
View > Layers > L4 Annotation Layer	ALT > V > L > 4		Sets active layer to Annotation Layer.	D	
 View > Pages > Insert New Page	ALT > V > P > I		Inserts a new page.	D	View
 View > Pages > Delete Page	ALT > V > P > D		Deletes the active page. A page must be empty before it can be deleted.	D	View
 View > Pages > First Page	ALT > V > P > F		Navigates to the first page.	D	View
 View > Pages > Previous Page	ALT > V > P > P		Navigates to the previous page.	D	View
 View > Pages > Next Page	ALT > V > P > N		Navigates to the next page.	D	View
 View > Pages > Last Page	ALT > V > P > L		Navigates to the last page.	D	View
View > Pages > Goto Page	ALT > V > P > G		Opens the Goto page dialog.	D	
 View > Pages > Page Comment	ALT > V > P > C		Opens the Page Comment dialog.	D	Page
View > Goto Child Diagram	ALT > V > H	Return	Opens the child diagram represented by the selected symbol.	D	

Table 26. Description for View Menu Commands (Continued)

Menu Command	Shortcut		Short Description	D/C	Toolbar
View > Goto Parent Diagram	ALT > V > A	Shift + Return	Opens the parent diagram represented by the selected symbol.	D	
View > Goto Referenced Diagram Not available for Topology Designer.	ALT > V > M	Ctrl + T	Opens the referenced diagram represented by the selected off-diagram symbol.	D	
 View > Subscribe for Live Data	ALT > V > B	Ctrl + F7	Start/Stop display of live data (on-line values) in the function diagram and/or in the Watch window.	D	
 View > Subscribe for Live Data All Open Diagrams		Ctrl + Alt + F7	Start/Stop display of live data (on-line values) for all open Function Diagrams.	D	
View > Show Last Value(s)	ALT > V > V		Show/Hide recent on-line values when subscription for life data has stopped.	D	
 View > Subscribe for Live Data All Output Ports	ALT > V > S	F9	Show/Hide recent on-line values when subscription for life data has stopped.	D	
 View > Subscribe for Live Data Connected Output Ports	ALT > V > U	F11	Show/Hide recent on-line values when subscription for live data has stopped.	D	
View > SVG/XML Source	ALT > V > X		Opens the SVG/XML Source window.	D/C	

- View > Toolbar > Customize**

Choose this menu command to open the [Customize](#) dialog to customize toolbars.

- **View > Status Bar**


Choose this menu command to turn on or off the display of the **Status Bar** on bottom of the designer window.



Figure 175. Status bar

The **Status Bar** displays, for example,

- short menu and toolbar button help texts.
- OPC quality and time stamp while **Online > Subscribe for Live Data** is active.
Note: Not available for Topology Designer.
- indicator ‘**NUM**’ and ‘**SCRL**’ for Numlock or Scroll-lock, respectively.
- indicator ‘**Read**’ for read-only components, diagrams, and layers.
Components and diagrams might be read-only due to access control and/or inheritance of relevant aspects and/or if ActiveX Run Mode is enabled.
- information about the currently displayed mode (view) of the designer, **Diagram** or **Component**.
- information about the diagram state, e.g. modified, generated, and loaded.
Note: Not available for Topology Designer.

If the display is turned on, the menu item is checked by  .

- **View > Grid**

Choose this menu command to turn on or off grid display in the current component/diagram document.

- **View > Snap to Grid**

Choose this menu command to snap components to the current grid on create, move, or copy.

- **View > Grid Properties**

Choose this menu command to open the [Grid Properties](#) dialog to adjust grid size and color.

- **View > Printable Area**

Check this menu command to outline the printable area in print preview according to the selected printer and print paper size.

- **View > Zoom Normal**

Choose this menu command to zoom the canvas display to default size (100%). See also [Zoom Normal](#).

- **View > Zoom Percent**

Choose this menu command and select any of the available submenus to zoom the canvas display to the required percentage of normal size.

- **View > Zoom Custom**

Choose this menu command to open the **Zoom** dialog to zoom the canvas display to the **Magnification** value selected. See also [Zoom Custom](#).

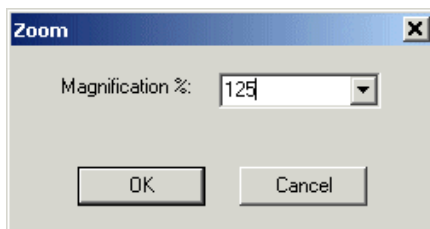


Figure 176. Zoom dialog

- **View > Zoom to Fit**

Choose this menu command to zoom the canvas to a size in order to display all components of the active page without the need for scrolling.

- **View > Layers > Properties**

Choose this menu command to open the [Layer Properties](#) dialog to set properties of the different layers.

- **View > Layers > L1 Master Page Layer**
Choose this menu command to make the **Master Page Layer** active.
- **View > Layers > L2 Background Layer**
Choose this menu command to make the **Background Layer** active.
- **View > Layers > L3 Logic Layer**
Choose this menu command to make the **Logic Layer** active.
- **View > Layers > L4 Annotation Layer**
Choose this menu command to make the **Annotation Layer** active.
- **View > Pages > Insert New Page**
Choose this menu command to insert a new page to the diagram.
- **View > Pages > Delete Page**
Choose this menu command to delete the current page of the diagram. A page to be deleted must be empty, i.e. there are no components on any layer besides the master page layer.
- **View > Pages > First Page**
Choose this menu command to navigate to the first page of the diagram.
- **View > Pages > Previous**
Choose this menu command to navigate to the previous page of the diagram.
- **View > Pages > Next**
Choose this menu command to navigate to the next page of the diagram.
- **View > Pages > Last**
Choose this menu command to navigate to the last page of the diagram.
- **View > Pages > Goto Page**
Choose this menu command to open the **Goto Page** dialog to enter the page number of interest in order to navigate to.
(see also [Navigation to a certain page](#)).

- **View > Pages > Page Comment**

Choose this menu command to open the [Page Comments](#) dialog to enter a comment for a single page or a set of pages.

- **View > Goto Child Diagram**

Choose this menu command to open the child (nested) diagram that is represented in the active diagram by the symbol selected.

This menu command is only accessible if a symbol representing a (nested) diagram is selected.

- **View > Goto Parent Diagram**

Choose this menu command to open the parent diagram that is represented in the active diagram by the symbol selected.

This menu command is only accessible if a parent diagram exists for the current diagram.

- **View > Goto Referenced Diagram**

Note: Not available for Topology Designer.

Choose this menu command to open the referenced diagram that is represented in the active diagram by the selected off-diagram reference symbol.

This menu command is only accessible if an off-diagram reference symbol is selected.

- **View > Subscribe for Live Data**

Choose this toggle menu command to start/stop display of live data (on-line values) in the designer diagram or in the Watch window (*Function Designer* only).

- **View > Subscribe for Live Data All Open Diagrams**

Choose this toggle menu command to start/stop display of live data (on-line values) for all opened Function Diagrams.

- **View > Show Last Value(s)**

Choose this toggle menu command to show/hide recent on-line values when subscription for live data has stopped.

- **View > Subscribe for Live Data All Output Ports**

Choose this menu command to display on-line values for all links at the output pins on the current page. The shortcut key for this menu option is F9.

- If selected, Subscribe for Live Data All Output Ports is active. i.e. on-line values are retrieved from the OPC server and displayed on the output pins of the current page.
- If unchecked, Subscribe for Live Data All Output Ports is inactive. i.e. no on-line values are retrieved from the OPC server. According to the setting of '**Show Last Value(s)**', the OPC items in the diagram display a '*' or the recent value retrieved, and the OPC items display an empty value or the recent value retrieved.

- **View > Subscribe for Live Data Connected Output Ports**

Choose this menu command to display on-line values for all links at the connected output pins on the current page. The shortcut key for this menu option is F11

- **View > SVG/XML Source**

Choose this menu command to open the **SVG/XML Source** window displaying the designer component / designer diagram represented as SVG/XML source code.

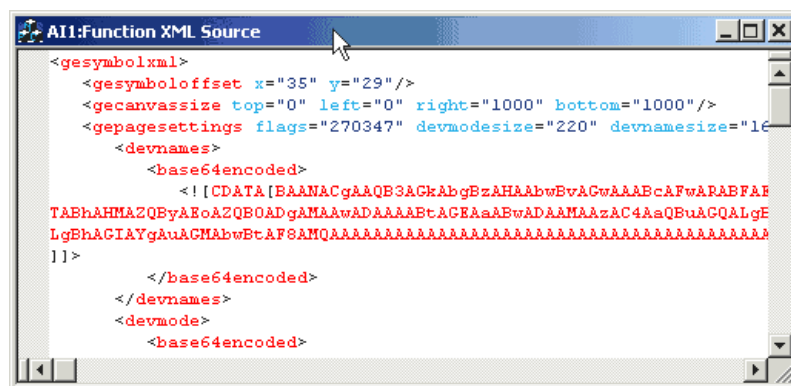


Figure 177. SVG/XML Source Window

Insert Menu

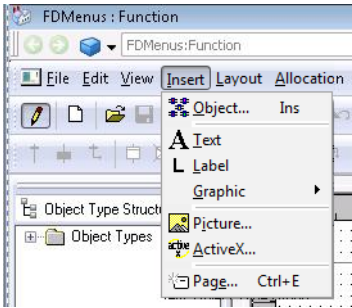


Figure 178. Insert Menu

Table 27. Description for Insert Menu Commands
















Menu Command	Shortcut	Description	D/C	Toolbar
 Insert > Object	ALT > I > Ins	Inserts an object (designer component) into the diagram.	D	
 Insert > Circle Port	ALT > I > C	Inserts a circle port.	C	Port
 Insert > Line Port	ALT > I > P	Inserts a line port.	C	Port
 Insert > Text	ALT > I > T	Inserts text.	D/C	Drawing
 Insert > Label	ALT > I > L	Inserts a label.	D/C	Drawing
 Insert > Graphic > Lines	ALT > I > G > L	Draws a single line.	D/C	Drawing
 Insert > Graphic > Polyline	ALT > I > G > P	Draws a multi-segmented line.	D/C	Drawing
 Insert > Graphic > Rectangle	ALT > I > G > R	Draws a rectangle.	D/C	Drawing
 Insert > Graphic > Polygon	ALT > I > G > O	Draws a polygon.	D/C	Drawing

Table 27. Description for Insert Menu Commands (Continued)

Menu Command	Shortcut	Description	D/C	Toolbar
 Insert > Graphic > Poly Curve	ALT > I > G > C	Draws a series of connected curves.	D/C	Drawing
 Insert > Graphic > Closed Curve	ALT > I > G > U	Draws a polycurve with connecting start and end points.	D/C	Drawing
 Insert > Graphic > Ellipse	ALT > I > G > P	Draws an ellipse.	D/C	Drawing
 Insert > Picture	ALT > I > I	Inserts a picture.	D/C	Drawing
 Insert > ActiveX	ALT > I > A	Inserts an ActiveX component.	D/C	Drawing
 Insert > Page	ALT > I > E	Creates a Page.	D	Page

- **Insert > Object**

In the Diagram view, click **Insert > Object** to open the **Insert Objects** dialog box. In the **Object Types** tab, select the required Structure and navigate and select the required object. Click **Apply**. The selected object is inserted in the Function Diagram.

See [Inserting/Creating Components](#).

- **Insert > Circle Port**

In Component view, choose this command to insert a circle port. See also [Inserting Ports](#).

- **Insert > Line Port**

In Component view, choose this command to insert a line port. See also [Inserting Ports](#).

- **Insert > Text**

Choose this menu command to insert a text component. See [Inserting Text Components](#).

- **Insert > Label**
Choose this menu command to insert a label. See [Inserting Label Components](#).
- **Insert > Graphic > Lines**
Choose this menu command to insert a single line. See also [Inserting a Line](#).
- **Insert > Graphic > Polyline**
Choose this menu command to insert a polyline.
- **Insert > Graphic > Rectangle**
Choose this menu command to insert a rectangle.
- **Insert > Graphic > Polygon**
Choose this menu command to insert a polygon.
- **Insert > Graphic > Poly Curve**
Choose this menu command to insert a polycurve.
- **Insert > Graphic > Closed Curve**
Choose this menu command to insert a closed curve.
- **Insert > Graphic > Ellipse**
Choose this menu command to insert an ellipse.
- **Insert > Picture**
Choose this menu command to open a standard File Open dialog from which you can select an image in either .bmp, .wmf or .dib format. See also [Inserting Pictures/Images](#).
- **Insert > ActiveX**
Choose this menu command to insert an ActiveX control, or a Graphic Element built with Graphics Builder. See also [Inserting ActiveX Controls](#).
- **Insert > Page**
In Diagram view, choose this menu command to create a new page. The new page normally will be inserted behind the active page.

If you choose this menu command when the 1st page is active, the system will ask you where you want to insert the new page:

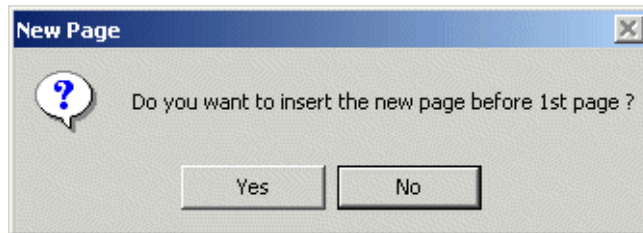


Figure 179. New Page

Click **Yes** or **No** according to the requirement.

See also [Pages](#).

Layout Menu

Table 28. Description for Layout Menu Commands























Menu Command	Shortcut	Description	D/C	Toolbar
 Layout > Align > Left	ALT > L > A > L	Aligns left.	D/C	Align
 Layout > Align > Center	ALT > L > A > C	Aligns centered.	D/C	Align
 Layout > Align > Right	ALT > L > A > R	Aligns right.	D/C	Align
 Layout > Align > Top	ALT > L > A > T	Aligns top.	D/C	Align
 Layout > Align > Middle	ALT > L > A > M	Aligns middle.	D/C	Align
 Layout > Align > Bottom	ALT > L > A > B	Aligns bottom.	D/C	Align
 Layout > Space Evenly > Across	ALT > L > S > A	Spaces components horizontally.	D/C	Layout
 Layout > Space Evenly > Down	ALT > L > S > D	Spaces components vertically.	D/C	Layout

Table 28. Description for Layout Menu Commands (Continued)

Menu Command	Shortcut	Description	D/C	Toolbar
 Layout > Make Same Size > Width	ALT > L > M > W	Makes components in the same width.	D/C	Layout
 Layout > Make Same Size > Height	ALT > L > M > H	Makes components in the same height.	D/C	Layout
 Layout > Make Same Size > Both	ALT > L > M > B	Makes components in the same size.	D/C	Layout
 Layout > Rotate > Free	ALT > L > R > F	Rotates components.	D/C	Rotate
 Layout > Rotate > Left	ALT > L > R > L	Rotates 90 degrees counter clockwise.	D/C	Rotate
 Layout > Rotate > Right	ALT > L > R > R	Rotates 90 degrees clockwise.	D/C	Rotate
 Layout > Flip > Horizontal	ALT > L > F > H	Flips around the horizontal center.	D/C	Rotate
 Layout > Flip > Vertical	ALT > L > F > V	Flips around the vertical center.	D/C	Rotate
 Layout > Order > Bring to Front	ALT > L > O > T	Brings to front.	D/C	Structure
 Layout > Order > Send to Back	ALT > L > O > K	Sends to back.	D/C	Structure
 Layout > Order > Bring Forward	ALT > L > O > F	Brings forward.	D/C	Structure
 Layout > Order > Send Backward	ALT > L > O > B	Sends backward.	D/C	Structure
 Layout > Group	ALT > L > G	Groups components.	D/C	Structure
 Layout > Ungroup	ALT > L > U	Ungroups components.	D/C	Structure

- **Layout > Align > Left**

Choose this menu command to align vertically selected components to the left edge of the anchor component.

- **Layout > Align > Center**
Choose this menu command to vertically align selected components to the center of the anchor component.
- **Layout > Align > Right**
Choose this menu command to vertically align selected components to the right edge of the anchor component.
- **Layout > Align > Top**
Choose this menu command to horizontally align selected components to the top of the anchor component.
- **Layout > Align > Middle**
Choose this menu command to horizontally align selected components to the center of the anchor component.
- **Layout > Align > Bottom**
Choose this menu command to horizontally align selected components to the bottom of the anchor component.
- **Layout > Space Evenly > Across**
Choose this menu command to space selected components evenly between the left-most and right-most components selected.
- **Layout > Space Evenly > Down**
Choose this menu command to space selected components evenly between the top-most and bottom-most components selected.
- **Layout > Make Same Size > Width**
Choose this menu command to change the width of selected components to match the width of the anchor component.
- **Layout > Make Same Size > Height**
Choose this menu command to change the height of selected components to match the height of the anchor component.

- **Layout > Make Same Size > Both**

Choose this menu command to change the width and height of selected components to match the size of the anchor component.

- **Layout > Rotate > Free**

Choose this menu command to set the canvas to rotate mode. Changes the cursor to a circular arrow when it is over a component, and allows you to grab a component and rotate it by mouse. See also [Rotate Free](#).

- **Layout > Rotate > Left**

Choose this menu command to rotate selected components by 90 degrees to the left. See also [Rotate Left/Right](#).

- **Layout > Rotate > Right**

Choose this menu command to rotate selected components by 90 degrees to the right.

- **Layout > Flip > Horizontal**

Choose this menu command to flip selected components 180 degrees around the X axis.

- **Layout > Flip > Vertical**

Choose this menu command to flip selected components 180 degrees around the Y axis.

- **Layout > Order > Bring to Front**

Choose this menu command to bring selected components to front so that they are not hidden by other components. See also [Bring to Front](#).

- **Layout > Order > Send to Back**

Choose this menu command to send the selected components back so that they can get hidden by other components in the foreground. See also [Send to Back](#).

- **Layout > Order > Bring Forward**

Choose this menu command to bring selected component(s) one level forward. See also [Bring Forward](#).

- **Layout > Order > Send Backward**

Choose this menu command to send selected component(s) one level backward. See also [Send Backward](#).

- **Layout > Group**

Choose this menu command to group selected components. All grouped components can now be manipulated as one component.

- **Layout > Ungroup**

Choose this menu command to ungroup a group of components.

Allocation Menu



The Allocation menu is not available in Topology Designer.

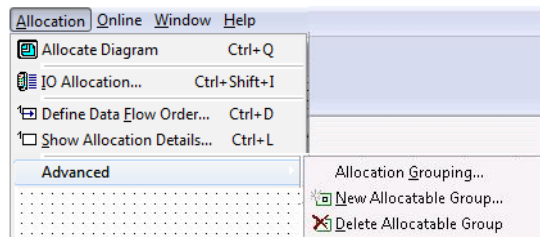








Figure 180. Allocation Menu

Table 29. Description for Allocation Menu Commands

Menu Command	Shortcut		Description	D/C	Toolbar
 Allocation > Allocate Diagram		Ctrl + Q	This menu option calls the Allocatable Group aspect and allows the user to allocate the Function Diagram to the required application in Control builder M.	D	
 Allocation > IO Allocation	ALT > A > I	CTRL+SHIFT +I	Opens the IO Allocation tool in the context of the diagram.	D	
 Allocation > Define Data Flow Order	ALT > A > F	CTRL +D	Opens a dialog to define the data flow order.	D	Allocation
 Allocation > Show Allocation Details	ALT > A > S	CTRL +L	Shows allocation group details like name, short name, data flow order, and color	D	Allocation
Allocation > Advanced > Allocation Grouping...*			This menu option groups the existing allocatable groups.	D	
 Allocation > Advanced > New Allocatable Group...*			This menu option is used to create a new allocatable group.	D	
 Allocation > Advanced > Delete Allocatable Group...*			This menu option is used to delete an existing allocatable group.	D	

* This option is supported only in Function Diagrams restored to 800xA 5.1.

- **Allocation > Allocate Diagram**

Choose this menu command to call the Allocatable Group aspect and to allocate the Function Diagram to the required application in Control builder M.

- **Allocation > IO Allocation**

In Diagram view, choose this menu command to allocate signals given by the diagram.

- **Allocation > Define Data Flow Order**

In Diagram view, choose this menu command to Define the Data Flow Order of the Function Diagram.

- **Allocation > Show Allocation Details**

In Diagram view, choose this menu command to display allocation details at function components, e.g. data flow order, name, short name, and color of the allocatable group.

- **Allocation > Advanced > Allocation Grouping...**

In Diagram view, choose this menu command to group the existing allocatable groups available in the Function Diagrams created prior to 800xA 5.1 and for diagrams restored to 800xA 5.1.

- **Allocation > Advanced > New Allocatable Group...**

In Diagram view, choose this menu command to create a new allocatable group for Function Diagrams created prior to 800xA 5.1 and for diagrams restored to 800xA 5.1.

- **Allocation > Advanced > Delete Allocatable Group...**

In Diagram view, choose this menu command to delete an existing allocatable group for Function Diagrams created prior to 800xA 5.1 and for diagrams restored to 800xA 5.1.



The following menu options are unavailable and appear dimmed for the newly created Function Diagrams using 800xA 5.1:

- Allocation grouping...
- New Allocatable Group...
- Delete Allocatable Group...

These menu options are available for Function Diagrams created prior to 800xA 5.1 and for diagrams restored to 800xA 5.1.

Online Menu



The Online Menu is not available in Topology Designer.

The online menu options are provided in Fig below:

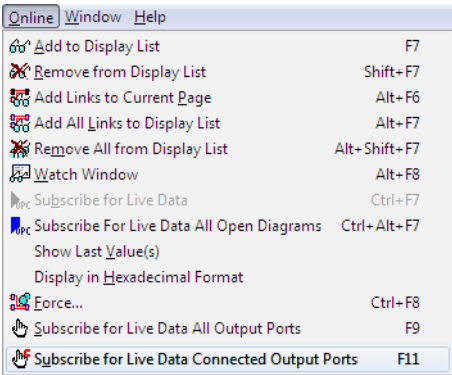


Figure 181. Online Menu

Table 30. Description for Online Menu Commands












Menu Command	Shortcut		Description	D/C	Toolbar
 Online > Add to Display List	ALT > O > A	F7	Add selected item (port, symbol, link) to display list.	D	Online
 Online > Remove from Display List	ALT > O > R	Shift +F7	Remove selected item(s) (port, symbol, link) from display list.	D	Online
 Online > Add Links to Current Page	ALT > O > P	ALT +F6	Add all links on current page to display list.	D	Online
 Online > Add all Links to Display List	ALT > O > L	ALT +F7	Add all links on current page to display list.	D	Online
 Online > Remove all from Display List	ALT > O > M	ALT+SHIFT +F7	Remove all links on current page from display list.	D	Online
 Online > Watch Window	ALT > O > W	ALT +F8	Display the Watch window to check and set variable values.	D	Online
 Online > Subscribe for Live Data	ALT > O > B	CTRL +F7	Start/Stop display of on-line values (live data)	D	Online

Table 30. Description for Online Menu Commands (Continued)

Menu Command	Shortcut		Description	D/C	Toolbar
 Online > Subscribe For Live Data All Open Diagrams		CTRL+ALT + +F7	In Diagram view, choose this toggle menu command to start/stop display of on-line values for all open diagrams.	D	
Online > Show Last Value(s)	ALT > O> V		Show/Hide recent on-line values when subscription for live data has stopped.	D	
Online > Display in Hexadecimal Format	ALT > O> H		Show/Hide on-line values in hexadecimal format.	D	
 Online > Force...	ALT > O> F	CTRL+F8	Opens the Force dialog to force applicable signals.	D	Online
 Online > Subscribe for Live Data All Output Ports	ALT > O> S	F9	The Function Diagram subscribes for Online Data at the output pins.		
 Online > Subscribe for Live Data Connected Output Ports	ALT > O> U	F11	The Function Diagram subscribes for Online Data at the connected output pins.		

- **Online > Add to Display List**

In Diagram view, choose this menu command to display on-line values from an OPC server for the selected port(s), symbol(s), link(s). In effect, the selected item(s) are added to the display list which collects all OPC items to be displayed while **Subscribe for Live Data** is active.

The command can be used while **Subscribe for Live Data** is inactive or active:

- If inactive, no on-line values are retrieved from the OPC server, and added items just display a “*” indicating that they are prepared to display on-line values.
- If active, on-line values are retrieved from the OPC server. In that case, added items display the on-line value retrieved.

- **Online > Remove from Display List**

In Diagram view, choose this menu command to no longer display on-line values for the selected port, symbol, or link. In effect, the selected item(s) are removed from the display list.

- **Online > Add Links to Current Page**

In Diagram view, choose this menu command to display on-line values for all links on the current page. In effect, all links on the current page are added to the display list.

- **Online > Add all Links to Display List**

In Diagram view, choose this menu command to display on-line values for all links available in a Function Diagram. In effect, all links on all pages of the Function Diagram are added to the display list.

- **Online > Remove all from Display List**

In Diagram view, choose this menu command to no longer display on-line values for all links on the current page.

- **Online > Watch Window**

In Diagram view, choose this menu command to open the Watch window in order to

- display and set on-line values in textual form in a list box.
- display and set on-line values for variables of structured data type.
- display and set on-line values from another function diagram
- display and set on-line values from an arbitrary aspect exposing OPC properties, e.g. a CBM application.

Set on-line value means that the OPC value is written once.

- **Online > Subscribe for Live Data**

In Diagram view, choose this toggle menu command to start/stop display of on-line values collected in the diagram's display list and/or collected in the Watch window.

- If checked, Subscribe for Live Data is active. i.e. on-line values are retrieved from the OPC server and displayed in the diagram or in the Watch window.
- If unchecked, Subscribe for Live Data is inactive. i.e. no on-line values are retrieved from the OPC server. According to the setting '**Show Last Value(s)**', the OPC items in the diagram display a '*' or the recent value

retrieved, and the OPC items in the Watch window display an empty value or the recent value retrieved.

- **Online > Subscribe For Live Data All Open Diagrams**

In Diagram view, choose this toggle menu command to start/stop display of on-line values collected in the diagram's display list and/or collected in the Watch window for all open diagrams.

- If checked, Subscribe for Live Data for all open diagrams is active. I.e. on-line values are retrieved from the OPC server and displayed in the diagram or in the Watch window.
- If unchecked, Subscribe for Live Data for all open diagrams is inactive. No on-line values are retrieved from the OPC server. According to the setting '**Show Last Value(s)**', the OPC items in the diagram display a '*' or the recent value retrieved, and the OPC items in the Watch window display an empty value or the recent value retrieved.

- **Online > Show Last Value(s)**

In Diagram view, choose this toggle menu command to show or hide recent on-line values when subscription for live data has stopped.

- If checked, recent values retrieved from the OPC server are still displayed in the diagram and/or in the Watch window.
- If unchecked, the OPC items in the diagram display a '*', and the OPC items in the Watch window display an empty value.

- **Online > Display in Hexadecimal Format**

This menu command displays the online values in hexadecimal format.

- **Online > Force...**

This dialog allows to force the Value (datatype RealIO, BoolIO) of an allocated input IO signal (object type CBM_AIS, CBM_DIS) or the IOValue (datatype RealIO, BoolIO) of an allocated input IO signal (object type CBM_AOS, CBM_DOS).

- **Online > Subscribe for Live Data All Output Ports**

The command adds labels to display live values for all the pages of function diagram and subscribes live data for the current page. The shortcut key for this menu option is F9.

- If selected, Subscribe for Live Data All Output Ports is active. i.e. on-line values are retrieved from the OPC server and displayed on the output pins of the current page.
- If unchecked, Subscribe for Live Data All Output Ports is inactive. i.e. no on-line values are retrieved from the OPC server.

- **Online > Subscribe for Live Data Connected Output Ports**

The command adds labels to display live values for the connected output ports and subscribes live data for the current page. The shortcut key for this menu option is F11.

- If selected, Subscribe for Live Data Connected Output Ports is active. i.e. on-line values are retrieved from the OPC server and displayed on the output pins of the current page.
- If unchecked, Subscribe for Live Data Connected Output Ports is inactive. i.e. no on-line values are retrieved from the OPC server.

Window Menu

Table 31. Description for Window Menu Commands

Menu Command	Shortcut	Description	D/C	Toolbar
Window > Workbook Mode	ALT > W > W	Turns on or off the workbook mode (display with tabs), see Window > Workbook Mode	D/C	
Window > New Window	ALT > W > N	Opens a new document window of an already open designer component or designer diagram. You can display different areas or different pages of a diagram at the same time.	D/C	
Window > Cascade	ALT > W > C	Arranges windows cascading.	D/C	
Window > Tile	ALT > W > T	Arranges windows tiled.	D/C	
Window > Arrange Icons	ALT > W > A	Arranges icons of minimized windows.	D/C	
Window > Close All	ALT > W > I	Close all open windows.	D/C	

- **Window > Workbook Mode**

Choose this menu command to display open windows in workbook mode. In workbook mode, MS-Excel sheet-like tabs are displayed at the bottom of the main window in order to navigate to the component/diagram document of interest by a single mouse click.

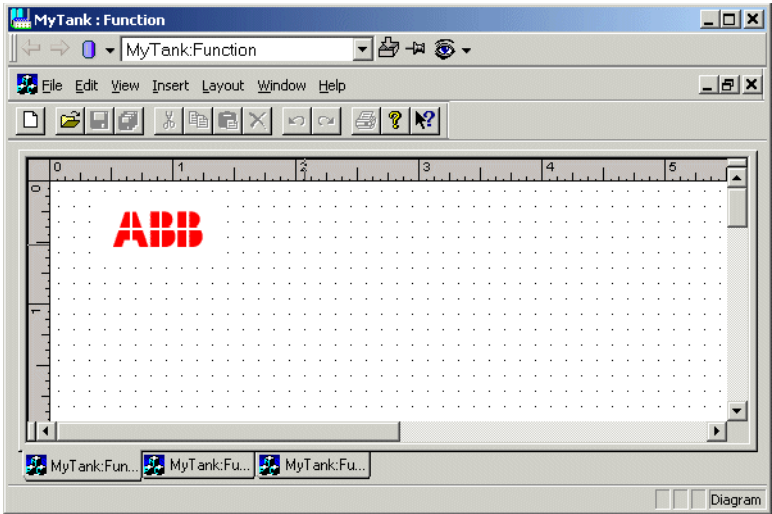



Figure 182. Windows Arranged in Workbook Mode

Help Menu

Table 32. Description for Help Menu Commands

Menu Command	Shortcut	Description	Toolbar
Help > Contents	ALT > H > C	Display on-line help contents, index, search page, and favorites.	Standard
Help > Reference Manual	ALT > H > U	Display this designer reference manual.	Standard
 Help > About	ALT > H > A	Display program information, version number and copyright.	Standard

- Help > Contents**
Choose this menu command to display the contents of designer’s on-line help.
- Help > Reference Manual**
Choose this menu command to display this designer reference manual via Adobe Acrobat Reader.

- **Help > About**

Choose this menu command to display product version and copyright information. Depending on the view (Diagram/Component view), the corresponding **About** dialog is displayed.

Context Menus

On right-click of mouse, context menus are opened according to the selected object/area in designer.

- The context menu of a designer component (right-click on a component) offers related menu options such as Cut, Copy, Paste, Order, and other aspects and verbs of the corresponding Aspect Object.
By that the user can quickly navigate to other aspects of each component from within the designer diagram.
- The diagram's context menu (right-click on the empty diagram background) offers designer menus like Paste, Grid, Zoom, and other aspects and verbs of the corresponding diagram Aspect Object. User can quickly navigate to other aspects of the diagram.
- The diagram's context menu (right-click on the empty diagram background) offers a **Diagram Viewer** option to view the Diagram from the Control Builder editor mode.
- The following options are available as context menus:
 - Allocate/UnAllocate Diagram
 - Define Data Flow Order
 - Generate Configuration Data
 - Generate Configuration Data (Full Build)
 - I/O Allocation



Most context menus are dynamic and the menu items are displayed according to the selected object.

The aspect categories are grouped and a submenu with all related instances is created.

User will find the following object-specific context menus:

- [Designer Aspect Context Menu](#)
- [Diagram/Component Document Context Menu](#)
- [Designer Component Context Menu](#)
- [Graphic Component Context Menu](#)
- [Dockable Window Context Menu](#)
- [Horizontal Scrollbar Context Menu](#)
- [Output Window Context Menu](#)
- [Toolbar Context Menu](#)
- [Vertical Scrollbar Context Menu](#)

Designer Aspect Context Menu

The designer aspect supports three views, see [Diagram/Component View](#):

- **Diagram**
- **Component**
- **Parent Diagram**

The Function aspect supports the additional aspect verbs:

- **Generate Configuration Data.**
- **Generate Configuration Data (Full Build)**
- **Synchronize Names.**
- **Upload Control Properties**
- **XML Type Description** (on object types only).



Only the relevant menu items appear in the context menu.

Right-click a designer aspect to open the context menu:

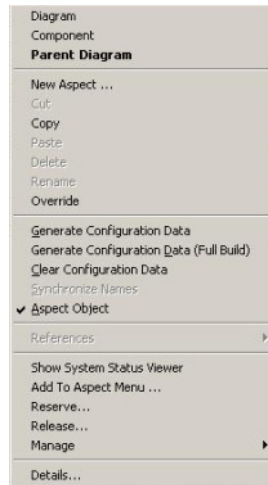


Figure 183. Context Menu of Designer Aspect

A corresponding context menu can be opened on the Allocatable Group aspect. The actual menu items are described in [Table 33](#):

Table 33. Designer Aspect / Allocatable Group Aspect Context Menu Commands

Menu Command	Description
Config View	On Allocatable Group aspect only: Opens Allocatable Group dialog.
Diagram	(1)On Designer aspect only: Opens a new overlap window and displays the designer diagram view of the current selected diagram aspect object, see Diagram/Component View .
Component	1) Opens a new overlap window and displays the designer component view of the current selected component aspect object, see Diagram/Component View .
Parent Diagram	1) Opens a new overlap window, displays the designer diagram view of the parent diagram of the current selected component aspect object, and navigates to the selected component, see Diagram/Component View .
New Aspect	1) Opens the dialog for creating a new aspect of an Aspect Object.
Cut	Cuts the selected aspect and places it on the clipboard.

Table 33. Designer Aspect / Allocatable Group Aspect Context Menu Commands (Continued)

Menu Command	Description
Copy	Copies the selected aspect to the clipboard.
Paste	Pastes the cut/copied aspect to the aspect list.
Delete	Deletes the selected aspect.
Rename	Allows to rename the selected aspect.
Override	If the aspect is inherited, this operation makes a local copy of the aspect.
Generate Configuration Data (Not available in Topology Designer)	Generate the configuration data for the diagram and all child (nested) diagrams. The command generates only diagrams in modified state (red traffic light indicator in the status bar).
Generate Configuration Data (Full Build) (Not available in Topology Designer)	Generate the configuration data for the diagram and all child (nested) diagrams, independent of the diagram state.
Clear Configuration Data	Clears the generated configuration data for the Function Diagram or Allocatable Group in CBM.
Synchronize Names (Not available in Topology Designer)	1) Synchronizes/maps the <code>..Name.Name</code> to <ul style="list-style-type: none"> <code>..Control Builder Name.Name</code>. <code>Name</code> parameter of control modules
Upload Control Properties (Init_Vals) (Not available in Topology Designer)	Uploads initial values from the Control Properties. In a Control Builder M project, the Control Property changes will not be written into Function Diagrams automatically. To get the Control Properties uploaded into a function diagram this aspect verb has to be used. The aspect command is also supported via Bulk Data Manager.
Upload Control Properties(Par_Conns) This menu command is visible only to the users belongs to the Software Developers User Group. (Not available in Topology Designer)	Uploads connection parameters from the Control Properties. In a Control Builder M project, the Control Property changes will not be written into Function Diagrams automatically. To get the Control Properties uploaded into a function diagram this aspect verb has to be used. The aspect command is also supported via Bulk Data Manager.

Table 33. Designer Aspect / Allocatable Group Aspect Context Menu Commands (Continued)

Menu Command	Description
Aspect Object (Not available in Topology Designer)	<p>On components only: Converts the Aspect Object to a Symbol Object.</p> <p>This command is also available on the Function Diagram Variable aspect: It switches all connected diagram references (sources and sinks) from Aspect Object to Symbol Object.</p> <p>In both cases the aspect command is also supported via Bulk Data Manager.</p> <p>Note: Always keep Diagram References and their Diagram Variables as symbol object only.</p>
References	If the Aspect Object has cross-references, for example the Aspect Object could be included in a graphic display, the display's name is shown when you select this item
Show System Status Viewer	Displays the system status for the diagram.
Add To Aspect Menu...	Add the selected diagram to Aspect Favorites.
Reserve	Allows to reserve the diagram.
Release	Allows to reserve the diagram.
Manage	Allows to View (Reserve, Release), Deploy and Update the Function Diagram.
Details	<p>Opens the designer aspect's Details dialog.</p> <p>The dialog consists of the following pages: Identification, Property View, Name Tool, Names & Paths, Permission, Lock Status, References.</p>

Diagram/Component Document Context Menu

In Component view or Diagram view, right-click any free area in a document window to open the following context menu:

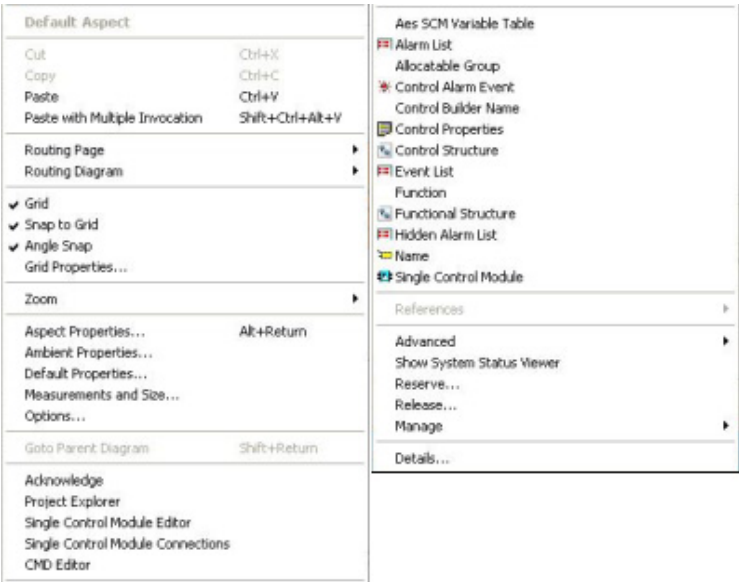


Figure 184. Diagram/Component Document Context Menu

The menu items are described in [Table 34](#):

Table 34. Diagram/Component Document Context Menu Commands

Menu Command	Description
Default Aspect or <Aspect>	If this menu item appears as Default Aspect (disabled), it indicates that no Default Aspect is defined for the current selected Aspect Object If this menu item appears as <Aspect> (for example, Function , Name , and so on), it navigates directly to the default <Aspect>-specific view
Cut	Cuts the Diagram/Component document and copies to the Clipboard
Copy	Copies the Diagram/Component document to the Clipboard.
Paste	Equals menu command Edit > Paste .

Table 34. Diagram/Component Document Context Menu Commands (Continued)

Menu Command	Description
Paste with Multiple Invocation	Equals menu command Edit > Paste with Multiple Invocation .
Routing Page > Freeze Routing	Disables auto routing capability of all link components on the current page.
Routing Page > Unfreeze Routing	Enables auto routing capability of all link components on the current page.
Routing Page > Reroute	Enables auto routing capability of all link components on the current page and all the link components of the current page get rerouted.
Routing Diagram > Freeze Routing	Disables auto routing capability of all link components of the diagram.
Routing Diagram > Unfreeze Routing	Enables auto routing capability of all link components of the diagram.
Grid	Equals menu command View > Grid .
Snap to Grid	Equals menu command View > Snap to Grid .
Angle Snap	
Grid Properties	Equals menu command View > Grid Properties .
Zoom > 50%	Equals menu command View > Zoom Percent > 50% .
Zoom > 75%	Equals menu command View > Zoom Percent > 75% .
Zoom > 100%	Equals menu command View > Zoom Percent > 100% .
Zoom > 200%	Equals menu command View > Zoom Percent > 200% .
Zoom > Zoom to Fit	Equals menu command View > Zoom to Fit .
Aspect Properties	Equals menu command Edit > Aspect Properties .
Ambient Properties	Equals menu command Edit > Ambient Properties .
Default Properties	Equals menu command Edit > Default Properties .
Measurements and Size	Equals menu command Edit > Measurements and Size .
Options ⁽¹⁾	Equals menu command Edit > Options .
<Aspect>	Navigates to the corresponding <Aspect> of the current Aspect Object.

Table 34. Diagram/Component Document Context Menu Commands (Continued)

Menu Command	Description
References	If the Aspect Object has cross-references, for example the Aspect Object could be included in a graphic display, the display's name is shown when you select this item
Advanced	Engineering Workplace only: Open Engineering Workplace Advanced menu with some useful commands for I/O Allocation, Bulk Data Manager, Document Manager, and Structure operations.
Details	Opens the Details dialog of the selected aspect object.

(1) All menu items from here on are part of the dynamic System 800xA Aspect Object menu.

Designer Component Context Menu

Right-click a designer component to open the following context menu:

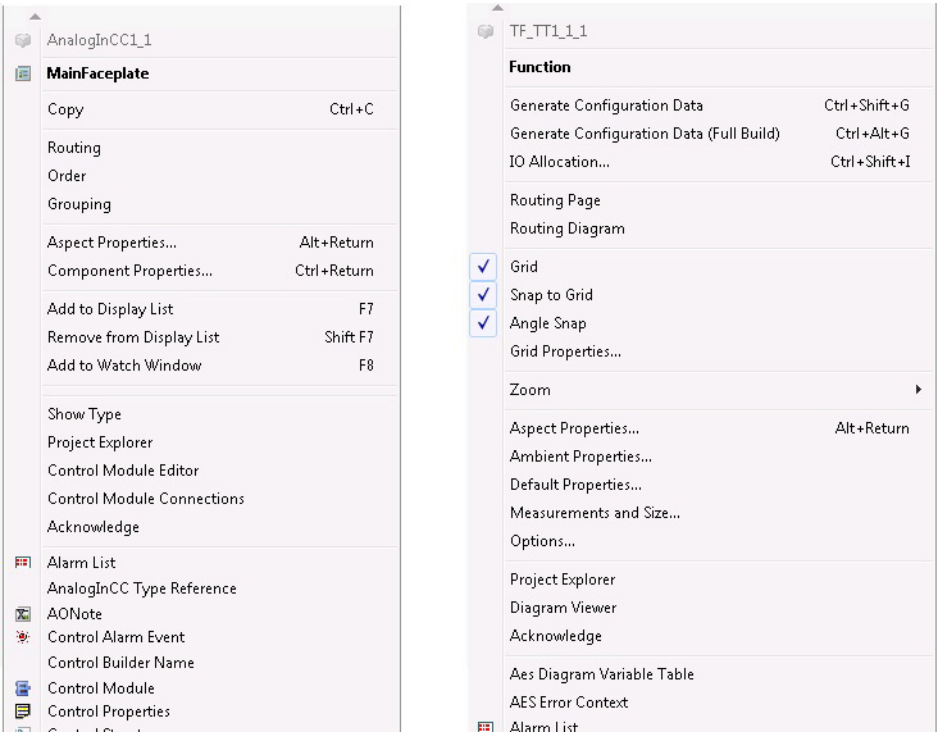


Figure 185. Designer Component Context Menu

The menu items are described in [Table 35](#):

Table 35. Designer Component Context Menu Commands

Menu Command	Description
Default Aspect or <Aspect>	If this menu item appears as Default Aspect (disabled), it indicates that no Default Aspect is defined for the current selected Aspect Object If this menu item appears as <Aspect> (for example, Function , Name , and so on), it navigates directly to the default <Aspect>-specific view.
Cut	Equals menu command Edit > Cut .
Copy	Equals menu command Edit > Copy .
Paste	Equals menu command Edit > Paste .
Delete	Equals menu command Edit > Delete .
Order > Bring Forward	Equals menu command Layout > Order > Bring Forward .
Order > Bring to Front	Equals menu command Layout > Order > Bring to Front .
Order > Send Backward	Equals menu command Layout > Order > Send Backward .
Order > Send to Back	Equals menu command Layout > Order > Send to Back .
Grouping > Group	Equals menu command Layout > Group .
Grouping > Ungroup	Equals menu command Layout > Ungroup .
Add to Display List (Not available in Topology Designer)	Equals menu command Online > Add to Display List .
Remove From Display List (Not available in Topology Designer)	Equals menu command Online > Remove from Display List .
Allocation Grouping (Not available in Topology Designer)	Equals menu command Allocation > Allocation Grouping .
Aspect Properties	Equals menu command Edit > Aspect Properties .
Component Properties	Equals menu command Edit > Default Properties .
Aspect Object (Not available in Topology Designer)	Toggles instance kind aspect object <-> symbol object.

Table 35. Designer Component Context Menu Commands (Continued)

Menu Command	Description
Show Hidden Ports (Not available in Topology Designer)	This menu item is enabled only for components with hidden ports.
Number of Inputs (Not available in Topology Designer)	This menu item is enabled only for components with variable (extensible) number of parameters,
Change Type (Not available in Topology Designer)	This menu item is enabled only for symbol objects, not for aspect objects.
Variable Properties (Not available in Topology Designer)	This menu item is enabled only for variables, diagram references, and diagram parameters.
Connection Mapping (Not available in Topology Designer)	Displays the Connection Mapping dialog in order to map connections between different structured data types.
Goto Child Diagram	Equals menu command View > Goto Child Diagram .
Goto Parent Diagram	Equals menu command View > Goto Parent Diagram .
Goto Reference	Opens the dialog to select and navigate to an off-diagram reference on another diagram.
Connect To Diagram Variable	Opens the dialog to define a connection to an off-diagram reference on another diagram.
Disconnect from Diagram Variable	Opens the dialog to remove a connection to an off-diagram reference on another diagram.
Number of Diagram References	Opens the Number of Diagram References dialog to define the maximum number of diagram references displayed in one column. This menu item is enabled only for diagram references.
Show Type ⁽¹⁾	Navigates to the object type in Object Type Structure
<Aspect>	Opens the <Aspect> dialog for the current Aspect Object.
References	If the Aspect Object has cross-references, for example the Aspect Object could be included in a graphic display, the display's name is shown when you select this item

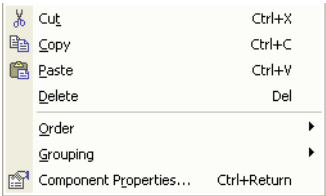
Table 35. Designer Component Context Menu Commands (Continued)

Menu Command	Description
Advanced	Engineering Workplace only: Open Engineering Workplace Advanced menu with some useful commands for I/O Allocation, Bulk Data Manager, Document Manager, and Structure operations.
Properties	Opens the Properties dialog for the selected aspect object, if any.

(1) All menu items from here on are part of the dynamic System 800xA Aspect Object menu.

Graphic Component Context Menu

Right mouse click on a graphic component (e.g. a primitive) to open the following context menu:

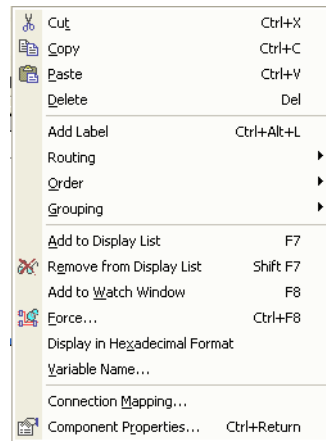


The menu items are described in [Table 36](#).

Link Component Context Menu

Note: Not available in Topology Designer.

Right-click a link component that connects two symbol ports to access the following context menu:



The menu items are described in [Table 36](#).

Symbol Port Context Menu

Note: Not available in Topology Designer.

Right-click a symbol port to access the following context menu:

Connect to PidCC1_1:Pv	Ctrl -
Hide Port(s)	Alt+Down
Aspect Properties...	Alt+Return
New Diagram Input Reference...	Ctrl+Alt+R
New Variable...	Ctrl+Alt+V
New IO Signal AIS...	Ctrl+Alt+I
New Joiner	Ctrl+Alt+J
Disconnect	Ctrl+Shift -
Disconnect All	Ctrl+Alt -
Add to Display List	F7
Remove from Display List	Shift F7
Add to Watch Window	F8
Force...	Ctrl+F8
Display in Hexadecimal Format	

The menu items are described in [Table 36](#):

Table 36. Graphic Component, Link, Port Context Menu Commands

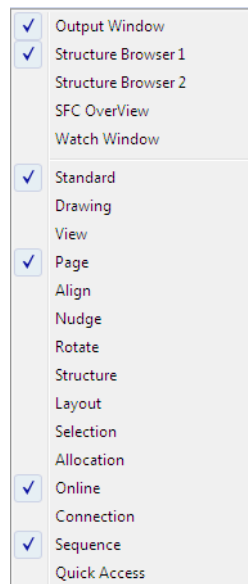
Menu Command	Description
Cut	Equals menu command Edit > Cut .
Copy	Equals menu command Edit > Copy .
Paste	Equals menu command Edit > Paste .
Delete	Equals menu command Edit > Delete .
Routing > Freeze Routing	Disables auto routing capability of link component.
Routing > Unfreeze Routing	Enables auto routing capability of link component.
Routing > Reroute	Enables auto routing capability of link component and the link component gets rerouted.
Order > Bring Forward	Equals menu command Layout > Order > Bring Forward .
Order > Bring to Front	Equals menu command Layout > Order > Bring to Front .
Order > Send Backward	Equals menu command Layout > Order > Send Backward .
Order > Send to Back	Equals menu command Layout > Order > Send to Back .
Grouping > Group	Equals menu command Layout > Group .
Grouping > Ungroup	Equals menu command Layout > Ungroup .
Add To Display List (Not available in Topology Designer)	Equals menu command Online > Add to Display List . (Links and ports only).
Remove from Display List (Not available in Topology Designer)	Equals menu command Online > Remove from Display List . (Links and ports only).
Variable Name (Not available in Topology Designer)	Opens the dialog Variable Name in order to check the mapped and generated variable name (Links only).
Connection Mapping (Not available in Topology Designer)	Opens the Connection Mapping dialog in order to map connections between structured data types.

Table 36. Graphic Component, Link, Port Context Menu Commands (Continued)

Menu Command	Description
Component Properties	Equals menu command Edit > Default Properties .
Hide Port(s) (Not available in Topology Designer)	Hides the selected port(s)
Aspect Properties	Equals menu command Edit > Aspect Properties (Ports only).
New Action	Adds new action to a step of a SPL diagram.

Toolbar Context Menu

Right-click the Menu/Toolbar area to access the following context menu:



The menu items are described in [Table 37](#):

Table 37. Toolbar Context Menu Commands

Menu Command	Description
Output Window	Show/Hide the Output Window as either a docked window or as a floating popup window. The Output Window lists results from the File > Check and File > Generate Configuration Data (Full Build) commands.
Structure Browser 1	Show/Hide the Structure Browser 1 Window as either a docked window or as a floating popup window. The Structure Browser Window can be used as an alternative to Plant Explorer's object browser window.
Structure Browser 2	Show/Hide the Structure Browser 2 Window as either a docked window or as a floating popup window. The Structure Browser Window can be used as an alternative to Plant Explorer's object browser window.
SFC Overview	Click SFC Overview to access the details of all sequences and sequence components (steps, transition, and action) that are available in a Function Diagram. In case of nested diagrams, open the relevant function Diagram to navigate using the sequence tree structure.
Watch Window	Show/Hide the Watch Window as either a docked window or as a floating popup window. The Watch Window displays and sets on-line values from any OPC server
Standard	Show/Hide the Standard toolbar with commands New, Open, Save, Print...
Drawing	Show/Hide the Drawing toolbar with commands Select, Line, Rectangle, Port, Text, Label...
View	Show/Hide the View toolbar with commands Layer, Zoom, Pan, Grid.
Page	Show/Hide the Page toolbar with commands New Page, Delete Page, Goto Page.
Align	Show/Hide the Align toolbar with commands Align Top, Middle, Left,,
Nudge	Show/Hide the Nudge toolbar with commands Nudge Up, Down, Left, Right.
Rotate	Show/Hide the Rotate toolbar with commands Rotate and Flip.
Structure	Show/Hide the Structure toolbar with commands Group, Ungroup, Bring to Front, Send to Back.
Layout	Show/Hide the Layout toolbar with commands Space Across, Same Width...
Selection	Show/Hide the Selection toolbar with commands for multi selection and network/symbol traversal.
Allocation (Not available in Topology Designer)	Show/Hide the Allocation toolbar. with commands Select/Create/Delete Allocatable Group, Define Data Flow Order.

Table 37. Toolbar Context Menu Commands (Continued)

Menu Command	Description
Online	Show/Hide the Online Menu toolbar with commands to Add links / Remove links etc.
Connection	Show/Hide the Connection toolbar with commands Connect, Disconnect, Invert.
Sequence	Show/Hide the Sequence toolbar with commands: Insert Transition, Insert Step, Insert Jump, Create Multiple Transitions / Steps, Renumber, Enables and disables execution of the actions associated with the step, Forces next step to become active, Forces previous step to become active.

Vertical Scrollbar Context Menu

Right-click a vertical scroll bar to access the following context menu:



The menu items are described in [Table 38](#).

Table 38. Vertical Scrollbar Context Menu Commands

Menu Command	Description
Scroll Here	Moves the scroll bar to the current cursor location.
Top	Moves the scroll bar to the top.
Bottom	Moves the scroll bar to the bottom.
Page Up	Moves the scroll bar one page division up. Same as click above the scroll bar.
Page Down	Moves the scroll bar one page division down. Same as click below of scroll bar.
Scroll Up	Moves the scroll bar one division up. Same as click on up arrow.
Scroll Down	Moves the scroll bar one division down. Same as click on down arrow.

Horizontal Scrollbar Context Menu

Right-click right a horizontal scrollbar to access the following context menu:



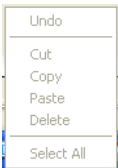
The menu items are described in [Table 39](#).

Table 39. Horizontal Scrollbar Context Menu Commands

Menu Command	Description
Scroll Here	Moves the scroll bar to the current cursor location.
Left Edge	Moves the scroll bar to the left edge.
Right Edge	Moves the scroll bar to the right edge.
Page Left	Moves the scroll bar one page division left. Same as click left of scroll bar.
Page Right	Moves the scroll bar one page division right. Same as click right of scroll bar.
Scroll Left	Moves the scroll bar one division left. Same as click on left arrow.
Scroll Right	Moves the scroll bar one division right. Same as click on right arrow.

Output Window Context Menu

Right-click the **Output Window**’s **Output** or **Build** page to access the following context menu:



The menu items are described in [Table 40](#).

Table 40. Output Window Context Menu Commands

Menu Command	Description
Undo	Equals menu command Edit > Undo .
Cut	Equals menu command Edit > Cut .
Copy	Equals menu command Edit > Copy .
Paste	Equals menu command Edit > Paste .
Delete	Equals menu command Edit > Delete .
Select All	Equals menu command Edit > Select All .

Dockable Window Context Menu

Right-click a **Structure Browser**'s or **Output Window**'s border/tab to access the following context menu:



The menu items are described in [Table 41](#):

Table 41. Dockable Window Context Menu Commands

Menu Command	Description
Allow Docking	Allows docking of the window/toolbar within the main window.
Hide	Hides the window/toolbar,
Clear Window	Clear the window contents (only available for some dockable windows, for example the Output window)

Working with Toolbars

User can get access to different useful commands by clicking the appropriate button using the [Toolbar Categories](#). These commands are mostly accessible through an

appropriate menu command, too. Use the [Customize](#) dialog to turn on or off the display of toolbars, or to create user-defined toolbars.

User can have different toolbar settings in the Diagram view and the Component view, in the preview window and overlap (separate) window.

Customize

Use the **Customize** dialog for:

- [Showing/Hiding Toolbars](#)
- [Creating a User-Defined Toolbar](#)
- [Adding Tool Buttons to Toolbars](#)
- [Deleting a User-Defined Toolbar](#)
- [Changing the Look of Toolbars](#)
- [Showing Tooltips](#)

Open this dialog through the menu command [View > Toolbar > Customize](#). The dialog contains the following two pages:

- [Toolbars](#)
- [Toolbar Commands](#)

Toolbars

Click the **Toolbars** tab of the **Customize** dialog to display the **Toolbars** page.

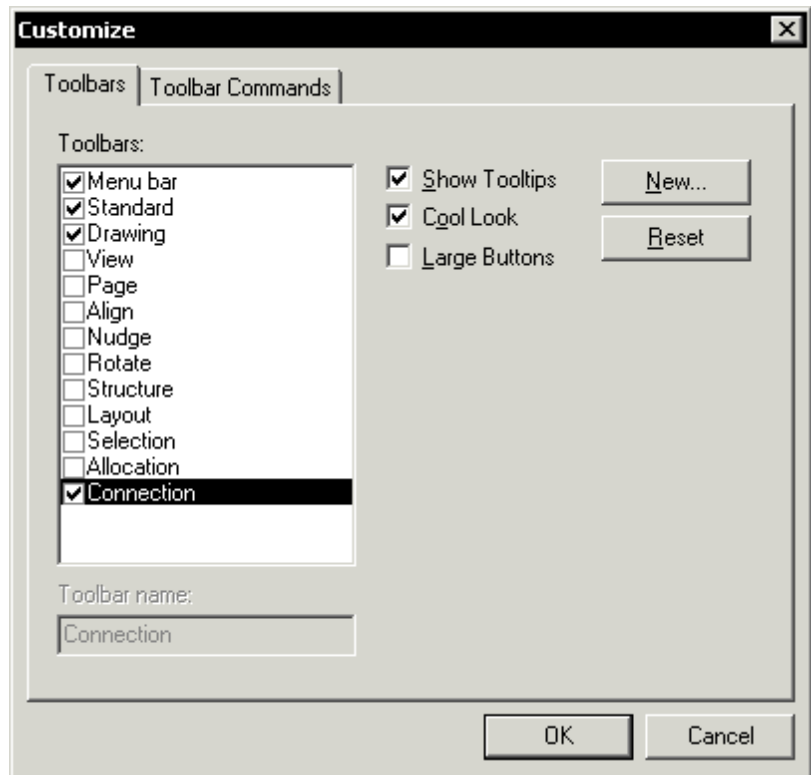


Figure 186. Customize Dialog: Toolbars Page

The **Toolbars** page of the **Customize** dialog contains the following fields, buttons, and other items:

- Toolbars (tab)
- Toolbars
- Toolbar name
- Show Tooltips

- [Cool Look](#)
- [Large Buttons](#)
- [New](#)
- [Reset](#)
- [Delete](#)

- **Toolbars (tab)**

Click the **Toolbars** tab to display the **Toolbars** page with all available toolbars.

For more details, refer to [Working with Toolbars](#).

- **Toolbars**

In the **Toolbars** list box, all available toolbars are listed. You can select multiple toolbars to display them in the application's main window.

Select or clear the appropriate check boxes (except the **Menu bar** which is protected against un-marking) to turn on or off the display. See also [Toolbar Categories](#) and [Showing/Hiding Toolbars](#).

- **Toolbar name.**

The **Toolbar name** field displays the name of the toolbar selected in the **Toolbars** list box. If a user-defined toolbar is selected, the name can be changed here. See also [Renaming a User-Defined Toolbar](#).

- **Show Tooltips**

Select or clear this check box to turn on or off the display of the tooltips. The tooltips are short context-sensitive description texts displayed in small popup boxes with a yellow background if you point to a button of a toolbar. See also [Showing Tooltips](#).

- **Cool Look**

Select or clear this check box to turn on or off the display of the tool buttons in the Cool Look manner. See also [Changing the Look of Toolbars](#).

- **Large Buttons**

Select or clear this check box to turn on or off the display of large buttons. See also [Changing the Look of Toolbars](#).

- **New**

Click this button to open the New Toolbar dialog for creating a new user-defined toolbar. See also

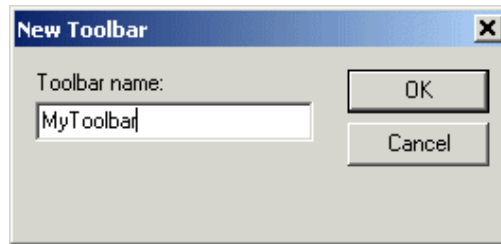


Figure 187. New Toolbar Dialog

Creating a User-Defined Toolbar

- **Reset**

Click this button to reset the layout of the current selected system toolbar to the original state. See also [Adding Tool Buttons to Toolbars](#).

- **Delete**

Click this button to delete the current selected user-defined toolbar. See also [Deleting a User-Defined Toolbar](#).

Toolbar Commands

Click on the **Toolbar Commands** tab of the Customize dialog to display the **Toolbar Commands** page.

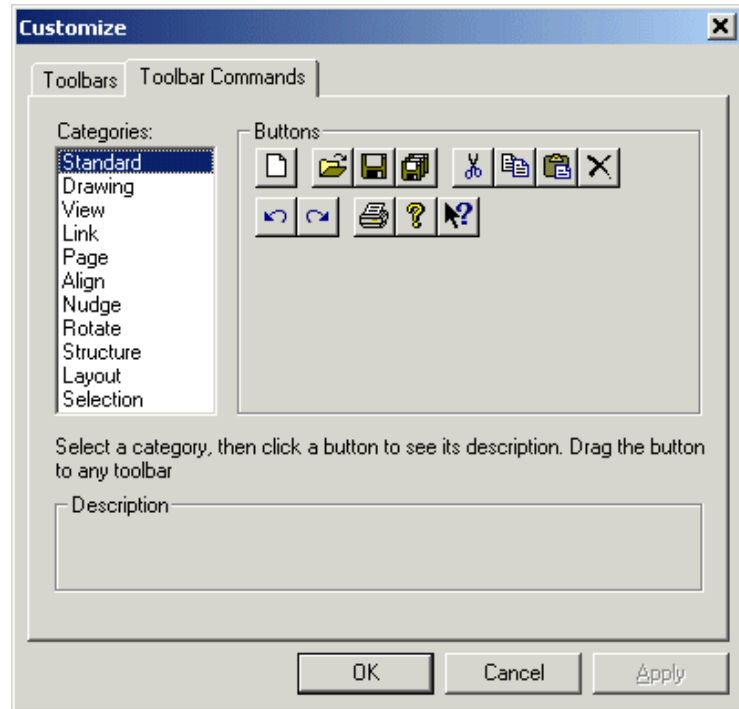


Figure 188. Customize Dialog, Toolbar Commands Page

- The **Toolbar Commands** property sheet of the [Customize](#) dialog contains the following fields, buttons, and other items: [Toolbar Commands](#)
- [Categories](#)
- [Buttons](#)

Description

- **Toolbar Commands**

Click this tab to display the **Toolbar Commands** page for getting access to the tool buttons (commands) in the **Buttons** area. From here, you can copy tool buttons to each existing toolbar. See also [Adding Tool Buttons to Toolbars](#).

- **Categories**

Displays all available [Toolbar Categories](#) delivered with the Function Designer. Select the wanted toolbar to display the buttons belonging to it. See also [Adding Tool Buttons to Toolbars](#).

- **Buttons**

Displays the tool buttons of the currently selected toolbar.

- **Description**

Displays the description of the currently selected tool button.

Toolbar Categories

The designer offers the following toolbars (toolbar categories):

- [Standard](#)
- [Drawing](#)
- [View](#)
- [Link](#)
- [Page](#)
- [Align](#)
- [Nudge](#)
- [Rotate](#)
- [Structure](#)
- [Layout](#)
- [Selection](#)

- [Standard](#)

- **Standard**

The [Standard](#) toolbar contains commands for handling the whole diagram like saving and printing, and it contains commands for handling components like copying and saving. You will find the most commands also as menu commands of the **File** and **Edit** menu.

- **Drawing**

The [Drawing](#) toolbar contains commands for inserting graphic objects (primitives, symbols, and so on) and their handling within the diagram (e.g. snapping to grid).

You will find the most commands also as menu commands of the **Insert** and **View** menu.

- **View**

The [View](#) toolbar contains commands for the layer handling and for zooming the diagram's/component's display. You will find the most commands also as menu commands of the **View** menu.

- **Page**

The [Page](#) toolbar contains commands for creating pages and for navigation between pages. You will find the command for creating a page also as menu command in the **Insert** menu.

- **Align**

The [Align](#) toolbar contains commands for aligning components within the diagram. You will find the commands also as menu commands of the **Layout** menu.

- **Nudge**

The [Nudge](#) toolbar contains commands for nudging components within the diagram. You will not find the commands also as menu commands.

- **Rotate**

The [Rotate](#) toolbar contains commands for rotating and flipping components. You will find the commands also as menu commands of the **Layout** menu.

- **Structure**

The **Structure** toolbar contains commands for moving components to back or to front, and to group components. You will find the commands also as menu commands of the **Layout** menu.

- **Layout**

The **Layout** toolbar contains commands for spacing and sizing components. You will find the commands also as menu commands of the **Layout** menu.

- **Selection**

The **Selection** toolbar contains commands for choosing the selection mode of components. You will not find the commands as menu commands but it is possible to execute these commands via keys.

- **Allocation**

Note: Not available in Topology Designer.

The **Allocation** toolbar contains commands for allocating function components and defining their data flow order. You will find the commands also as menu commands of the **Allocation** menu.

- **Online**

Note: Not available in Topology Designer.

The **Online** toolbar contains commands for viewing and forcing online values (OPC Properties). You will find the commands also as menu commands of the **Online** menu.

- **Connection**

The **Connection** toolbar contains commands for connecting, disconnecting, and inverting ports.

- **Sequence**

Note: Not available in Topology Designer.

The **Sequence** toolbar contains commands for editing the overview diagram of a sequence and forcing steps and transitions.

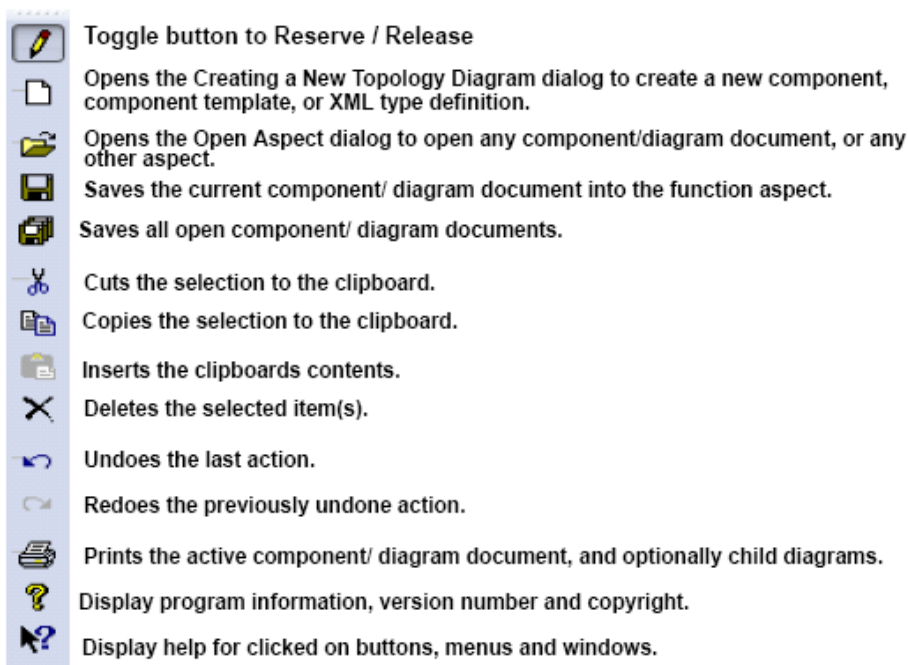
- **Quick Access**

Note: Not available in Topology Designer.

The [Quick Access](#) toolbar can be used to insert a symbol, allocate / unallocate diagrams, generate configuration data, and subscribe for live data.

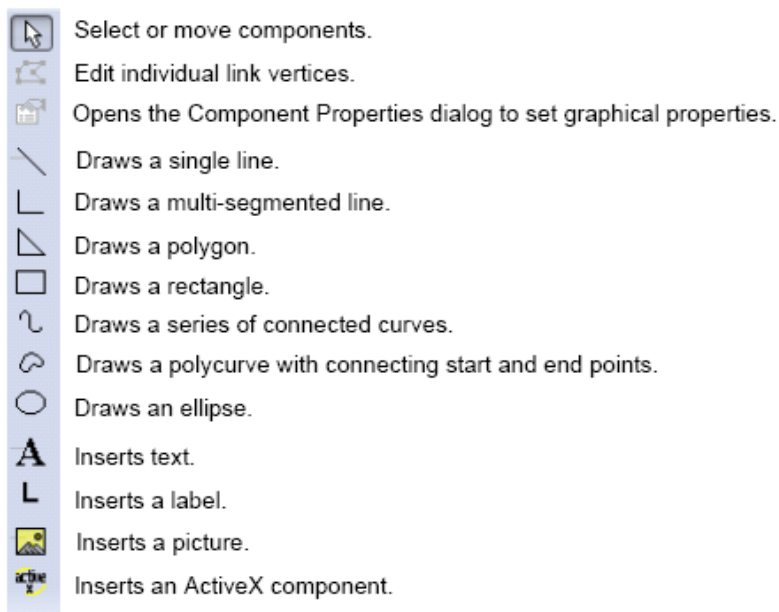
Standard

The **Standard** toolbar of the Toolbar Categories contains the following tool buttons:



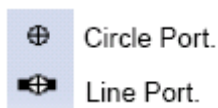
Drawing

The **Drawing** toolbar of the Toolbar Categories contains the following tool buttons:



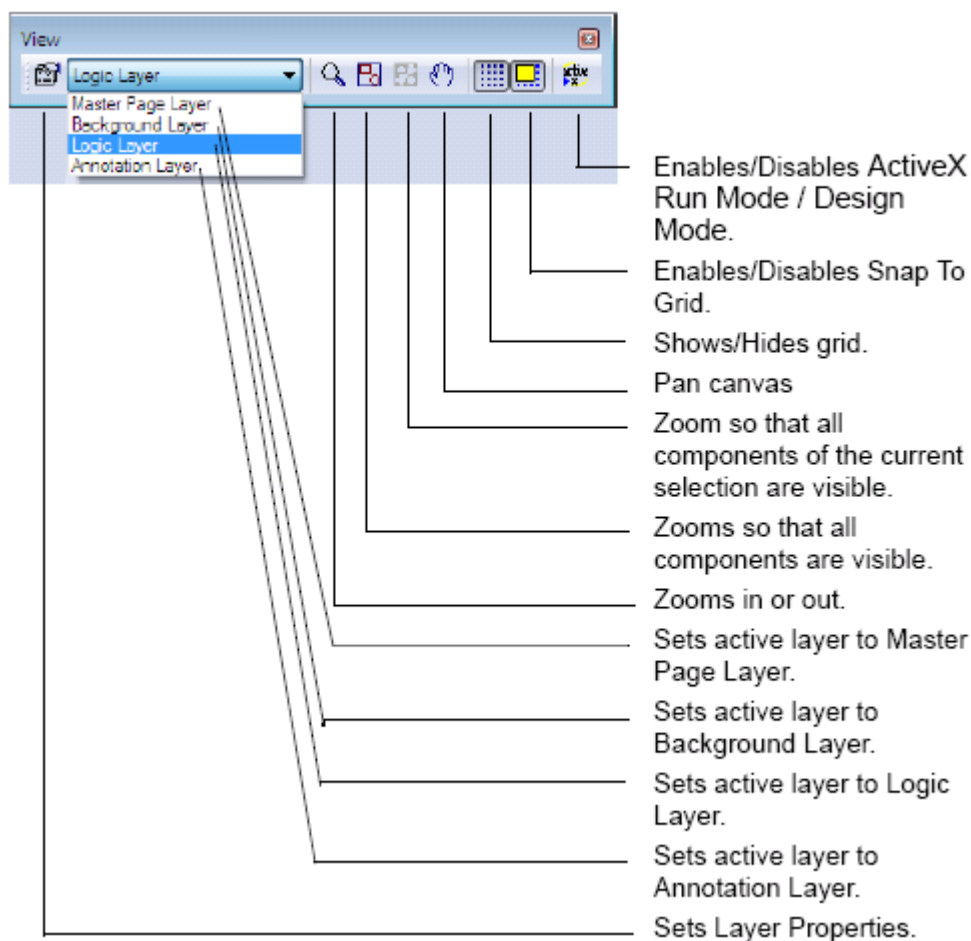
Port

The **Port** toolbar of the Toolbar Categories contains the following tool buttons:



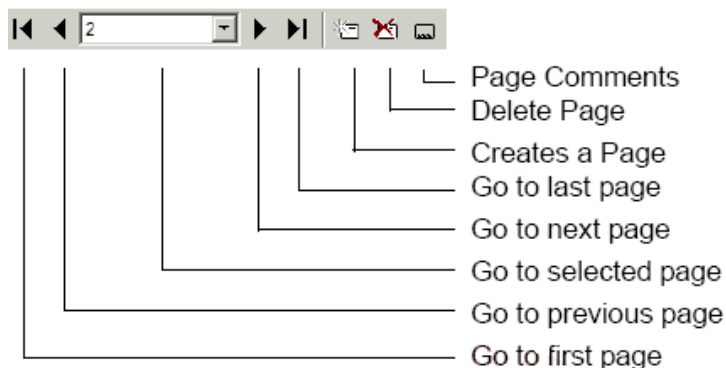
View

The **View** toolbar of the Toolbar Categories contains the following tool buttons:

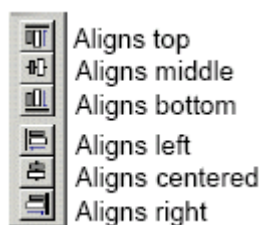


Page

The **Page** toolbar of the Toolbar Categories contains the following tool buttons:

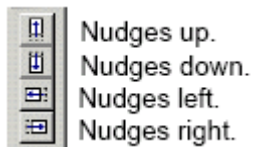
**Align**

The **Align** toolbar of the Toolbar Categories contains the following tool buttons:



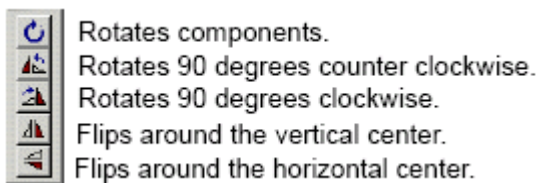
Nudge

The **Nudge** toolbar of the Toolbar Categories contains the following tool buttons:



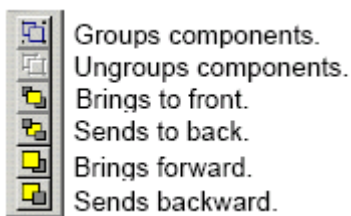
Rotate

The **Rotate** toolbar of the Toolbar Categories contains the following tool buttons:



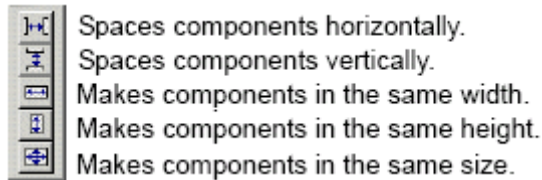
Structure

The **Structure** toolbar of the Toolbar Categories contains the following tool buttons:



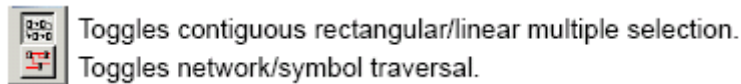
Layout

The **Layout** toolbar of the Toolbar Categories contains the following tool buttons:



Selection

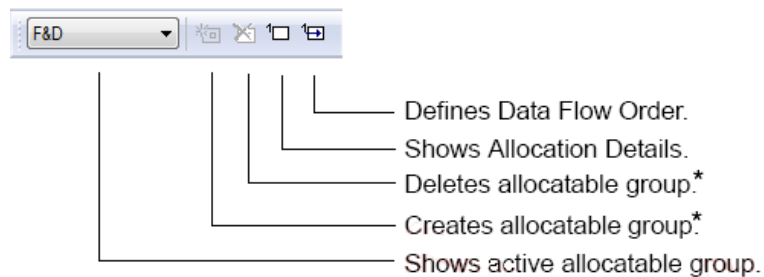
The **Selection** toolbar of the Toolbar Categories contains the following tool buttons:



Allocation

Note: Not available in Topology Designer.

The **Allocation** toolbar of the Toolbar Categories contains the following tool buttons:




* The toolbar options to delete or create allocatable groups are available only for Function Diagrams restored to system version 5.1 and later versions.


Online

Note: Not available in Topology Designer.


The **Online** toolbar of the Toolbar Categories contains the following tool buttons:




Subscribe for Live Data.




Subscribe for Live Data For All Diagrams




Add to Display List.




Remove from Display List.




Adds Links to Current Page.




Add all links to Display List.




Removes all links from Display List.




Forces OPC properties.



Shows the Watch Window.



Subscribe for Live Data All Output Ports.



Subscribe for Live Data Connected Output Ports.

Connection

The **Connection** toolbar of the Toolbar Categories contains the following tool buttons:

1234

✓

✗

↺

✗

✗

Disconnects complete network.

Disconnects selected port only.

Inverts selected port.

Cancels connect string.

Enters connect string.

Connect string combo box

Sequence

The **Sequence** toolbar of the Toolbar Categories contains the following tool buttons:

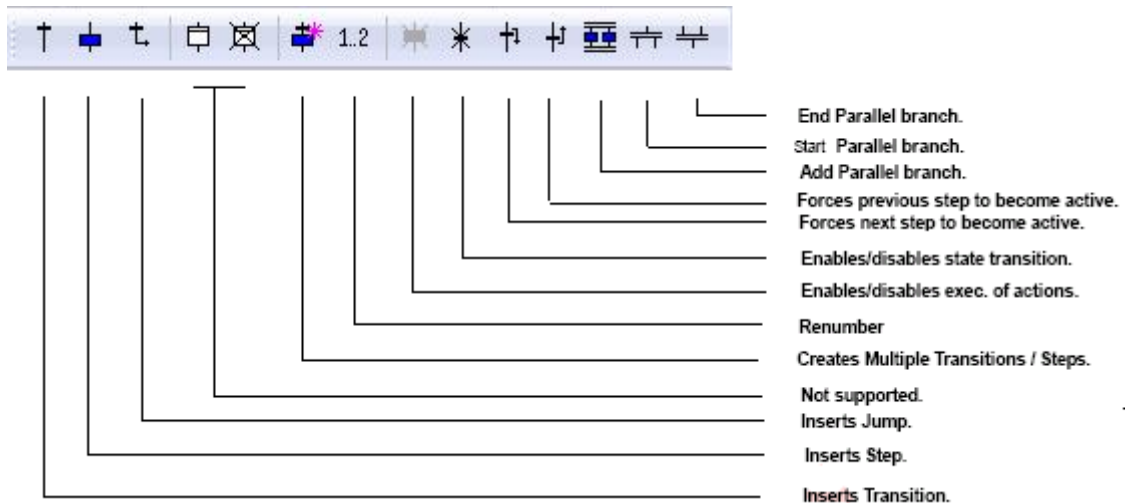


Figure 162. SPL Sequence Toolbar



SFC in Function Designer does not support the following:

- Reset.
- Hold.
- Disable all Actions.

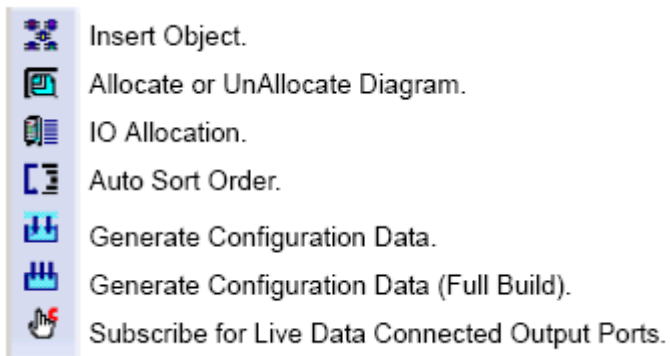


The following features are not supported in Function Diagram:

- Creating subsequence.
- Setting the priority to execute different branches.

Quick Access

The Quick Access toolbar of the Toolbar Categories contains the following tool buttons:



The **Standard**, **Page**, **Online**, and **Quick Access** toolbars are the default toolbars that are available on opening a Function Diagram for the first time in a newly created 800xA 5.1 system or a restored system.

SFC Overview

Right-click the menubar / toolbar area and click **SFC OverView** from the available context menu, as shown in [Figure 163](#).

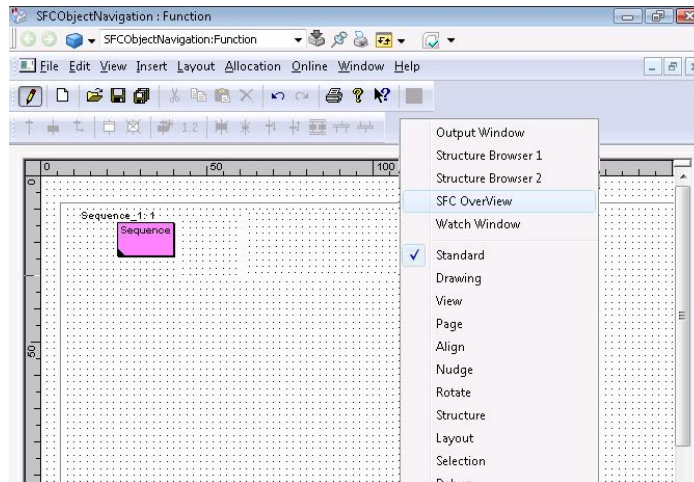


Figure 163. SFC Overview Menu Option

A new dockable window that contains the sequence tree structure appears.

User can navigate through the sequence tree structure and obtain details regarding all the sequences and sequence components (steps, transition, and action) available in that particular Function Diagram (refer to [Figure 164](#)). The Function Diagram containing the sequence is automatically saved on expanding the sequence tree structure.

In case of nested diagrams, open the relevant Function Diagram to navigate using the sequence tree structure.

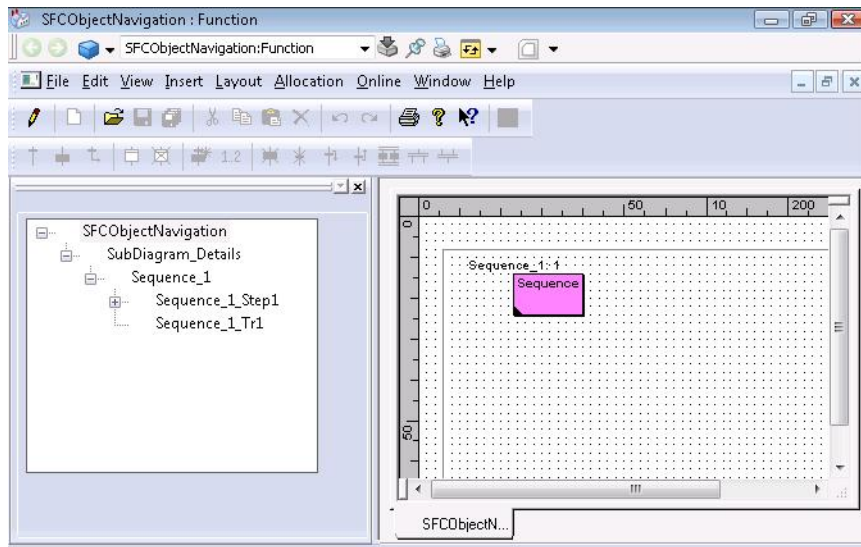
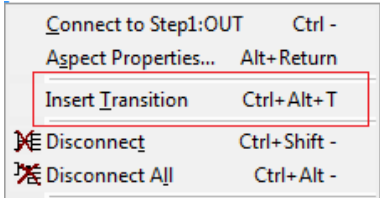
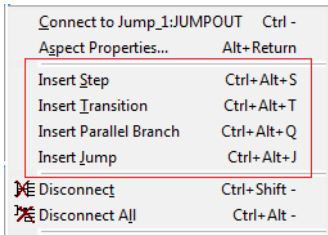
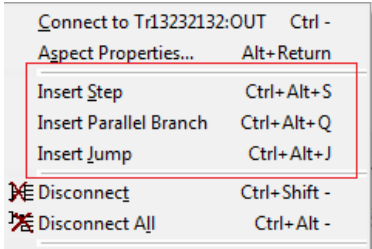


Figure 164. SFC Overview

User can also click the desired sequence component to access it directly.

Right-click a component in the sequence diagram to access the SPL context menu. The context menu is dynamic and provides options to insert steps, transitions, parallel branches, and jump. The context menu varies with the selected component, as provided in [Table 42](#).

Table 42. SPL Context Menus

Selected Component	Available Context Menu
Step	
Jump	
Transition	



When a module is inserted through the context menu, the connection between the active port and the inserted module is automatically created.

Table 43 provides the description for the SPL context menu commands.

Table 43. SPL Context Menu Command Description

Menu Command	Shortcut	Description
Insert <u>S</u> tep	Ctrl+Alt+S	Inserts a step in the sequence diagram.
Insert <u>T</u> ransition	Ctrl+Alt+T	Inserts a transition in the sequence diagram.
Insert Parallel <u>B</u> ranch	Ctrl+Alt+Q	Inserts a parallel branch in the sequence diagram.
Insert <u>J</u> ump	Ctrl+Alt+J	Inserts a jump in the sequence diagram.

Showing/Hiding Toolbars

To display the required toolbars in the main window, do as follows:

1. Open the **Customize** dialog.
2. Select the required toolbars for display:
 - To turn on the display of a toolbar, select the appropriate check box.
 - To turn off the display of a toolbar, clear the appropriate check box.
3. Click **OK**.

Alternatively, to show or hide a toolbar, do as follows:

1. Right-click the blank area of the menu bar or toolbar area. A pop-up menu will appear, displaying all available toolbars.
2. To turn on or off the display of a toolbar, click on it. All marked toolbars will be displayed.

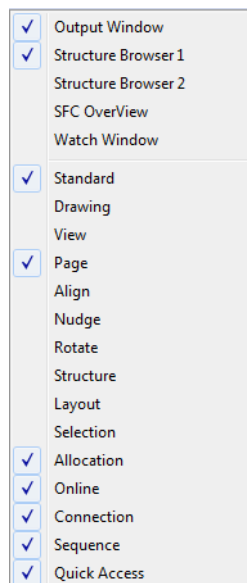


Figure 167. Context Menu for Dockable Toolbar/Window Selection



If different toolbars are arranged in a single row, some of the toolbar options appear beyond the active window and will not be visible to the user. It is recommended to avoid arranging the different toolbars in a single row.

Creating a User-Defined Toolbar

To create a new, user-defined toolbar, do as follows:

1. Open the **Customize** dialog.
2. Click **New** on the **Toolbar** tab. The **New Toolbar** dialog appears.
3. Enter the name for the new toolbar in the **Toolbar name** field.
4. Click **OK**.



The new (yet empty) toolbar is created as separate window on the application's main window with the toolbar name as window title. For adding tools (buttons) to the toolbar, refer to [Adding Tool Buttons to Toolbars](#).

Adding Tool Buttons to Toolbars

To add tool buttons to a toolbar, do as follows:

1. Open the **Customize** dialog.
2. Click on the **Toolbar Commands** tab.
3. Select the toolbar with the wanted tool button from the **Categories** list box.
4. Drag-and-drop the wanted tool button from the **Buttons** group box to the target toolbar out of the **Customize** dialog.
5. Click **OK**.

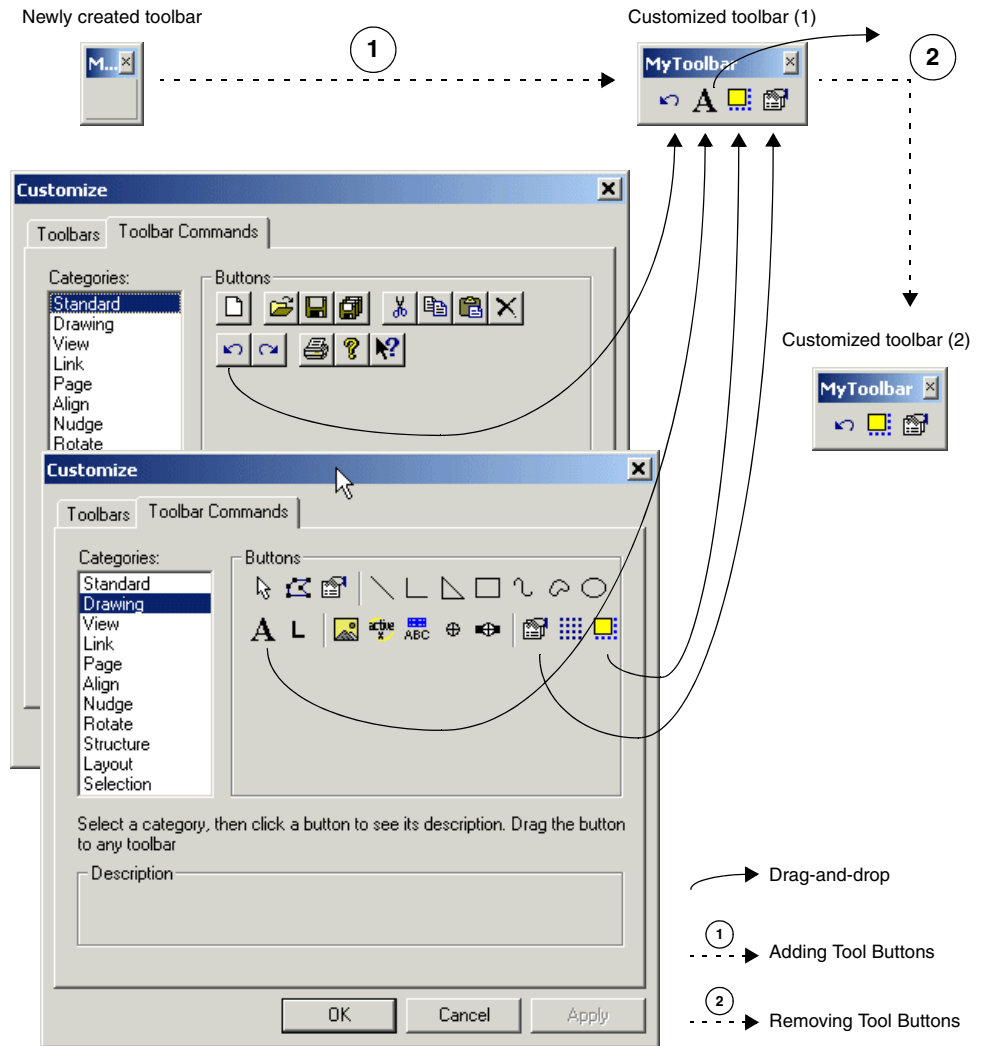


Figure 168. Collating a Toolbar (Adding/Removing Tool Buttons)

Removing Tool Buttons from a Toolbar

To remove tool buttons from a toolbar, do as follows:

1. Open the **Customize** dialog.
2. Point to the tool button within the toolbar from where it shall be removed, drag-and-drop it out of the toolbar
3. Click **OK** to close **Customize** dialog.



Removing tool buttons from a toolbar is only possible if the **Customize** dialog is open.

Resetting the Composition of a Toolbar

To reset a system (not user-defined) toolbar to its original composition, do as follows:

1. Open the **Customize** dialog.
2. Click on the **Toolbar Commands** tab.
3. From the **Categories** list box, select the (system) toolbar to be reset.
4. Click **Reset**.
5. Click **OK**.



Resetting is only possible for system toolbars not for user-defined toolbars.

Renaming a User-Defined Toolbar

To rename a user-defined toolbar, do as follows:

1. Open the **Customize** dialog.
2. Select the wanted toolbar from the **Toolbars** list box on the **Toolbars** page.
3. Rename the toolbar in the **Toolbar** name field.
4. Click **OK**.

Deleting a User-Defined Toolbar

To delete a user-defined toolbar, do as follows:

1. Open the **Customize** dialog.
2. Select the wanted toolbar from the **Toolbars** list box on the **Toolbars** page.
3. Click **Delete**.
4. Click **OK**.

Changing the Look of Toolbars

You can display the toolbar buttons in a **Cool Look** manner. Then, the buttons looks like as follows:

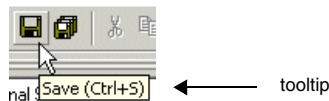


To change the look of the toolbars, do as follows:

1. Open the **Customize** dialog.
2. Depending on the wanted look, mark or un-mark the **Cool Look** check box:
 - To turn on the **Cool Look** display, mark the check box.
 - To turn off the **Cool Look** display, un-mark the check box.
3. Click **OK**.

Showing Tooltips

Tooltips are short context-sensitive descriptions of the toolbar buttons. A tooltip is displayed on pointing to the appropriate toolbar button.



To turn on or off the display of the tooltips, do as follows:

- 1. Open the **Customize** dialog.
- 2. Mark or un-mark the **Show Tooltips** check box:
 - To turn on showing tooltips, mark the check box.
 - To turn off showing tooltips, un-mark the check box.
- 3. Click **OK**.

General Accelerator Keys

The following table lists the general accelerator keys for having quick access to certain functions.

Table 44. General Accelerator Keys Description

Key	Context	Description
Cursor key (up, down, right left)	on symbols	Symbol traversal: Selects prev/next symbol according to graphical order
	on ports	Network traversal: Selects prev/next port or connection link according to graphical order
	on connection links	Network traversal: Selects prev/next port or connection link according to graphical order
SHIFT + left click	on symbols	Contiguous multiple selection
	on ports	Contiguous multiple selection
SHIFT + cursor key (up, down, right, left)	on symbols	Contiguous multiple selection
	on ports	Contiguous multiple selection
SHIFT + left drag	on symbols	Moves selected symbols only in x/y direction
	on symbol selection handle	Sizes selected symbols and keep aspect ratio
	on empty area (selection rectangle)	Selects (only) symbols in the selection rectangle

Table 44. General Accelerator Keys Description (Continued)

Key	Context	Description
CTRL + left click	on symbols	Disjoint multi selection
	on ports	Disjoint multi selection
	on connection links	Multi selection
	on link vertex	Removes link vertex
CTRL + cursor	on symbols	Disjoint multi selection
	on ports	Disjoint multi selection
	on connection links	Disjoint multi selection
CTRL + left drag	on symbols	Copies selected symbols
	on symbol selection handle	Sizes selected symbols and keep center
	on empty area (selection rectangle)	Selects only connections in the selection rectangle
Shift + Ctrl + left click	on link vertex	freeze / unfreeze link vertex
Shift + Ctrl + left drag	on function component(s)	Insert function component(s) from/into another structure
ALT + left click	on port	Connects marked port to this port
	on selected connection link	Inserts link vertex
ALT + left drag	on symbols	Moves selected symbols, ignore grid setting
	on symbol selection handle	Sizes selected symbols, ignore grid setting
	on empty area (selection rectangle)	Selects only ports in the selection rectangle
Space + cursor	on selected objects	Moves selected objects by cursor keys 1/10 of the module (Nudge)
N + cursor		

Table 44. General Accelerator Keys Description (Continued)

Key	Context	Description
Space + SHIFT + cursor	on selected objects	Moves selected objects by cursor keys 1/2 of the module (Nudge)
N + SHIFT + cursor		
P + left drag	-	Pan diagram by mouse
P + cursor		Pan diagram by cursor keys 1/10 of the module
P + SHIFT + cursor		Pan diagram by cursor keys 1/2 of the module
NUMPAD “5”	on symbols	Toggles cursor key navigation context from symbol traversal to network traversal
	on ports/connection links	Toggles cursor key navigation context from network traversal to symbol traversal
NUMPAD “-”	on ports	Marks this port for connection
ALT + NUMPAD “-”	on ports	Connects this port to marked port by keyboard
DEL	on symbols	Deletes selected symbol(s)
	on ports	Deletes selected port(s)
	on connection links	Deletes selected connection(s) (disconnect)
F2 (Not in Topology Designer)	MMS editor	Rename sender/receiver diagrams and access variables Change cycle time
CTRL+F5 (Not in Topology Designer)	MMS editor	Print the tree with MMS configuration data.
CTRL+F6 (Not in Topology Designer)	MMS editor	Copy the tree contents with MMS configuration data onto the clipboard.
CTRL+right arrow (Not in Topology Designer)	MMS editor	Expand the selected tree node.

Open Aspect

Use the **Open Aspect** dialog to navigate to and open an aspect of any Aspect Object. Open the dialog by choosing designer's **File > Open** menu command:

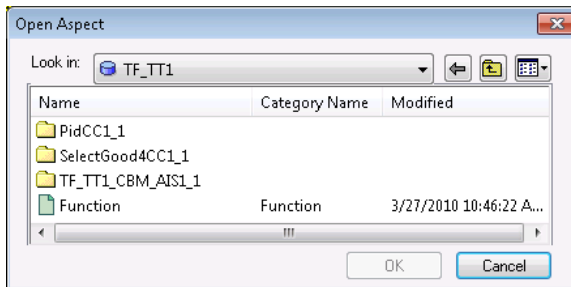


Figure 169. Open Aspect Dialog

The **Open Aspect** dialog is similar to the common File Open dialog. Instead of selecting a file, you select an aspect to open.

You do this by browsing through structures and Aspect Object hierarchies as you do in the common File Open dialog when browsing through directories. Aspect Objects display a folder icon, and Aspects display their aspect icon.

- On top level, the **Look in** drop-down box offers all 800xA system structures.
- User selects a structure to display all Aspect Objects on root level of the selected structure
- User selects an Aspect Object to display all its aspects and child Aspect Objects.

Inserting/Creating Components

There are different workflows while inserting/creating components:

- In Component view,
 - you insert/create graphic components (e.g. primitives) to define a designer component symbol respectively the type of it.

- In Diagram view,
 - you create new designer component Aspect Objects in designer structure. Typically, you drag & drop an object type with a designer aspect from Object Type Structure into the logic layer of a designer diagram.
 - you insert an existing designer component Aspect Object from another structure. Typically, you Shift+Ctrl drag & drop the Aspect Object from e.g. Control Structure into a designer diagram.
 - you insert/create graphic components, e.g. primitives, on the master page, logic, background, or annotation layer.
- From Diagram view,
 - you insert an existing designer component Aspect Object into another structure. Typically, you Shift+Ctrl drag & drop the Aspect Object from the designer diagram e.g. into Control Structure.
 - you can even insert designer component Aspect Objects into Bulk Data Manager. Typically, you drag & drop the Aspect Objects from the function diagram into a MS Excel sheet with activated Bulk Data Manager.

General Procedure for Inserting/Creating a Graphic Component in Component/Diagram View

To insert/create a graphic component, do as follows:

1. Select a graphic component to be inserted in a diagram by toolbar or menu command.
2. Left-click on the diagram where you want to insert the graphic component. This point will be the graphic components's upper left corner.
3. Move the mouse to define the graphic components's size/length/direction.
4. Release the mouse button.

The graphic component(s) is (are) sized according to the modifier key(s). In general,

 - SHIFT locks the aspect ratio (sizes equally in both x and y direction)
 - CTRL draws component centered on point where you started

- ALT ignores grid setting.

Table 45. Modifier Keys for inserting/creating graphic components

Key	Description
None	Size
SHIFT	Size and keep aspect ratio
CTRL	Size and center to starting point
SHIFT+CTRL	Size, keep aspect ratio, and center to starting point
ALT	Size and ignore grid setting
ALT+SHIFT	Size, keep aspect ratio, and ignore grid setting
ALT+CTRL	Size, center to starting point, and ignore grid setting
ALT+SHIFT+CTRL	Size, keep aspect ratio, center to starting point, and ignore grid setting

User can insert the following graphic components:

- [Inserting Graphic Primitives](#) (Component View and Diagram View)
- [Inserting Text Components](#) (Component View and Diagram View)
- [Inserting Label Components](#) (Component View and Diagram View)
- [Inserting Ports](#) (Component View only)
- [Inserting Pictures/Images](#) (Component View and Diagram View)
- [Inserting ActiveX Controls](#) (Component View and Diagram View)
- [Inserting Graphic Elements build with Graphics Builder](#) (Component View and Diagram View)

General Procedure for Inserting a Designer Component in Diagram View

To create a new designer component Aspect Object, do as follows:

1. Select an object type with designer aspect from Object Type Structure.
2. Drag & Drop it into the designer diagram.

- 3. A new Aspect Object gets created in designer structure.
- 4. Its symbol gets displayed on the designer diagram.

To insert an existing designer component Aspect Object instance into a designer diagram, do as follows:

- 1. Select an Aspect Object with Designer aspect from any structure, except Object Type Structure and Designer structure (because insert of one and the same designer component in two designer diagrams is not supported).
- 2. Hold down the Shift+Ctrl keys while dragging and dropping it into the designer diagram. These are the same modifier keys as used in Plant Explorer to insert Aspect Objects.
- 3. The Aspect Object gets inserted into designer structure.
- 4. Its symbol gets displayed on the designer diagram.

To insert an existing designer component Aspect Object from the designer diagram into another structure, e.g. Control Structure, do as follows:

- 1. Select a designer component on the designer diagram.
- 2. Hold down the Shift+Ctrl keys while dragging and dropping it from the designer diagram into another structure. These are the same modifier keys as used in Plant Explorer to insert Aspect Objects.
- 3. The Aspect Object gets inserted into the other structure.

Table 46. Modifier Keys for Inserting/Creating Designer Components

Key	Description
None	Create designer component from Object Type Structure. Move designer component into other structure, or into other diagram
CTRL	Copy designer component
SHIFT+CTRL	Insert designer component into/from another structure



It can happen unintentionally that a component gets removed from the diagram and from designer structure by just moving the component outside the diagram frame. In effect, the component gets moved from the diagram to some structure that is selected in the structure browser. An according warning is given in this case

Inserting Graphic Primitives

If you insert graphic primitives in Component view or in the Diagram view, the cursor appearance will give you information about the allowed drop location. In that case, the cursor will change its appearance from an arrow to a plus sign. After insertion has finished, the cursor's appearance is reset to an arrow. The inserted graphic primitive is displayed with its vertices.

Inserting a Line

To insert a line, do as follows:

1. Choose the menu command **Insert > Graphic > Lines** or click the appropriate toolbar button.
2. Move the cursor to the source point on the diagram.
3. Press the mouse button on the source point of the line, drag it to the target point and then release it. The line is displayed with its vertices.

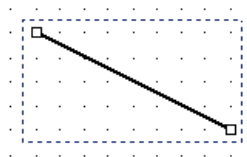


Figure 170. Line

Inserting a graphic primitive also includes the following options:

- Inserting a Polyline
- Inserting a Polygon

- Inserting a Rectangle
- Inserting a Polycurve
- Inserting a Closed Curve
- Inserting an Ellipse

Inserting Text Components

To insert a text component, do as follows:

1. Choose the menu command **Insert > Text** or click the appropriate toolbar button.
2. Click on the place in the diagram where the text component shall be inserted.
3. In-cell edit the text string inside the text component.
4. Modify the text component properties to define fonts, alignment, multiple lines with word wrapping etc.



Inserting Label Components

Labels are special text components which are associated with an owner component, a symbol or a (connection) link. Labels get positioned in relation to their owner.


When you move a symbol or a link, its label(s) are also moved. However, labels can be selected, moved, and modified independently of their owner.

The label's orientation property tells the label to position itself at one of nine orientation points around the owner symbol or owner link. You can also define how much distance there should be between a label and its owner.

To label a symbol

1. Choose menu command **Insert > Label** or click  in the **Drawing** toolbar. The cursor changes to .
2. In Diagram view, click inside a symbol where you want to associate and position the label. A label component gets created.
In Component view, click on any place in the symbol definition where the label shall be inserted. A label component gets created.

To label a link (connection line)

1. Choose menu command **Insert > Label** or click **L** in the **Drawing** toolbar. The cursor changes to .
2. Click on the link where you want to associate and position the label. A label component with default name gets created.
3. To modify the default link name,
 - double-click the label and modify the name directly, or
 - right click on the link to open its context menu, choose **Component Properties** and modify the name in the **Name** field on the **General** page of the **Component Properties** dialog.

The link will now have the name of the label as displayed in the following figure:

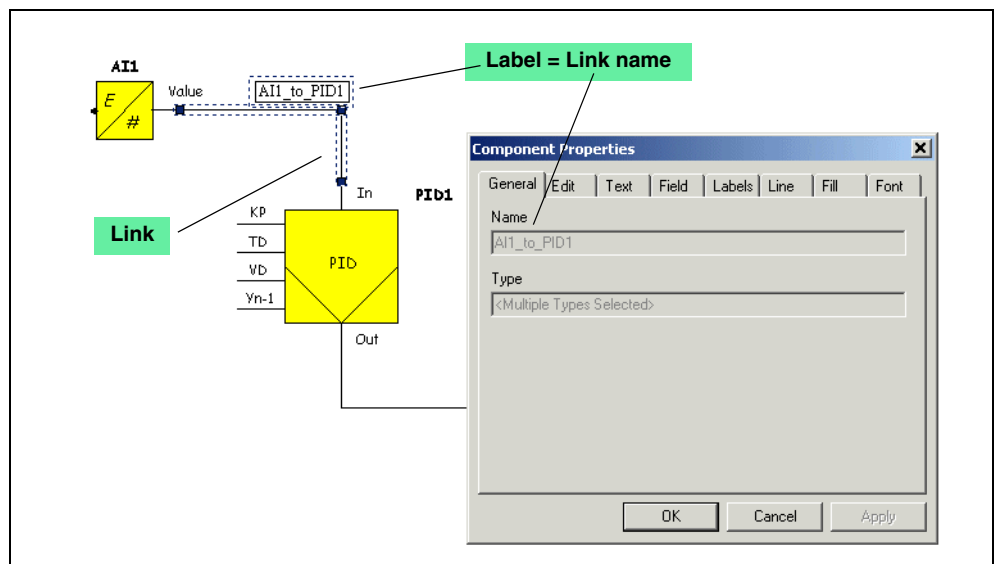


Figure 171. Labeled Link



While inserting or editing the name, label, or text in a Function Diagram, text box with blue background is displayed, to overcome this problem, press Enter or click outside the text box.

Inserting Ports


A port defines a location on a symbol at which other symbol ports can be connected to. A port always belongs to exactly one symbol, and can be visible or invisible. Ports can be used to create connections between any two symbols.

Ports can only be defined in Component view (see [Diagram/Component View](#)).

1. Open the designer's Component View.
2. Click the appropriate toolbar button from the [Drawing](#) toolbar or choose the corresponding menu command [Insert > Circle Port](#) or [Insert > Line Port](#).
3. A circle port consists of just a circle 'hot spot' to connect to. To insert a circle port, simply click on the place in the symbol definition, where you want to insert the port.
4. A line port consists of a line and a circle 'hot spot' to connect to. To insert a line port, press and hold the mouse button on the place in the symbol definition where you want to start the line port, drag the mouse and release the button where the line port's circle 'hot spot' shall end.
Thus to insert an 'input' line port at the left side of a symbol definition, drag the mouse from right to left, and to insert an 'output' port at the right side of a symbol definition, drag the mouse from left to right.

Inserting Pictures/Images

To insert a picture/image, do as follows:

1. Click the Image button from the [Drawing](#) toolbar or choose the [Insert > Picture](#) menu command.
2. From the displayed standard File Open dialog, select the wanted picture/image in either .bmp, .wmf or .dib format.
3. Click **Open**. The dialog is closed and the mouse pointer changes to .
4. Click on the place in the diagram where you want to insert the image.

Open

Use this standard dialog to open a file such as a picture/image for inserting it into a symbol definition or into a diagram.

Access this dialog via the menu commands **Insert > Picture** or via the appropriate toolbar button.

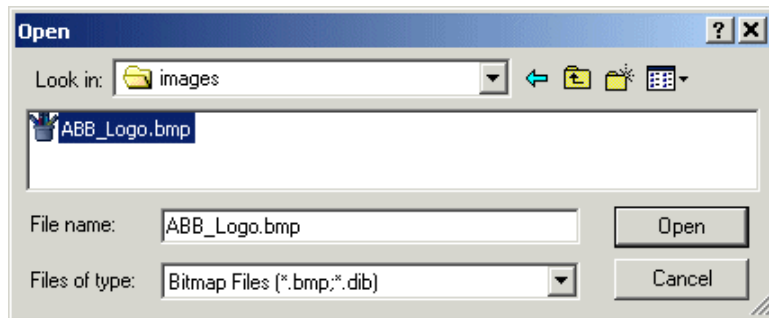


Figure 172. Open Dialog

- **Open**
Click **Open** to confirm the image selection for inserting the file (for example, an image file) into the active diagram.
- **Cancel**
Click **Cancel** to quit the dialog without inserting an image.

Inserting ActiveX Controls

To insert an ActiveX Control, do as follows:

1. Click the ActiveX button from the **Drawing** toolbar or choose the **Insert > ActiveX** menu command. The **Insert ActiveX Control** dialog appears:

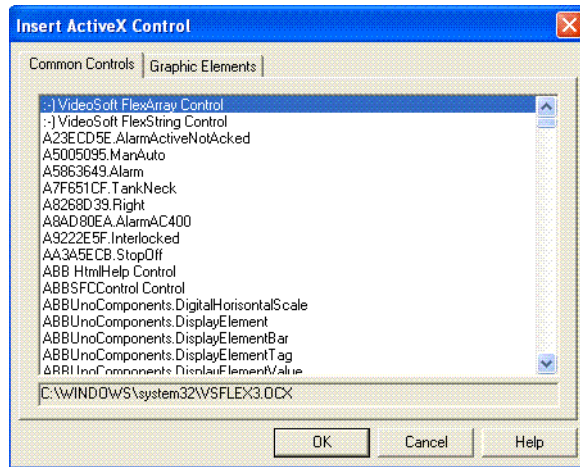



Figure 173. Insert ActiveX Control

2. Select the wanted ActiveX Control from the list.
3. Click **OK**. The dialog is closed and the mouse pointer changes to .
4. Click on the place in the diagram where you want to insert the ActiveX Control.

Inserting Graphic Elements build with Graphics Builder

To insert a graphic element built with *Graphics Builder*, do as follows:

1. Click the ActiveX button from the **Drawing** toolbar or choose the **Insert > ActiveX** menu command. The **Insert ActiveX Control** dialog appears.
2. Select the page **Graphic Elements**:

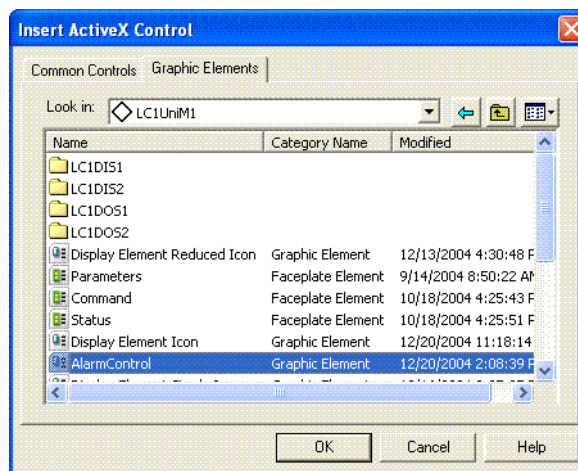



Figure 174. Insert Graphic Elements

3. Select the wanted graphic element from the list.
4. Click **OK**. The dialog is closed and the mouse pointer changes to .
5. Click on the place in the diagram where you want to insert the graphic element.

Inserting Designer Components

Designer components can be inserted into a designer diagram by creating new instances from an object type in Object Type Structure, or by inserting existing Aspect Objects from other structures (for example, from Control Structure).

Inserting Designer Components from Object Type Structure

To insert new designer components from Object Type Structure into a designer diagram, choose one of the following methods:

- [Inserting via Drag & Drop](#)
- [Inserting via Menu Command](#)

Inserting via Drag & Drop

1. Select Object Type Structure from any structure browser of designer or from the object browser of Plant Explorer.
2. Choose the wanted object type from Object Type Structure.
3. Drag the object type to the wanted location in the diagram and drop it.
4. For *Function Designer* only: An optional **New Component Name** dialog gets displayed in order to accept or change the proposed default name and to define the data flow order of the new function component, see [Figure 249](#).

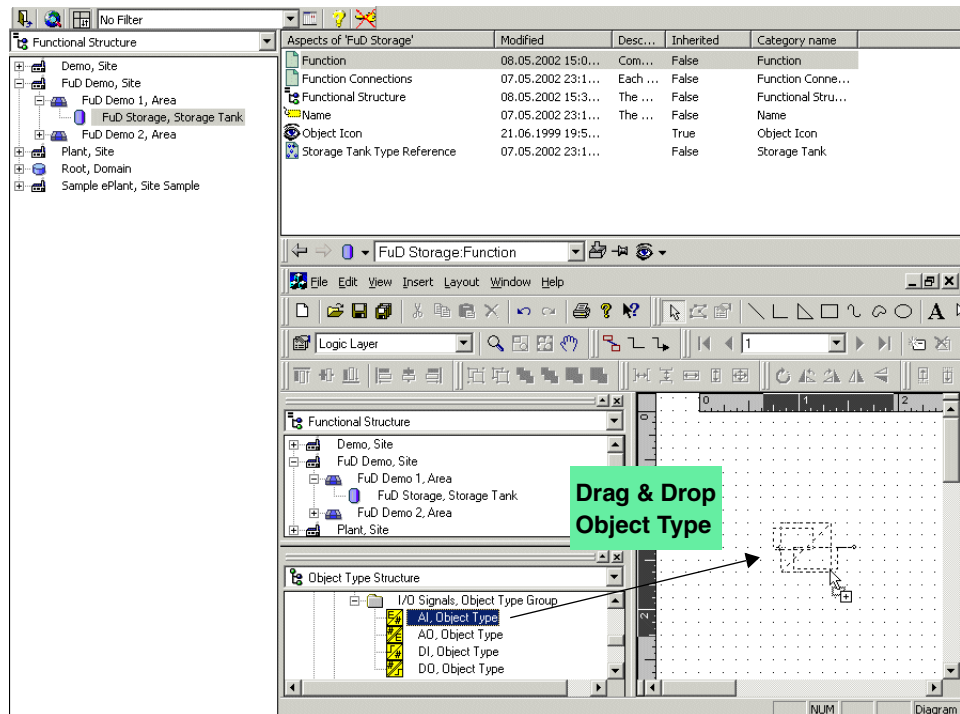


Figure 175. Drag & Drop from Object Type Structure to Function Diagram

After having dropped the object type in the diagram, a new designer component object is created (instantiated) in the designer diagram. According to the object

type's Create Info definition, an Aspect Object is created (instantiated) as child of the diagram object in designer structure.

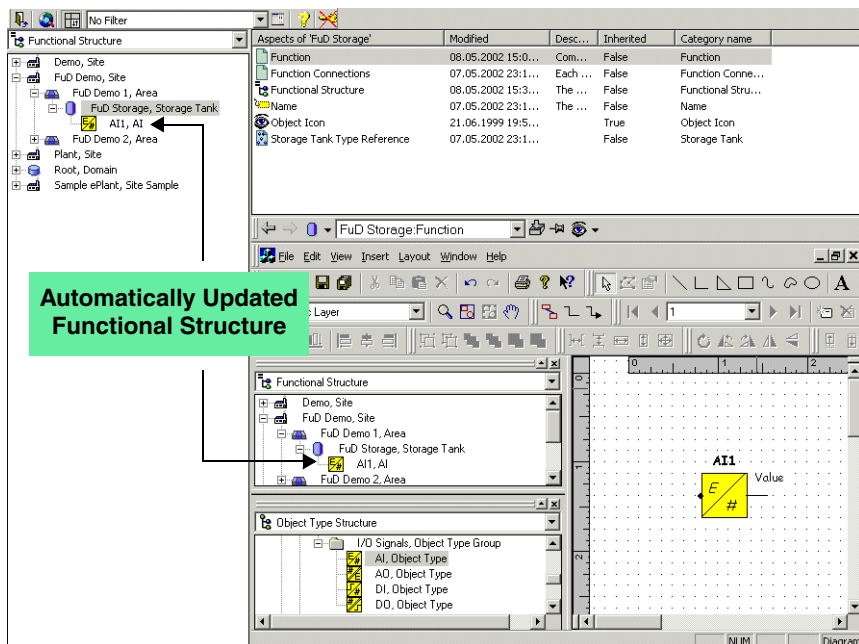


Figure 176. Automatically Updated Functional Structure

Inserting via Menu Command

1. Choose menu command **Insert > Object**. The **Insert Objects** dialog appears.
2. On the **Object Type** page, select the wanted structure (for example, designer structure, Object Type Structure) from the drop-down list.
3. Select the wanted object type with a designer aspect from the list.
4. Note: Not available for Topology Designer.

An optional **New Component Name** dialog gets displayed in order to accept or change the proposed default name and to define the data flow order of the new designer component, see [Figure 249](#).

5. A new designer component object is created (instantiated) and automatically placed in the designer diagram at the next free location. If necessary, a new page is created. According to the object type's create info definition, a new Aspect Object is created (instantiated) as child of the diagram object in designer structure.
6. If you want to apply several designer components, click **Apply** and continue with step 3.
If you want to apply only one designer component, click **OK**.

Insert Objects

Use this dialog to insert designer components by selecting them from the appropriate structure browser on the dialog's **Object Types** page.

Open this dialog via the menu command **Insert > Object**.

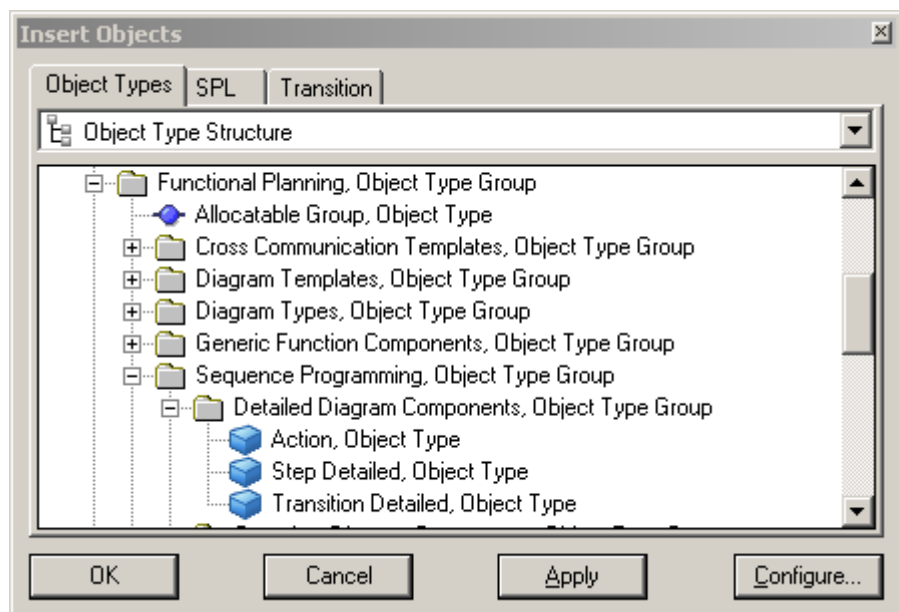


Figure 177. Insert Objects Dialog

- **Object Types**

Click this tab to navigate through various Structures and select specific Object Types.

- **SPL**

Click this tab to get list of objects that can be used in SPL diagrams.

- **Transition**

Click this tab to get list of Object types like Arithmetic expressions, Comparison expressions, and Boolean type logical expressions.

- **OK**

Click **OK** to apply the selected object to the active diagram and close the dialog.

- **Cancel**

Click **Cancel** to quit the dialog without applying the selected object.

- **Apply**

Click **Apply** to apply (auto-insert) the selected object to the active diagram and leave the dialog open.

- **Help**

Click **Help** to get dialog-specific help.

Configure Favorite Pages

Function Designer supports to configure your own favorite pages in the Insert Objects dialog to reduce navigation time during insertion of objects into function diagrams.

To configure favorite pages, do as follows:

1. Open Insert Object dialog via the menu command **Insert > Symbol**.
2. Press **Configure** to open the Configure Favorite Object Types dialog.
3. Press Add Page.
4. In the Favorites Page Name edit box enter a name for the page.

5. Press OK.
6. From the Object Type Structure in the object browser on the left side of Configure Favorite Object Types dialog drag and drop your favorite object types / groups of object types / libraries into the favorite page on the right side of the dialog. The favorite page shows the favorite object types in a flat list.
7. Press OK.

A new tab with the favorite page name is created in the **Insert Object** dialog.



Function designer supports to configure more than one favorite pages.

Inserting Designer Components from/into Other Structures

Inserting into other structures

It is often useful to place designer component Aspect Objects also in other structures than in designer structure.

For example: Place function component I/O signal objects of Functional Structure into Control Structure to allocate them.

To insert Aspect Objects from a diagram into another structure, use drag-and-drop with SHIFT+CTRL.

Inserting from other structures

To insert existing Aspect Objects from other structures into designer structure, use drag-and-drop with SHIFT+CTRL.

For example: Place already allocated I/O signals from Control Structure into a function diagram.

Viewing Components

If user has complex designer component definitions or complex diagrams, it is very important to have a possibility to find and display that particular part of the diagram. Designer offers comfortable **Zoom** and **Pan** functionality. Using them, user can always have an optimal view and good overview of the diagrams.

Zoom

Zooming means to scale up or down the display of an arbitrary area in Function Designer's Diagram/Component. Zooming is supported by several menu and toolbar commands described in the following.

Zoom Normal

Choose the **View > Zoom Normal** menu command to zoom to set (reset) the default size of components (100%). All other zoom values refer to that size.

Zoom Percent

Designer offers some predefined values for zooming accessible via menu commands or from a context menu:

Choose one of the following submenu options of **Zoom Percent** to zoom to the required size:

- **View > Zoom Percent > 50%**
- **View > Zoom Percent > 75%**
- **View > Zoom Percent > 100% (Normal)**
- **View > Zoom Percent > 200%**

Zoom Percent via context menu

Right-click an empty area (not on a component) and choose one of the following commands from the context menu:

- **Zoom > 50%**
- **Zoom > 75%**
- **Zoom > 100%**
- **Zoom > 200%**

Zoom Custom

You can adjust an arbitrary value for zooming the display of components. For it, open the **Zoom** dialog by choosing the **View > Zoom Custom** menu command and

enter the wanted value in the **Magnification %** field or choose one of the predefined values from the drop-down list box.

Zoom to Fit

Zoom to fit means to display all available components of the designer aspect's Diagram view/Component view. You get this overview by doing one of the following alternatives:

- Choose the menu command **View > Zoom to Fit**
- Choose **Zoom > Zoom to Fit** from the context menu
- Choose the **Zoom to Fit** toolbar command.

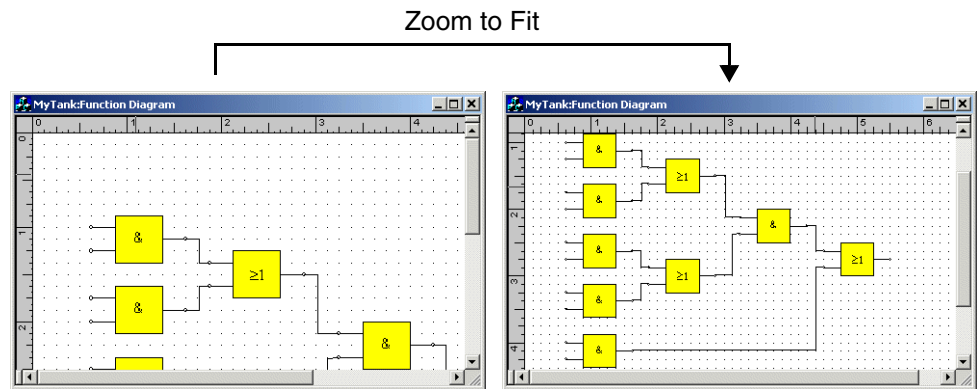


Figure 178. View: Zoom to Fit

Zoom to Selection

Zoom to selection means to display all selected components maximum sized in the current window. Choose this command to display the wanted selection in the optimal (maximum) size.

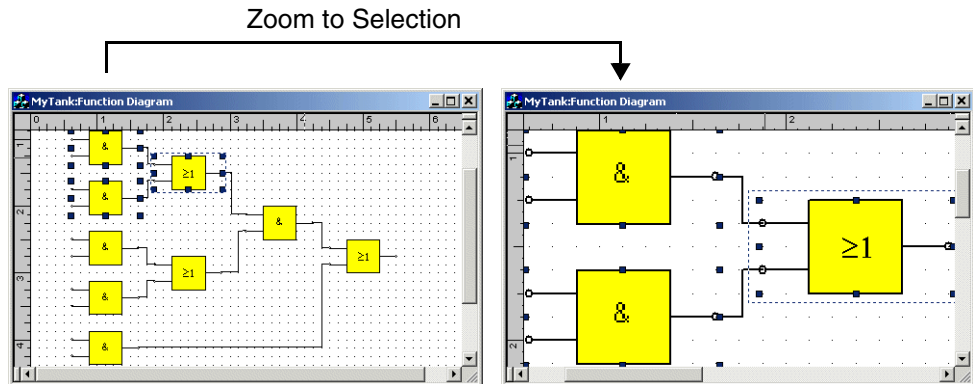




Figure 179. View: Zoom to Selection

Zoom in steps via Mouse

It is possible to zoom via mouse. Do this as follows:



1. Click the Zoom button  from the View toolbar. The cursor in the Diagram/Component view changes to .
2. Zoom by mouse-clicks as follows:
 - To zoom increasing, click left onto the area you want to have centered after zooming.
 - To zoom decreasing, click right onto the area you want to have centered after zooming.

Pan

Using the Pan functionality means to move the whole content of the diagram/component document within the corresponding window.

It can be easier to use the Pan functionality than to configure the wanted area via the horizontal and vertical scroll bars.

To use the Pan functionality, do as follows:

1. Click the Pan button  from the View toolbar. The cursor in the Diagram/Component view changes to .
2. Move the cursor to the center of the area you want to move.
3. Press the left mouse button and move the diagram/component document to the wanted location.
4. Release the mouse button.

Selecting Components

You can select or deselect one or multiple components by mouse or by keyboard in the drawing area of the active layer in the active view. The selection can include components on one page or on multiple pages. Selected components are highlighted and define the context for commands like move, delete, cut, copy, and so on.

The components displayed in the drawing area are ordered by the top-left coordinates of their surrounding rectangle in the sense of top to bottom and left to right. Assuming the coordinates $(x, y)=(0, 0)$ at the top-left corner and x growing to the right and y growing to the bottom, a component precedes another one

- if it is located above (lower y value)
- if it is located to the left (lower x value)

This order is called graphical order. Independent from that, there might exist some logical order, for example, the insertion or data flow order of symbols.



Selection always follows the graphical order, not the logical order.

Multiple Selection

Multiple selection (multi-selection) includes only components of the same type. Following type-specific selections are possible:

- [Selecting Symbols](#)

- [Selecting Ports](#)
- [Selecting Connections](#)

Symbols, ports and connections are selected according to the modifier key(s). In general,

- None (without pressing any key) selects all symbols, ports and connections lying completely inside the selection rectangle
- SHIFT filters and selects symbols only
- CTRL filters and selects connections only
- ALT filters and selects ports only

Table 47. Modifier Keys for Selecting by Dragging a Selection Rectangle

Key	Description
None	Select all objects inside selection rectangle
SHIFT	Select symbols only
CTRL	Select connections only
SHIFT+CTRL	Select symbols and connections only
ALT	Select ports only
ALT+SHIFT	Select ports and symbols only
ALT+CTRL	Select ports and connections only
ALT+SHIFT+CTRL	Select all (As None)



When switching between different layers or splitter views (windows), the selection data is kept per layer and view.

Anchor and Focus

Selected components are highlighted by their selection handles. One of the selected components will be the anchor, one will have the (input) focus. The component having the focus is indicated by a focus rectangle as shown in the following figure:

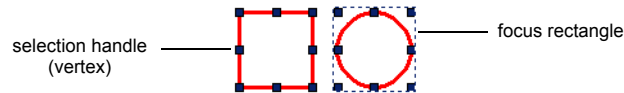


Figure 180. Selection Handles and Focus Rectangle


- On multiple selection via mouse,
 - the anchor will be the upper left selected component and
 - the lower right component will have the focus.
- On multiple selection via keyboard,
 - the anchor will be the first selected component and
 - the last selected component will have the focus.

Contiguous Multiple Selection

For selecting several contiguous components, you have the following methods:


- **Contiguous rectangular multiple selection:**
This kind of multiple selection is rectangle oriented. The selected area on a diagram is determined by a rectangle given from the coordinates of the selection start point (anchor) and the selection endpoint (focus, diagonal of the rectangle).
- **Contiguous linear multiple selection:**
This kind of multiple selection is row oriented (as selecting text in MS Word or any other word processing system). The selected area on a diagram is determined by the graphical order of the components. All objects between (according to graphical order) the anchor and the component having the focus will be selected.



The behavior for contiguous multiple selection can be set using  of the toolbar:

- Button pressed means contiguous multiple linear selection
- Button not pressed means contiguous multiple rectangle selection.



The behavior for network/symbol traversal with cursor keys can be set via  of the toolbar or by selecting a port/symbol:

- Button pressed means network traversal (ports and connection links)
- Button not pressed means symbol traversal.

Disjoint selection

Using mouse and keyboard modifiers, it is also possible to disjointedly select components (see also [General Accelerator Keys](#)).

Selecting Symbols

For selecting symbols, you have the following possibilities:

- [Selecting Single Symbols](#)
- [Contiguous Rectangular Multiple Selection of Symbols](#)
- [Contiguous Linear Multiple Selection of Symbols](#)
- [Selecting Symbols via the Components Dialog](#)

Selecting Single Symbols

To select single symbols (not in graphical order) from the designer window, choose one of the following methods:

- To select only one symbol, simply click it.
- To select several symbols, click the first symbols, press and hold the CTRL key, and then click the other wanted symbols.

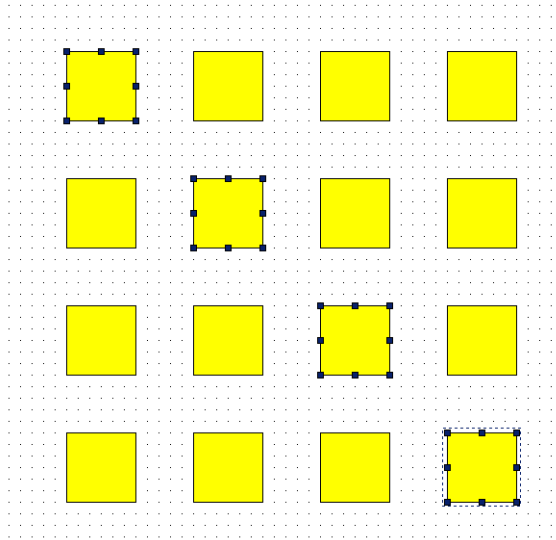


Figure 181. Selection of Single Symbols

Contiguous Rectangular Multiple Selection of Symbols

To select contiguous symbols within a rectangular diagram area, you have two methods:

- [Selecting symbols by defining a rectangle selection using the mouse](#)
- [Selecting symbols by defining a rectangle selection through keyboard.](#)



All relevant symbols must be located totally in the selection rectangle

Selecting symbols by defining a rectangle selection using the mouse

1. Left-click on some background in the drawing area.
2. Move the mouse and drag a selection rectangle as needed

3. Release the mouse button.

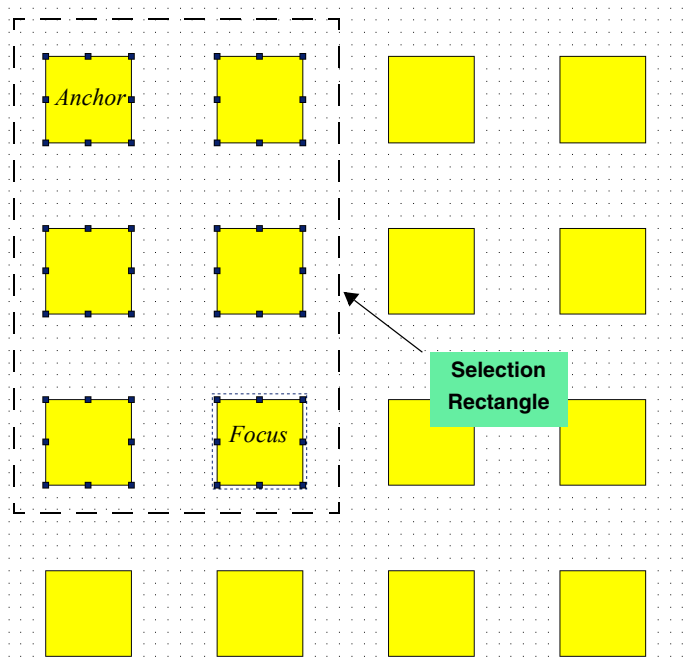




Figure 182. Contiguous Rectangular Multiple Selection

Selecting symbols by defining a rectangle selection through keyboard

1. Select the anchor symbol by clicking on it.
2. Switch to the Contiguous Rectangular Multiple Selection mode by clicking  of the Selection toolbar (button/icon must be released)
3. Press SHIFT and select the symbol defining the other (diagonal) end of the selection rectangle. You can do this via mouse or using the cursor keys. This symbol automatically gets the focus.

Contiguous Linear Multiple Selection of Symbols

To select all symbols, located in the horizontal (diagram) stripe between two symbols, the anchor symbol and the symbols having (getting) the focus, do as follows:

1. Select the anchor symbol by clicking on it.
2. Switch to the Contiguous Linear Multiple Selection mode by clicking  of the Selection toolbar (button/icon must be pressed)
3. Press SHIFT and select the symbols defining the other end of the selection stripe. This symbol automatically gets the focus.

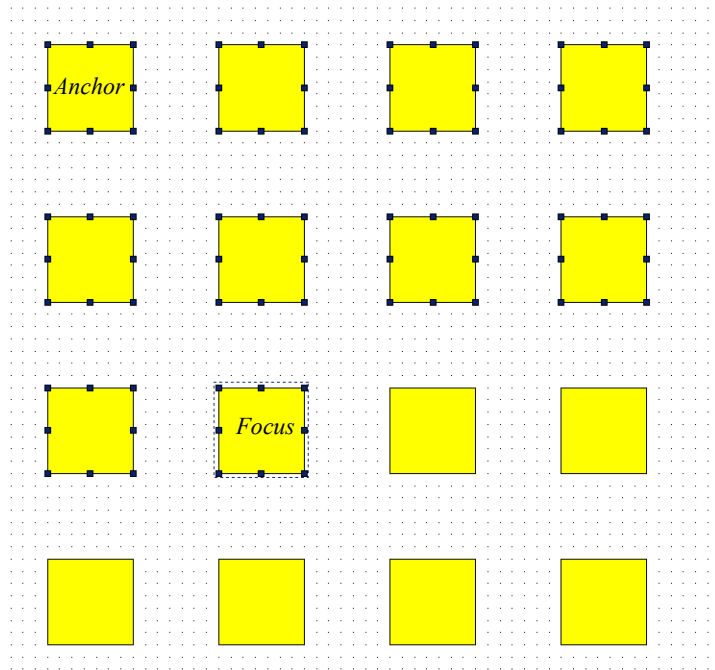


Figure 183. Contiguous Linear Multiple Selection of Symbols

Selecting Symbols via the Components Dialog

To select symbols from the [Components](#) dialog's list box, do as follows:

- To select only one symbol (component) from the list, simply click it.
- To select several series listed symbols, click the first symbol of the series, then press and hold the Shift key, and then click the last symbol of the series. All symbols of the series will be highlighted.
- To select several symbols not listed in series. Press and hold the CTRL key, then click the wanted symbols.



This kind of selection doesn't mark the selected symbols with vertices inside the diagram.

Selecting Ports

For selecting ports, you have the following possibilities:

- [Selecting Single Ports](#)
- [Contiguous rectangular multiple selection of Ports](#)
- [Contiguous linear multiple selection of Ports](#)

Selecting Single Ports

To select single ports (not in graphical order) from the designer window, choose one of the following methods:

- To select only one port, simply click it.

- To select several ports, click the first port, then press and hold the CTRL key, and click the other wanted ports.

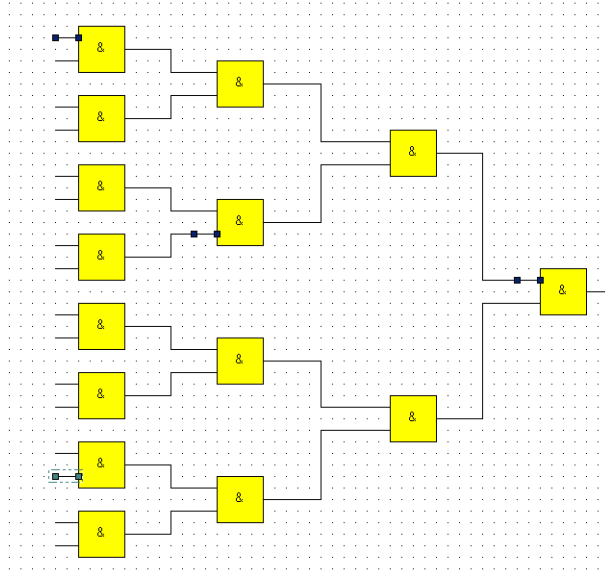


Figure 184. Selection of Single Ports

Contiguous rectangular multiple selection of Ports

To select the contiguous ports within a rectangular diagram area, you have two methods:

- [Selecting symbols by defining a rectangle selection using the mouse](#)
- [Selecting symbols by defining a rectangle selection through keyboard](#)



All relevant ports must be located totally in the selection rectangle.

Selecting ports by defining a rectangle selection through mouse

1. Left-click on some background in the drawing area.

2. Press ALT, move the mouse and drag a selection rectangle as needed
3. Release the mouse button and ALT key.

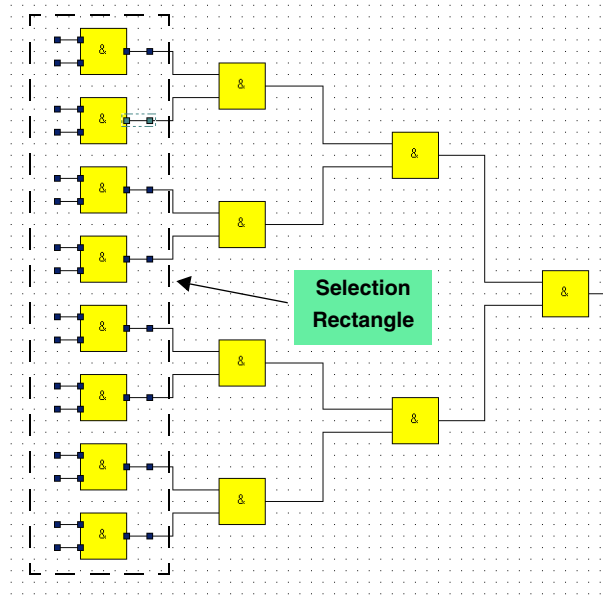



Figure 185. Contiguous Rectangular Multiple Selection of Ports via Mouse

Selecting ports by defining a rectangle selection through keyboard

1. Select the anchor port by clicking on it.
2. Switch to the Contiguous Rectangular Multiple Selection mode by clicking  of the Selection toolbar (button/icon must be released)

3. Press SHIFT and select the port defining the other (diagonal) end of the selection rectangle. You can do this via mouse or using the cursor keys.

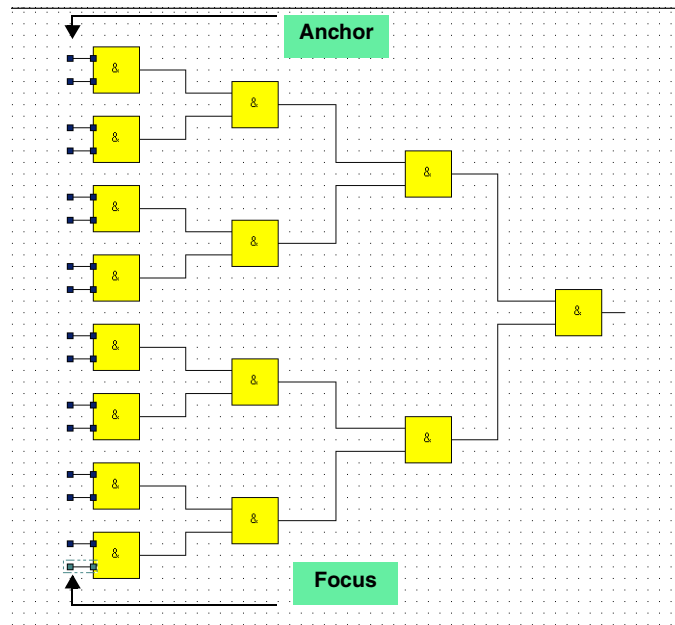



Figure 186. Contiguous Rectangular Multiple Selection of Ports via Keyboard

Contiguous linear multiple selection of Ports

To select all ports, located in the horizontal (diagram) stripe between two ports, the anchor port and the port having (getting) the focus, do as follows:

1. Select the anchor port by clicking on it.
2. Switch to the Contiguous Linear Multiple Selection mode by clicking  of the Selection toolbar (button/icon must be pressed)

3. Press SHIFT and select the port defining the other end of the selection stripe. This port automatically gets the focus.

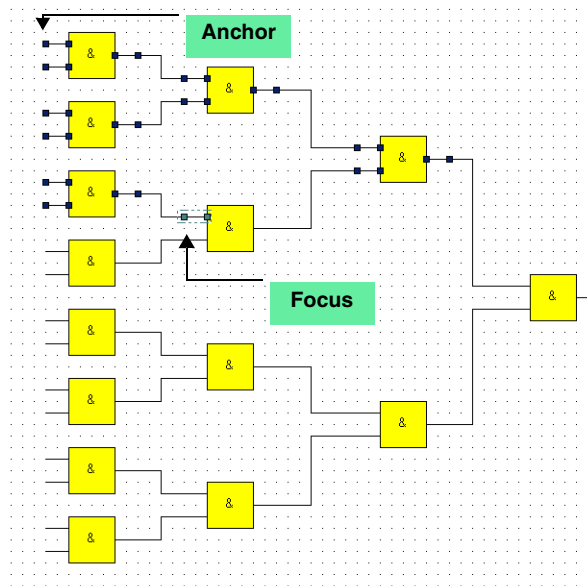


Figure 187. Contiguous Linear Multiple Selection of Ports

Selecting Connections

For selecting connections, you have the following possibilities:

- [Selecting Single Connections](#)
- [Contiguous Rectangular Multiple Selection of Connections](#)
- [Contiguous Linear Multiple Selection of Connections](#)

Selecting Single Connections

To select single connections (not in graphical order) from the designer window, choose one of the following methods:

- To select only one connection, simply click it.

- To select several connections, click the first connection, then press and hold the CTRL key, and click the other wanted connections.

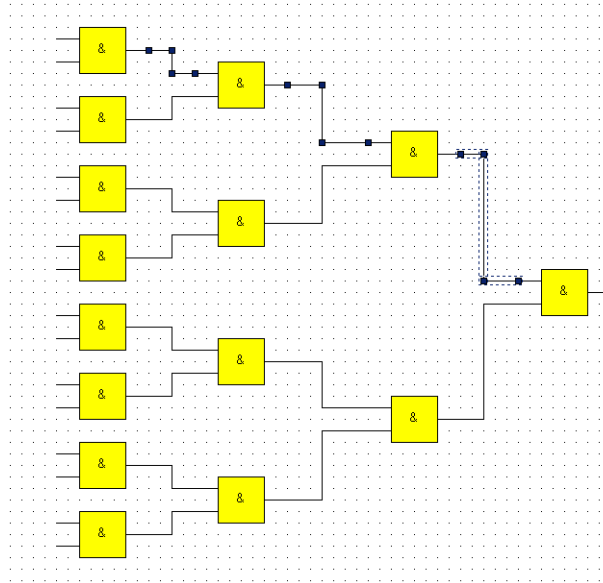


Figure 188. Selection of Single Connections

Contiguous Rectangular Multiple Selection of Connections

To select the contiguous connections within a rectangular diagram area, you have two methods:

- [Selecting symbols by defining a rectangle selection using the mouse](#)
- [Selecting ports by defining a rectangle selection through keyboard](#)



All relevant connections must be located totally in the selection rectangle

Selecting connections by defining a rectangle selection via mouse

1. Press left mouse button on some background in the drawing area.

2. Press CTRL, move the mouse and drag a selection rectangle as needed
3. Release the mouse button and CTRL key.

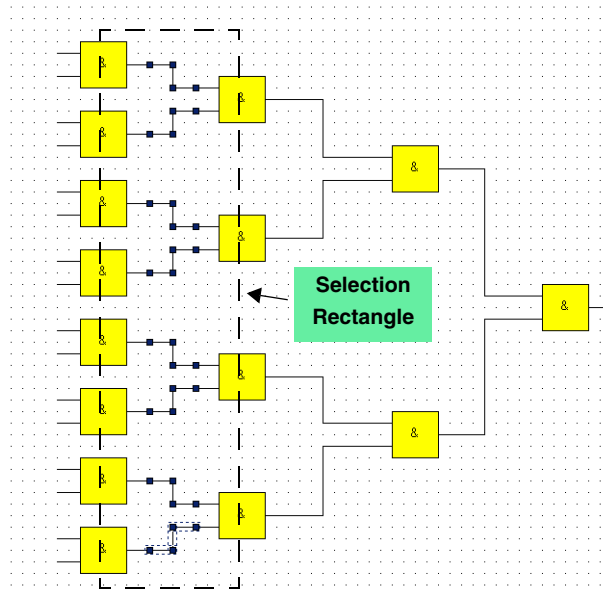




Figure 189. Contiguous Rectangular Multiple Selection of Connections via Mouse

Selecting connections by defining a rectangle selection via keyboard

1. Select the anchor connection by clicking on it.
2. Switch to the Contiguous Rectangular Multiple Selection mode by clicking  of the Selection toolbar (button/icon must be released)
3. Press SHIFT and select the connection defining the other (diagonal) end of the selection rectangle. You can do this via mouse (click on the target connection) or using the cursor keys.

Contiguous Linear Multiple Selection of Connections

To select all connections, located in the horizontal (diagram) stripe between two connections, the anchor connection and the connection having (getting) the focus, do as follows:

1. Select the anchor connection by clicking on it.
2. Switch to the Contiguous Linear Multiple Selection mode by clicking  of the Selection toolbar (button/icon must be pressed)
3. Press SHIFT and select the connection defining the other end of the selection stripe. This connection automatically gets the focus.

Layout of Components

For manipulating the layout of components, designer offers the following methods:

- [Grid](#)
- [Move](#)
- [Size](#)
- [Align](#)
- [Rotate](#)
- [Flip](#)
- [Order](#)
- [Group](#)

Grid

Working with a grid is useful for positioning components on same levels (horizontally or vertically). To enable and adjusting a grid, use the [Grid Properties](#) dialog.

Grid Properties

The Grid Properties dialog allows you to set

Access this dialog through the **View > Grid Properties** menu command.

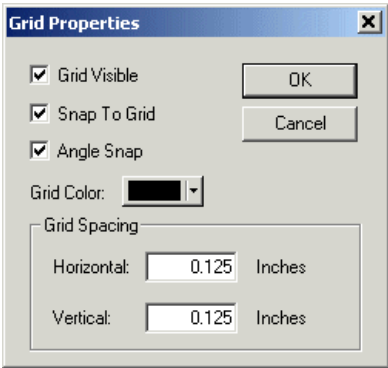


Figure 190. Grid Properties Dialog

Table 48. Description of Grid Properties

Grid Property	Description
Grid Visible	Select this check box to make the grid visible. Otherwise unmark it.
Snap to Grid	Select this check box to snap to the grid on creating, moving, or copying components. Otherwise unmark it.
Angle Snap	Select this check box to snap to the grid on rotating components.
Grid Color	Click on the drop-down list box to open the Color dialog for selecting the grid color.
Grid Spacing	Groups the fields for entering the grid spacing values.
Horizontal	Enter the value for horizontal grid spacing. The unit is adjusted in the Measurements and Size dialog.
Vertical	Enter the value for vertical grid spacing. The unit is adjusted in the Measurements and Size dialog.

Move

For moving components, choose one of the following methods:

- [To move components through drag-and-drop with left-click.](#)
- [To move components through drag-and-drop with right mouse.](#)



If components are moved between different diagrams, the Functional Structure is updated automatically.

See also:

- [To cut and paste components inside designer diagram via context menu.](#)
- [To cut and paste components inside designer diagram via menu command.](#)

To move components through drag-and-drop with left-click

1. Select the required components (see [Selecting Components](#))
2. Left-click on one of the selected components.
3. Move the mouse.
4. Release the mouse button.

The selected component(s) are moved according to the modifier key(s):

- SHIFT moves only in x/y direction relative to center
- ALT ignores grid setting.

Table 49. Modifier Keys for Moving Components via Left Mouse

Key	Description
None	Move
SHIFT	Move only in x/y direction relative to center
ALT	Move, ignore grid setting
ALT+SHIFT	Move only in x/y direction relative to center, ignore grid setting

To move components through drag-and-drop with right mouse

1. Select the required components (see [Selecting Components](#))

- 2. Right-click on one of the selected components.
- 3. Move the mouse to drag the selected component(s).
- 4. Release the mouse button. A context menu is displayed.
- 5. Choose **Move Here** from the context menu.

Table 50. Modifier Keys for Right Mouse Handling

Key	Description
None	Displays context menu: Move Here Copy Here Cancel
SHIFT	Moves only in x/y direction relative to center. Displays context menu: Move Here Copy Here Cancel
CTRL	As None
SHIFT+CTRL	As CTRL, ignore SHIFT
ALT	Ignores grid setting. Display context menu: Move Here Copy Here Cancel.
ALT+SHIFT	Ignore grid setting and move only in x/y direction relative to center. Display context menu: Move Here Copy Here Cancel.
ALT+CTRL	As None
ALT+SHIFT+CTRL	As ALT+CTRL, ignore SHIFT

Size

To change the size of components, do as follows:

- 1. Left-click on a selection handle (vertex) of selected component(s)
- 2. Move the mouse.
- 3. Release the mouse button.
The component(s) is (are) sized according to the used modifier key(s):
 - SHIFT locks the aspect ratio (sizes equally in both x and y direction)
 - CTRL sizes keeping the component(s) center
 - ALT ignores grid setting

Table 51. Modifier Keys for Sizing Components

Key	Description
None	Size
SHIFT	Size and keep aspect ratio
CTRL	Size and keep center
SHIFT+CTRL	Size and keep aspect ratio and keep center
ALT	Size, ignore grid setting
ALT+SHIFT	Size and keep aspect ratio, ignore grid setting
ALT+CTRL	Size and keep center, ignore grid setting
ALT+SHIFT+CTRL	Size and keep aspect ratio and keep center, ignore grid setting

Align

To align selected components, Function Designer offers the following methods:

For vertically aligning:

- [Align Left/Center/Right](#)

For horizontally aligning:

- [Align Top/Middle/Bottom](#)



One of the selected components is the anchor component. A dashed border indicates the anchor component. All other components will be aligned with the anchor component.

Align Left/Center/Right

To align selected components vertically with the left side of the anchor component, choose the menu command **Layout > Align > Left**.

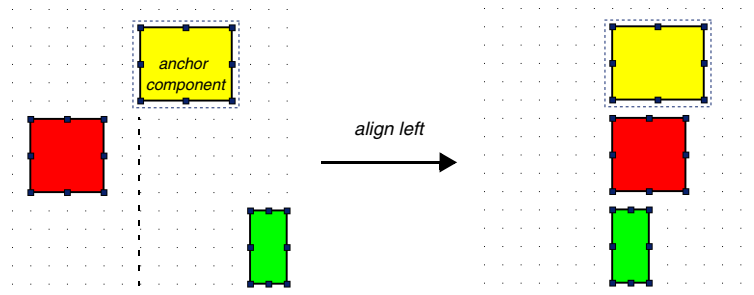


Figure 191. Aligning Components Left

To align selected components vertically with the center/right of the anchor component, choose the respective menu command:

- **Layout > Align > Center.**
- **Layout > Align > Right.**

Align Top/Middle/Bottom

To align selected components horizontally with the top/middle/bottom side of the anchor component, choose the required menu command:

- **Layout > Align > Top.**
- **Layout > Align > Middle.**
- **Layout > Align > Bottom.**



Rotate

Function Designer provides menu and toolbar commands for rotating components:

- **Rotate Free**
- **Rotate Left/Right**

Rotate Free

To rotate a component freely, do as follows:

1. Choose the **Layout > Rotate > Free** menu command or the  command from the **Rotate** toolbar.
2. Point on a component (single, selection, group). The cursor changes to .
3. Left-click and move the mouse until the component's preview is displayed with the wanted rotating angle.
4. Leave the mouse button.

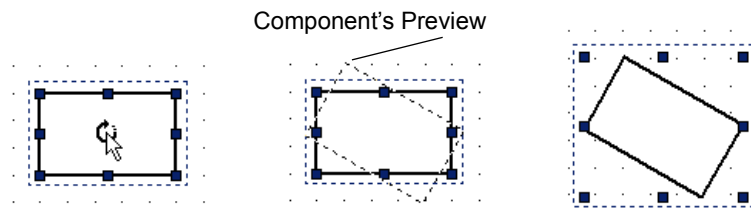




Figure 192. Component rotating free


Rotate Left/Right

To rotate the selected components by 90 degrees to the left/right, choose:

- **Layout > Rotate > Left** menu command or the  command from the **Rotate** toolbar.
- **Layout > Rotate > Right** menu command or the  command from the **Rotate** toolbar.

Flip

Designer provides menu and toolbar commands for flipping components horizontally/vertically. Choose:

- **Layout > Flip > Horizontal** menu command or the  command from the **Rotate** toolbar to flip horizontally.

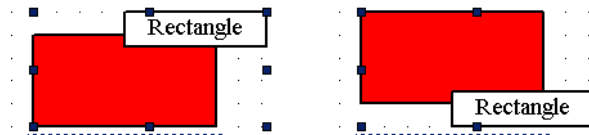



Figure 193. Flip Horizontal

- **Layout > Flip > Vertical** menu command or the  command from the **Rotate** toolbar to flip vertically.

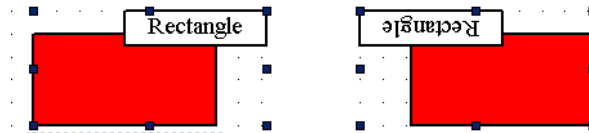


Figure 194. Flip Vertical

Order

Designer provides menu and toolbar commands for ordering components within the diagram/component document:

- **Bring to Front** means to move a component in the foreground of the diagram/component document so that no other component could overlap (cover) it.
- **Send to Back** means to move a component in the background of the diagram/component document so that all other components would overlap (cover) it.


- **Bring Forward** means to move a component one level higher (forward) within the internal component order of the diagram/component document.
- **Send Backward** means to move a component one level lower (backward) within the internal component order of the diagram/component document.



Because moving a component forward/backward regards all components of a diagram/component document, it can be that nothing seems to happen on executing the **Bring Forward/Send Backward** command. This is the case if the order of non-overlapping components is changed.

Bring to Front

To bring a component to front (to the highest level), select a component and choose one of the following methods:

- Choose the **Layout > Order > Bring to Front** menu command.
- Click the **Structure** toolbar button  .
- Choose the **Order > Bring to Front** command of the **Graphic Component Context Menu/Designer Component Context Menu** context menu.

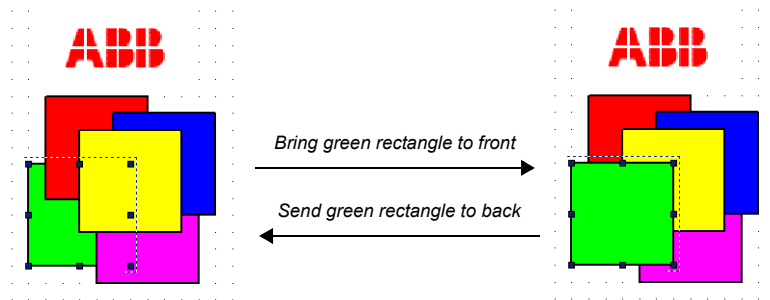



Figure 195. Component Structure Order: Bring to Front / Send to Back


Send to Back

To send a component to back (to the lowest level), select it and choose one of the following methods:

- Choose the **Layout > Order > Send to Back** menu command.
- Click the **Structure** toolbar button  .
- Choose the **Order > Send to Back** command of the **Graphic Component Context Menu/Designer Component Context Menu** context menu.

Bring Forward

To bring a component forward (to the next higher level), select it and choose one of the following methods:

- Choose the **Layout > Order > Bring Forward** menu command.
- Click the **Structure** toolbar button  .
- Choose the **Order > Bring Forward** command of the **Graphic Component Context Menu/Designer Component Context Menu** context menu.

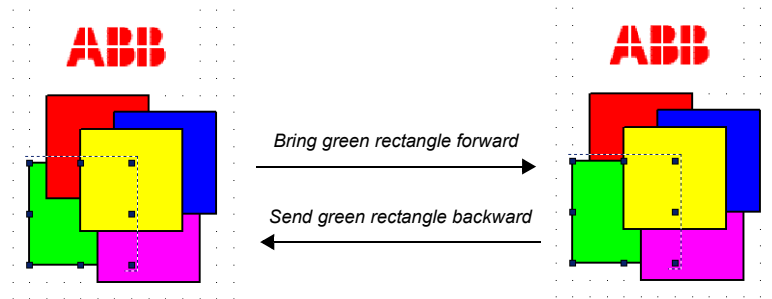



Figure 196. Component Structure Order: Bring Forward / Send Backward

Send Backward

To bring a component backward (to the next lower level), select it and choose one of the following methods:

- Choose the **Layout > Order > Send Backward** menu command.
- Click the **Structure** toolbar button  .

- Choose the **Order > Send Backward** command of the [Graphic Component Context Menu/Designer Component Context Menu](#) context menu.


Group

Several selected components can be grouped or ungrouped:

- [Group](#) several components to handle them as one component.
- [Ungroup](#) a component group to handle each component separated.


Group

To group several components, select them and choose one of the following methods:

- Choose the [Layout > Group](#) menu command.
- Click the [Structure](#) toolbar button  .
- Choose the **Grouping > Group** command of the [Graphic Component Context Menu/Designer Component Context Menu](#) context menu.

Ungroup

To ungroup a component group, select the group and choose one of the following methods:

- Choose the [Layout > Ungroup](#) menu command.
- Click the [Structure](#) toolbar button  .
- Choose the **Grouping > Ungroup** command of the [Graphic Component Context Menu/Designer Component Context Menu](#) context menu.

Cut/Copy/Paste/Delete Components

There are different workflows while cutting/copying/pasting/deleting components:

- In Component view,
 - you cut/copy/paste/delete graphic components (primitives) to define a designer component symbol or symbol template.

- In Diagram view,
 - you cut/copy/paste/delete designer components within the diagram. Designer structure gets automatically updated for function component Aspect Objects.
 - you paste designer components copied from another diagram or from designer structure. Designer structure gets automatically updated for designer component Aspect Objects.
 - you cut/copy/paste/delete graphic components on the master page, logic, background, or annotation layer.
- From Diagram view,
 - you cut/copy designer components, to paste them into another designer diagram or into the designer structure. Designer structure gets automatically updated for designer component Aspect Objects.



Cut in diagram, paste in Plant Explorer object browser (Functional Structure) is not supported.

Cut in Plant Explorer object browser (Functional Structure), paste in diagram is not supported.

In each case, the following actions are possible:

- [Cut and Paste](#)
- [Copy and Paste](#)
- [Delete](#)



The behavior on pasting symbols respectively connection links depends on the settings done in the [Options](#) dialog (see [Paste](#)).

Cut and Paste

To cut and paste components via menu commands, use one of the following methods:

- [To cut and paste components inside designer diagram via context menu](#)
- [To cut and paste components inside designer diagram via menu command](#)

You can also move components via drag & drop. For it, see also:

- To move components through drag-and-drop with left-click
- To move components through drag-and-drop with right mouse

To cut and paste components inside designer diagram via context menu

1. Select the wanted components (see [Selecting Components](#))
2. Click right over the selected components and choose **Cut** from the [Graphic Component Context Menu](#).
3. Click right on the wanted target location.
4. Choose **Paste** from the [Graphic Component Context Menu](#).

To cut and paste components inside designer diagram via menu command

1. Select the wanted components (see [Selecting Components](#))
2. Choose the menu command **Edit > Cut**. The selected components are removed from the diagram/component document window and copied to the Clipboard.
3. Click on the wanted target location.
4. Choose the menu command **Edit > Paste**. The selected components are pasted from the Clipboard.



If step 3 is left, the pasted components are located into the upper left corner of the diagram/component window.



Cut/Paste Function Blocks and Control Modules in Function Diagram:
On contrast to Delete, Cut in the diagram does NOT delete the Control Structure aspect, only the Functional Structure aspect. A Paste will create a "copy" of the Function Block / Control Module.

Copy and Paste

To copy and paste components inside a designer diagram, use one of the following methods:

- To copy components inside designer structure through left mouse drag and drop

- To copy components inside designer diagram through right mouse drag-and-drop
- To copy components inside designer diagram through context menu
- To copy components inside designer diagram through menu command



The designer structure is updated automatically for designer component Aspect Objects.

To copy and paste designer component Aspect Objects inside designer structure, use one of the following methods:

- To copy a designer component Aspect Object inside designer structure through drag and drop
- To copy a designer component Aspect Object inside designer structure through context menu

To copy components inside designer structure through left mouse drag and drop

1. Select the required components (see [Selecting Components](#))
2. Left-click on one of the selected components.
3. Press CTRL and moves the mouse to the wanted location.
4. Release CTRL key and mouse button.

The selected component(s) are copied according to the modifier key(s):

- ALT ignores grid setting.

Table 52. Modifier keys for copying components via left mouse

Key	Description
CTRL	Copy
SHIFT+CTRL	As CTRL, ignore SHIFT
ALT+CTRL	Copy, ignore grid setting
ALT+SHIFT+CTRL	As ALT+CTRL, ignore SHIFT

To copy components inside designer diagram through right mouse drag-and-drop

1. Select the required components (see [Selecting Components](#))
2. Right-click on one of the selected components.
3. Move the mouse to drag the selected component(s).
4. Release the mouse button. A context menu is displayed.
5. Choose **Copy Here** from the context menu.

Table 53. Modifier keys for moving via right mouse key

Key	Description
None	Displays context menu: Move Here Copy Here Cancel
SHIFT	Moves only in x/y direction relative to center. Displays context menu: Move Here Copy Here Cancel
CTRL	As None
SHIFT+CTRL	As CTRL, ignore SHIFT
ALT	Ignores grid setting. Display context menu: Move Here Copy Here Cancel.

Table 53. Modifier keys for moving via right mouse key (Continued)

Key	Description
ALT+SHIFT	Ignore grid setting and move only in x/y direction relative to center. Display context menu: Move Here Copy Here Cancel.
ALT+CTRL	As None
ALT+SHIFT+CTRL	As ALT+CTRL, ignore SHIFT

To copy components inside designer diagram through context menu

1. Select the wanted components (see [Selecting Components](#))
2. Click right over the selected components and choose **Copy** from the [Graphic Component Context Menu](#).
3. Click right on the wanted target location.
4. Choose **Paste** from the [Graphic Component Context Menu](#).

To copy components inside designer diagram through menu command

1. Select the wanted components (see [Selecting Components](#))
2. Choose the menu command **Edit > Copy**. The selected components are copied to the Clipboard.
3. Click on the wanted target location.
4. Choose the menu command **Edit > Paste**. The selected components are pasted from the Clipboard.



If step 3 is left, the pasted components are located moved a little bit right down from the components' source location to make the copied components conspicuously.

To copy a designer component Aspect Object inside designer structure through drag and drop

1. Open the designer structure in the Engineering Studio Workplace.
2. Select the wanted object (component).
3. Drag the object and drop it on the wanted target location in the designer structure.



The copied designer component is automatically inserted in the designer diagram.

To copy a designer component Aspect Object inside designer structure through context menu

1. Open the designer structure in the Engineering Studio Workplace.
2. Select the wanted object (component).
3. Click right and choose **Copy** from the context menu.
4. Select the target location (object with designer aspect) in the designer structure.
5. Click right and choose **Paste** from the context menu.



The copied designer component is automatically inserted in the designer diagram.

Paste Options

If you copy (or cut and paste) a selected range of designer components, you can define which links (connections) from, to, or within the selection shall be copied together with the designer components. The following variants (or combinations of them) are possible:

- [Copying Designer Components without Links](#)
- [Copying Designer Components including Internal Links](#)
- [Copying Designer Components including External Input Links](#)
- [Copying Designer Components including External Output Links](#)

Copying Designer Components without Links

To copy a selected range of designer components without any links, unmark all check boxes as displayed in the following figure:

Figure 197 displays the result of copying a selected range of components (copy of Symbol2+3 to Symbol5+6):

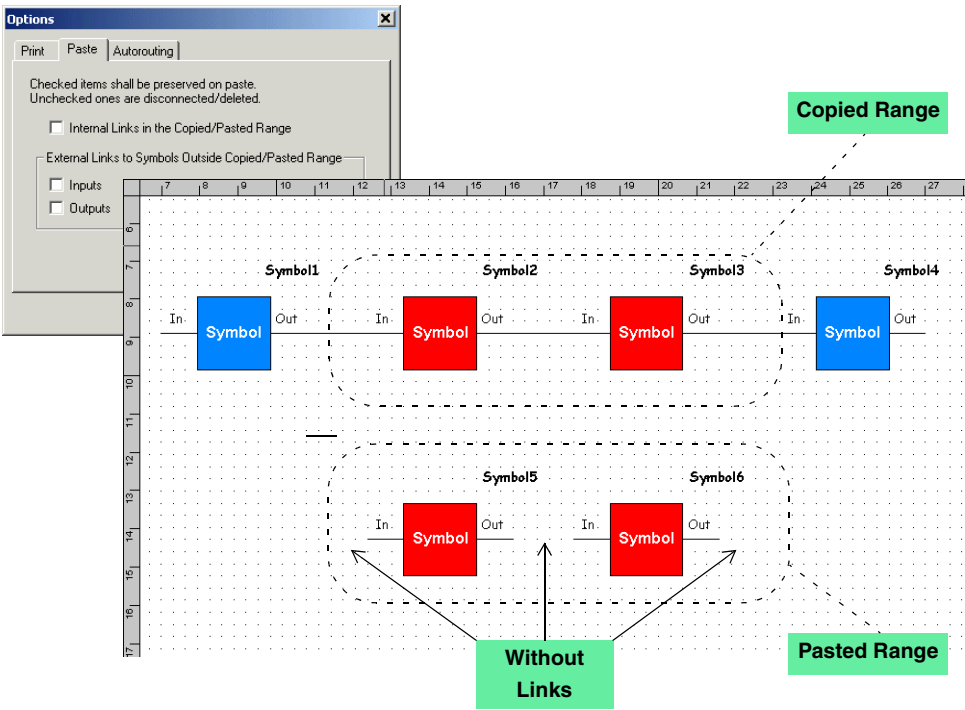


Figure 197. Paste Option for Internal Links

Copying Designer Components including Internal Links

To copy a selected range of designer components including internal links, mark the **Internal Links in the Copied/Pasted Range** check box of the paste options. The following figure displays an example for it (copy of Symbol2+3 to Symbol5+6):

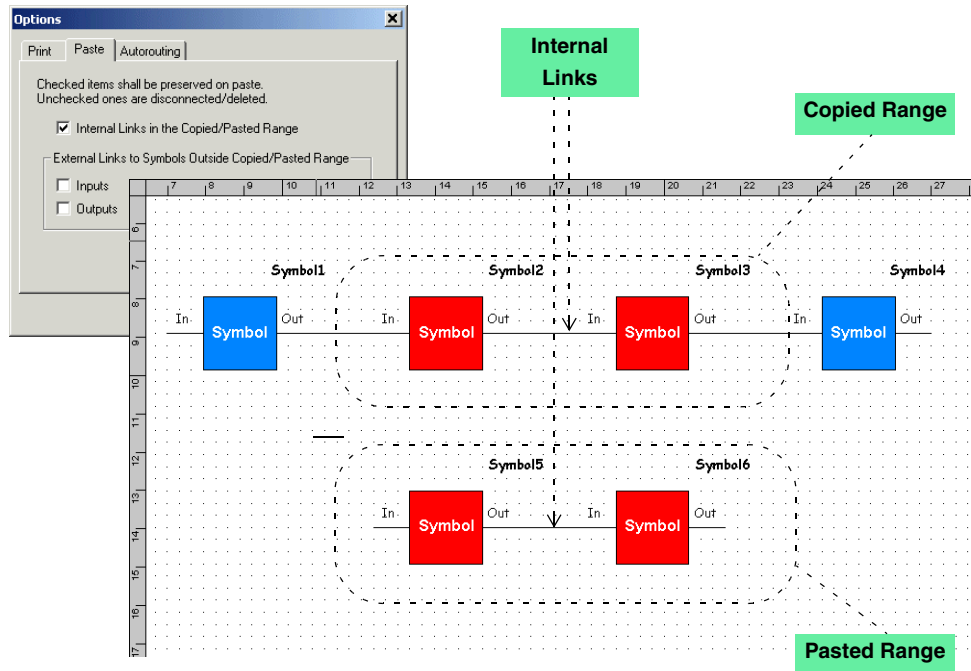


Figure 198. Paste Option for Internal Links, Example

Copying Designer Components including External Input Links

To copy a selected range of designer components including internal links, mark the **Internal Links in the Copied/Pasted Range** check box of the paste options. The following figure displays an example for it (copy of Symbol2+3 to Symbol5+6):

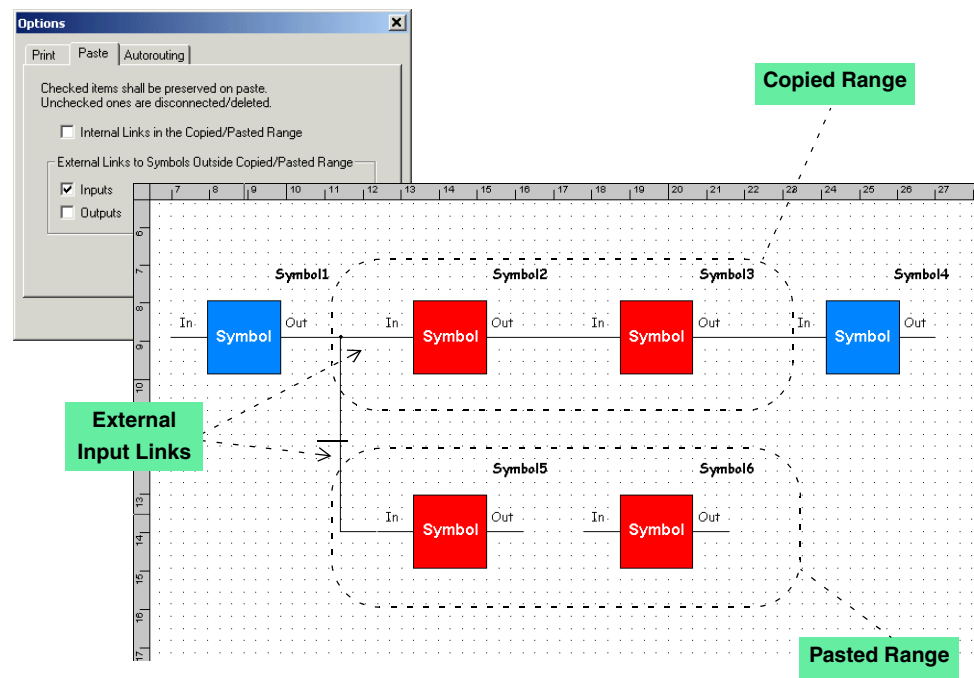


Figure 199. Paste Option for External Input Links

Copying Designer Components including External Output Links

To copy a selected range of designer components including internal links, mark the **Internal Links in the Copied/Pasted Range** check box of the paste options. The following figure displays an example for it (copy of Symbol2+3 to Symbol5+6):

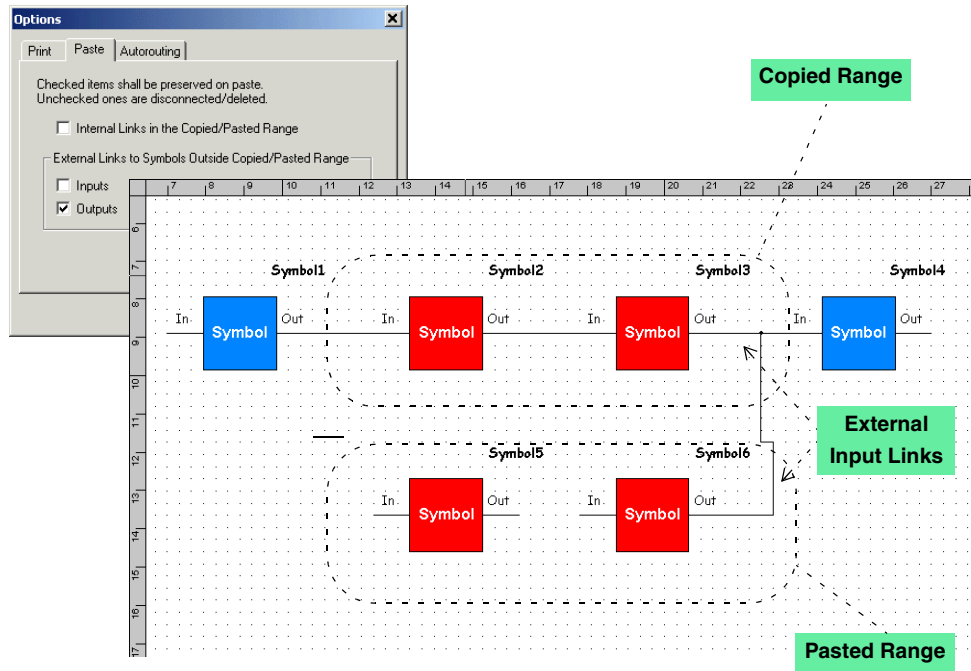


Figure 200. Paste Option for External Output Links

Delete

To delete components, choose one of the following methods:

- To delete components from a designer diagram
- To delete designer components from the designer structure
- To delete components via Components dialog

you can do it directly from the diagram, or you can use the [Component Properties](#) dialog:

- To delete components from the Function Diagram
- To delete components via Components dialog

To delete components from a designer diagram

1. Select the wanted component(s). See [Selecting Components](#).
2. Click the **DEL** key, or choose **Edit > Delete**.

The object is automatically removed from the designer structure.

To delete designer components from the designer structure

1. Open the designer structure in the Engineering Studio Workplace
2. Choose the component to be deleted.
3. Click right and choose **Delete** from the object-specific context menu.

The object is automatically removed from the designer diagram.

To delete components via Components dialog

1. Open the [Components](#) dialog.
2. Select the wanted component(s) from the list box.
(See also [Selecting Symbols via the Components Dialog](#).)
3. Click **Delete**.
4. Click **OK**.



Delete Function Block or Control Module: Function Blocks and Control Modules get inserted in Control Structure, too. If user deletes them in Plant Explorer from Functional Structure, they still exist in Control Structure and inside Control Builder.

As a consequence,

- user cannot create further instances with the same name and
- on copy diagram in Functional Structure, the still existing instance gets inserted into the copied diagram.

Delete Function Blocks and Control Modules either:

- within Function Diagram (this will remove them from both structures),
- within Plant Explorer with object context menu, item **Advanced > Delete from all Structures**.

Defining Component Properties

Each component of a designer diagram is described by its properties. These properties can be adjusted in appropriate dialogs.

- Ambient properties valid for the whole diagram document can be adjusted in the [Ambient Properties](#) dialog.
- Default graphical properties common for newly created graphical components (primitives) can be adjusted in the [Default Properties](#) dialog.
- Graphic properties for a single selected component or for multiple selected components can be adjusted in the [Component Properties](#) dialog.
- Aspect properties / parameters for the designer diagram, for a single selected designer component, or for a single selected port or can be adjusted in the corresponding dialog.

Ambient Properties

In this dialog, you set ambient properties which are valid for the whole diagram. E.g. background color for the diagram, selection handle colors, foreground and background color for ActiveX Controls and so on. The appropriate menu command is only accessible in Function Designer's Diagram view.

To access the **Ambient Properties** dialog, verify that designer's Diagram view is active, then choose menu command **Edit > Ambient Properties**. The dialog displays the following property pages:

- [Colors](#)
- [ActiveX Controls](#)
- [Off Page Connector](#)

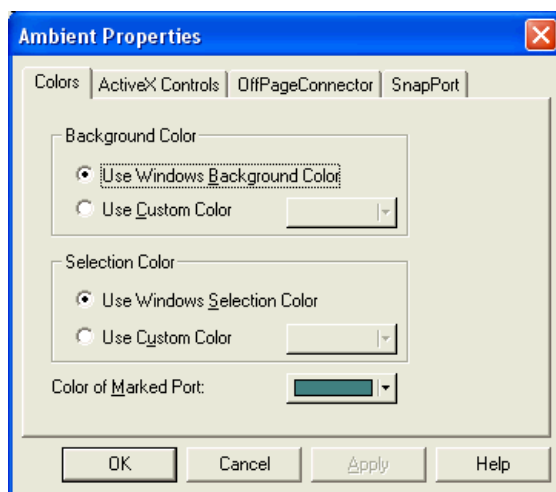


Figure 201. Ambient Properties Dialog

The **Ambient Properties** dialog contains the following buttons common for all dialog pages:

- **Colors**
Click on the **Colors** tab to access the ambient properties on the [Colors](#) page.
- **ActiveX Controls**
Click on the **ActiveX Controls** tab to access the ambient properties on the [ActiveX Controls](#) page.

- **Off Page Connector**

Click on the **Off Page Connector** tab to access the ambient properties on the [Off Page Connector](#) page.

- **OK**

Click **OK** to save your settings and to close the dialog.

- **Cancel**

Click **Cancel** to quit the dialog without saving your setting modifications.

- **Apply**

Click **Apply** to save your settings and leave the dialog open.

Colors

On the **Colors** page of the [Ambient Properties](#) dialog, you have access to the properties for setting the colors for background, selection handles and ports.

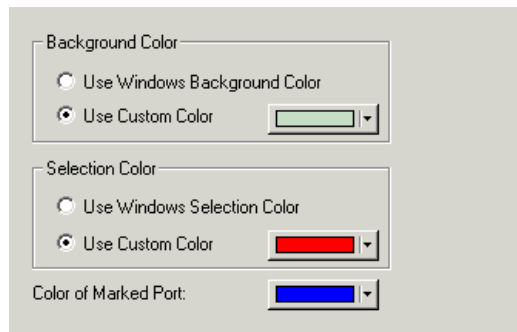


Figure 202. Ambient Properties Dialog: Colors Page

This page contains the following property items:

Background Color

Groups the controls for setting the background color.

- **Use Windows Background Color**

Set this option, if the diagram shall have the same background color as the operating system windows.

- **Use Custom Color**

Set this option to get access to the drop-down color table from where you can choose the wanted background color for the diagrams.

Selection Color

Groups the controls for setting the color of the selection handles.

- **Use Windows Selection Color**

Set this option, if the diagram shall have the same selection color as the operating system windows.

- **Use Custom Color**

Set this option to get access to the drop-down color table from where you can choose the wanted selection (handle) color for the diagrams.

Color of Marked Port

- **Color of Marked Port**

Set this option to get access to the drop-down color table from where you can choose the wanted color of marked ports.

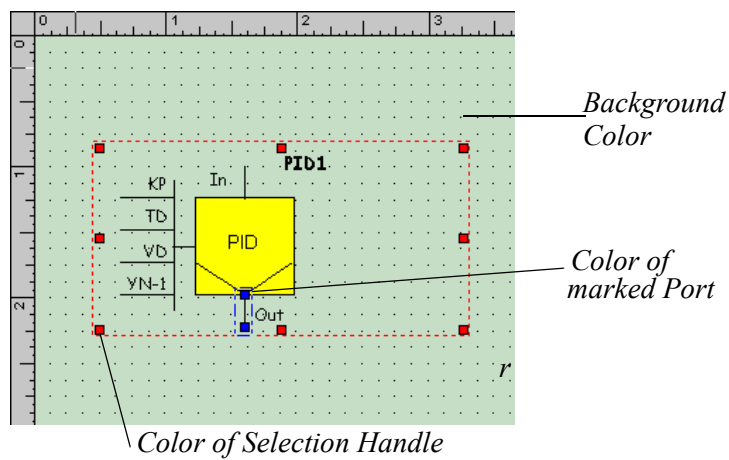


Figure 203. Ambient Properties, Modified Colors (Example)

For more colors than the offered ones, click on **Other** to open the [Color](#) dialog.



Figure 204. Drop-down Color Table

ActiveX Controls

On the **ActiveX Controls** page of the [Ambient Properties](#) dialog, you have access to properties defining the foreground and background color for ActiveX Controls.

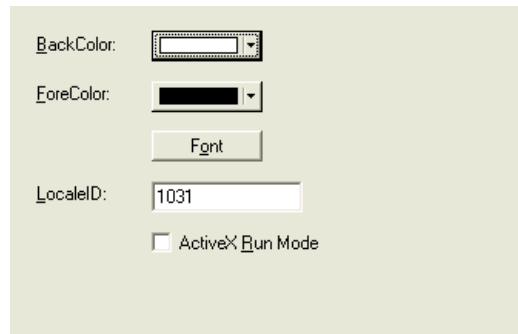


Figure 205. Ambient Properties Dialog: ActiveX Controls Page

This page contains the following property items:

- [Design Mode](#)
- [Background Color](#)
- [Use Windows Background Color](#)
- [Use Custom Color](#)
- **Back Color**

Choose the background color for ActiveX Controls from the drop-down color table by clicking on the down pointer and selecting a color by clicking on it.

For more colors, click on **Other** to open the [Color](#) dialog.



Figure 206. Drop-down Color Table

- **Fore Color**

Choose the foreground color for ActiveX Controls from the drop-down color table by clicking on the down pointer and selecting a color by clicking on it.

See [Figure 206](#). For more colors, click on **Other** to open the [Color](#) dialog.

- **Font**

Click on the font button. The [Font](#) dialog will appear where you can set font option such as type, style, size, and so on.

- **ActiveX Run Mode**

Set this option to enable “ActiveX Run Mode”. If enabled you can operate ActiveX Controls (e.g. press buttons, edit text fields, etc.). If disabled you can select ActiveX Controls and perform graphic operations (e.g. Copy, Move, Size, etc.).

If “ActiveX Run Mode” is enabled the diagram is set read-only.

Off Page Connector

On the **Off Page Connector** page of the [Ambient Properties](#) dialog, you have access to the source and sink of the Off-Page Connectors.

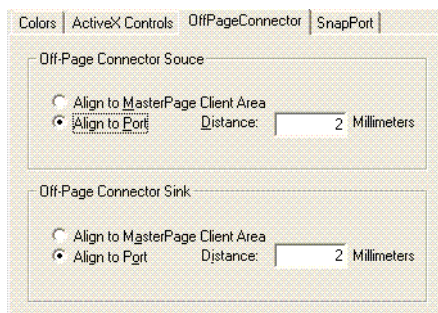


Figure 207. Ambient Properties Dialog: Off Page Connector Page

This page contains the following property items:

Off-Page Connector Source

- **Align to Master Page Client Area**
Set this option to align off-page connectors at the left side of the master page client area.
- **Align to Port**
Select this option to align off-page connectors near the connected port.
- **Distance**
Specify the distance between port and off-page connector.

Off-Page Connector Sink

- **Align to Master Page Client Area**
Set this option to align off-page connector sink to the right side of the master page client area.
- **Align to Port**
Select this option to align off-page connectors near the connected port.
- **Distance**

Specify the distance between port and off-page connector.

Font

Use this standard dialog to set the font type, style, size, color and some other effects such as strikethrough or underline. The font options are valid for the current selected object.

This dialog will be opened from other designer dialogs, for example, by clicking a **Font** button (see [Figure 205](#)).

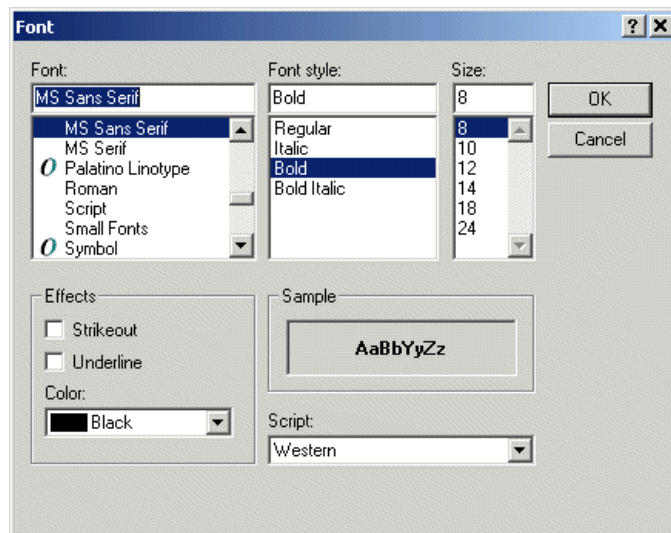


Figure 208. Font Dialog

Component Properties

Graphic Components are defined by a number of graphical properties. To determine which properties are associated with such a component, you need to open the **Component Properties** dialog:

Setting Component Properties

To access the **Component Properties** dialog, select one or more components, and choose one of the following methods:

- Choose the **Edit > Default Properties** menu command.
- Right-click on a component, and choose **Properties** from the context menu.
- Click the corresponding toolbar button from the **Drawing** toolbar.
- Click the **Properties** button from the **Components** dialog.

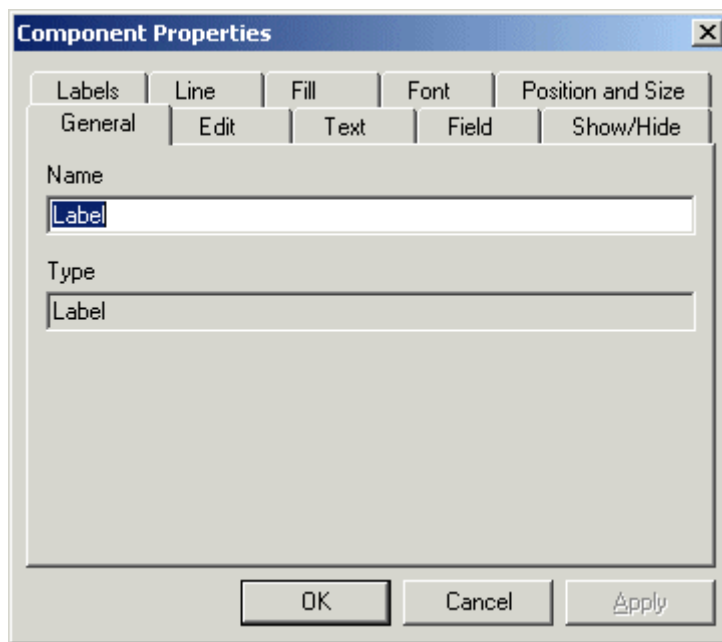


Figure 209. Component Properties Dialog: General Page



If the dialog is open from the **Components** dialog, the dialog's title will be **Properties** instead of **Component Properties**.

Depending on the selected component(s), the appropriate pages are displayed.

The **Component Properties** dialog contains the following tabs (pages) and buttons accessible from all dialog pages:

- [Edit](#)
- [Field](#)
- [Fill](#)
- [Font](#)
- [General](#)
- [Labels](#)
- [Line](#)
- [Position and Size](#)
- [Text](#)
- [OK](#)
- [Cancel](#)
- [Apply](#)
- **Edit**

Click on the **Edit** tab to access the properties on the [Edit](#) page.

- **Field**

Note: Not available for Topology Designer

Click on the **Field** tab to access the properties on the [Field](#) page.

- **Fill**

Click on the **Fill** tab to access the properties on the [Fill](#) page.

- **Font**

Click on the **Font** tab to access the properties on the [Font](#) page.

- **General**

Click on the **General** tab to access the properties on the [General](#) page.

- **Labels**

Note: Not available for Topology Designer

Click on the **Labels** tab to access the properties on the [Labels](#) page.

- **Line**

Click on the **Line** tab to access the properties on the [Line](#) page.

- **Position and Size**

Click on the **Position and Size** tab to access the properties on the [Position and Size](#) page.

- **Text**

Note: Not available for Topology Designer

Click on the **Text** tab to access the properties on the [Text](#) page.

Edit

On the **Edit** page of the [Component Properties](#) dialog, you have access to properties defining which editing actions you can perform on a component. For instance, you can prevent stretching a component.

This page is identical with the [Edit](#) page of the [Default Properties](#) dialog.

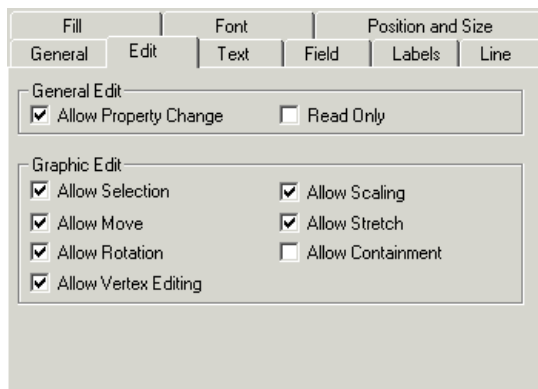


Figure 210. Component Properties Dialog: Edit Page

The page contains the following property items:

- [Read Only](#)
- [Allow Selection](#)
- [Allow Move](#)
- [Allow Rotation](#)
- [Allow Vertex Editing](#)
- [Allow Scaling](#)
- [Allow Stretch](#)
- [Allow Containment](#)

- **Read Only**

Check this option to lock the component for any editing actions. Then, it is no longer possible to change the position, appearance, or any other graphical properties of the component. All editing options, except **Allow Selection**, will be disabled.

- **Allow Selection**

Check this option to allow selection of the component using the mouse.



If this option is not set, the component can only be selected via the [Components](#) dialog.

- **Allow Move**

Check this option to allow moving the component within designer diagrams.

- **Allow Rotation**

Check this option to allow rotating the component.

- **Allow Vertex Editing**

Check this option to allow editing (adding, moving, deleting) the component's vertices.

- **Allow Scaling**

Check this option to allow changing the component's size. This option is corresponding to the option [Allow Stretch](#):

- If the option **Allow Stretch** is set, on changing the component's size, height and width can be changed separately.
- If the option **Allow Stretch** is not set, on changing the size of the component, the relationship between height and width is kept.



If this option is turned off, the option **Allow Stretch** is not accessible.

- **Allow Stretch**

Check this option to allow changing the component's height and width separately. This option is only active if the option **Allow Scaling** is set.

- **Allow Containment**

Check this option to allow that on moving the selected component, all other components overlapped by (contained in) the selected one will automatically moved, too.

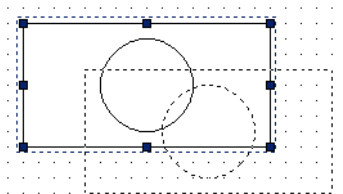


Figure 211. Component Property: Allow Containment

[Figure 211](#) displays an example for moving a rectangle overlapping (containing) a circle. Both components are moved, although only the rectangle is selected and the components are not grouped.

Field

On the **Field** page of the Component Properties dialog, you define document and aspect property references.

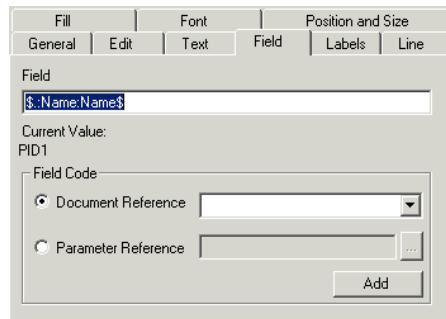


Figure 212. Component Properties Dialog: Field Page

This page contains the following property items:

- [Field](#)
- [Current Value](#)
- [Field Code](#)
- [Document Reference](#)
- [Parameter Reference](#)
- [Add](#)
- **Field**

This field contains the **Field Code** defined as **Document Reference** or **Aspect Property Reference**. You can edit this field code definition and mix fixed texts with references,

e.g. 'This is page \$Page of \$LastPage'. You can also enter multiline field codes by pressing the 'Return' key as line separator. In that case make sure to also check **Multiple Lines** in Component Properties Dialog: [Text](#) page.

- **Subscribe for Live Data**
Check **Subscribe for Live Data** if the **Field Code** contains references to properties retrieved from an OPC server, i.e. on-line values. This setting just prepares the field for display of on-line values. To activate/start on-line display for all such prepared fields, use command [Online > Subscribe for Live Data](#).
- **Show Last Value**
Check **Show Last Value** in order to display the recent on-line value after [Online > Subscribe for Live Data](#) has stopped. Uncheck in order to display an empty string instead.
- **Current Value**
This field displays the current value of the **Field Code** entered in **Field**.
- **Field Code**
Groups the fields for setting the **Document Reference** or **Aspect Property Reference**.
- **Document Reference**
Choose a predefined document reference from the drop-down list box. The following document references can be used:

Table 54. Document References as Field Parameters

Document Reference	Description
\$?	Optional prefix to be displayed if evaluation fails. E.g. \$?*OPCValue displays a * if no OPC value can be retrieved.
\$AllocDetails (Not valid for Topology Designer.)	Allocation details as defined in the Allocation > Allocate Diagram command. E.g. data flow order, allocatable group name and color.
\$AllocGroup (Not valid for Topology Designer.)	Name of assigned Allocatable Group Aspect Object
\$Date	Current date
\$GroupName	Group name

Table 54. Document References as Field Parameters (Continued)

Document Reference	Description
\$LastPage	Last page number of the current document
\$Name	Name property of the component
\$NextPage	Page number following to the current page
\$OPCForced	Online > Subscribe for Live Data : On-line indication of a forced signal.
\$OPCIOStatus	Online > Subscribe for Live Data : On-line indication of the IO status.
\$OPCIOValue	Online > Subscribe for Live Data : On-line IO value of associated IO signal.
\$OPCValue	Online > Subscribe for Live Data : On-line value of associated port, link, or signal from OPC server.
\$Order	Data Flow Order
\$Page	Current page number
\$PgComment	Current page comment.
\$RefPage	Off-Page Connector: Referenced page number
\$RefPort	Off-Page Connector: Referenced port
\$RefSymbol	Off-Page Connector: Referenced symbol (function component)
\$SymbolName	Symbol name
\$Time	Current time

- **Aspect Property Reference**

Edit an aspect property reference, e.g. '\$.:Name:Name\$'. Or press '...' to open the Aspect Property Reference dialog.

- **Add**

Click this button to add the selected Document/Aspect Property Reference to the **Field** field.

Fill

On the **Fill** page of the Component Properties dialog, you have access to properties defining how a component is filled. The most commonly used fill is a solid fill, which is a foreground color with no hatching. You can also choose a hatched fill pattern with a background color.

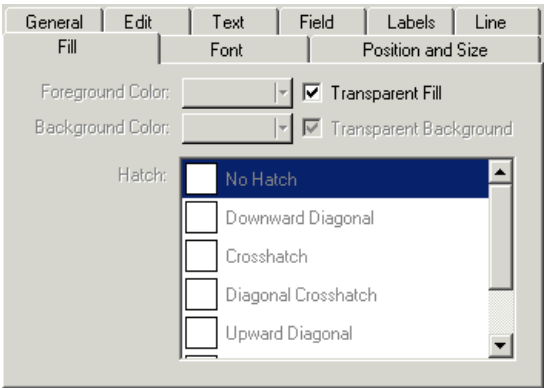


Figure 213. Component Properties Dialog: Fill Page

This page which is identical with the **Fill** page of the **Default Properties** dialog. It contains the following property items:

- **Foreground Color**
From the **Foreground Color** drop down list, select the component's foreground (hatch) color.
- **Transparent Fill**
Mark this option, to make the component transparent, that is to deactivate all fill settings. Neither the background nor the hatch will be displayed.

- **Background Color**

From the **Background Color** drop down list, select the component's background color.

- **Transparent Background**

Mark this option, to make the background transparent. Only the hatch will be visible.

- **Hatch**

From the **Hatch** list box, select the hatch displayed as foreground in the component.

Font

On the **Font** page of the Component Properties dialog, you have access to properties defining how the text appears in a component. You can choose any True Type font and then apply any typographical effects such as italic or underline.

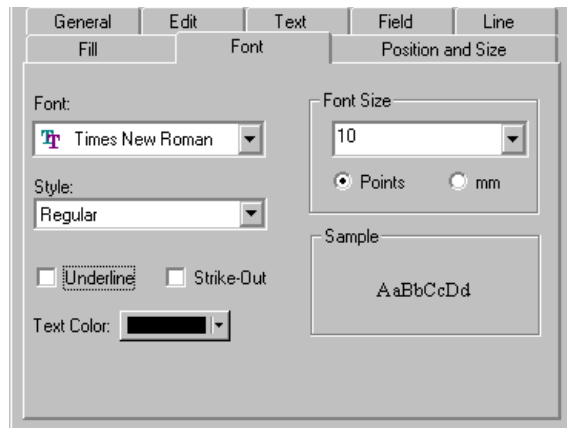


Figure 214. Component Properties Dialog: Font Page

This page which is identical with the [Font](#) page of the [Default Properties](#) dialog and nearly identical with the Font standard dialog. It contains the following property items:

-
-
-
- Underline
- Strike-Out
- Text Color
- Sample
-

Select the font wanted for the selected text from the list by clicking on it.

-

Select the font angle/weight wanted for the selected text from the list by clicking on it.

-

Select the font size wanted for the text from the list by clicking on it.

- **Underline**

Mark this option to underline the selected text.

- **Strike-Out**

Mark this option to strike-out (strike through) the selected text.

- **Text Color**

Click on the down pointer and select a color by clicking on it.

For more colors, click on **Other** to open the [Color](#) dialog.



Figure 215. Color Drop-Down Table

- **Sample**

General

On the **General** page of the Component Properties dialog, you have access to properties such as the name and type of the component. The properties displayed here depend on the type of the component:

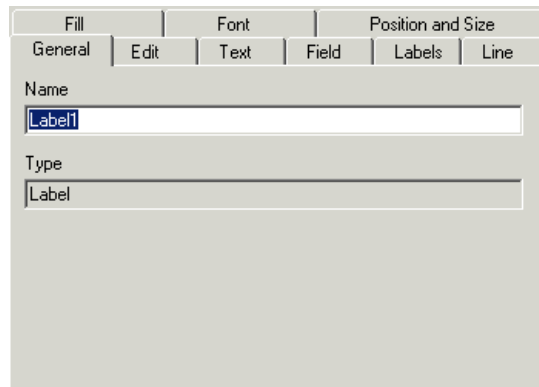


Figure 216. Component Properties Dialog: General Page

This page contains the following property items:

- [Name](#)

- **Type**

- **Name**

Displays the graphical name of the selected component. If only one component is selected, the name can be changed.

- **Type**

Displays the graphical type of the selected component. If you have selected several components, the field displays the entry <Multiple Types Selected>.

Labels

On the **Labels** page of the Component Properties dialog, you have access properties defining the orientation of the label component.

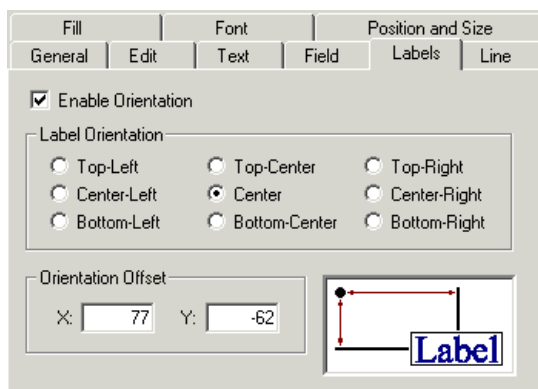


Figure 217. Default Properties Dialog: Labels Page

This page which is identical with the **Labels** page of the **Default Properties** dialog, contains the following property items:

- **Enable Orientation**
- **Label Orientation**
- **Top-Left**

- Center-Left
- Bottom-Left
- Top-Center
- Center-Center
- Bottom-Center
- Top-Right
- Center-Right
- Bottom-Right
- Orientation Offset
- X
- Y
- <Orientation Preview>

- **Enable Orientation**

Check to enable the following label orientation options.

- **Label Orientation**

Groups the options for label orientation.

- **Top-Left**

Check this option to place the label on top-left of the selected component.

- **Center-Left**

Check this option to place the label on center-left of the selected component.

- **Bottom-Left**

Check this option to place the label on bottom-left of the selected component.

- **Top-Center**

Check this option to place the label on top-center of the selected component.

- **Center-Center**

Check this option to place the label on center-center of the selected component.

- **Bottom-Center**
Check this option to place the label on bottom-center of the selected component.
- **Top-Right**
Check this option to place the label on top-right of the selected component.
- **Center-Right**
Check this option to place the label on center-right of the selected component.
- **Bottom-Right**
Check this option to place the label on bottom-right of the selected component.
- **Orientation Offset**
Groups the properties for setting the orientation offset. Orientation offset means moving the label orientation in x and y direction.
- **X**
Enter the value (in the selected measurement unit, e.g. 'mm') for moving the orientation offset in x direction. Positive value result in moving to right, a negative values result in moving to the left.
- **Y**
Enter the value (in the selected measurement unit, e.g. 'mm') for moving the orientation offset in y direction. Positive values result in moving down, negative values result in moving up.
- **<Orientation Preview>**
Displays the meaning of the label orientation offset in x (horizontal) and y (vertical) direction.

Line

On the **Line** page of the Component Properties dialog, you have access to properties defining how the lines appear in the component. You can set color, style and width.

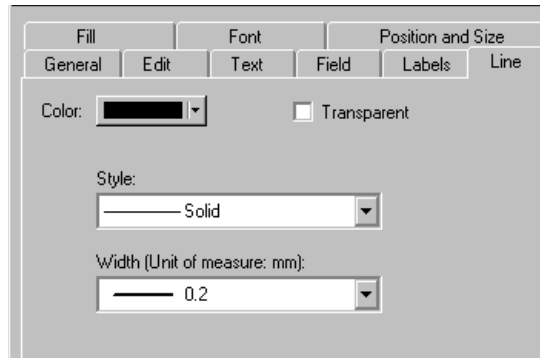


Figure 218. Component Properties Dialog: Line Page

This page which is identical with the [Line](#) page of the [Default Properties](#) dialog, contains the following property items:

- [Color](#)
- [Transparent](#)
- [Style](#)
- [Width](#)
- **Color**

Use this drop down list for selecting the line color for the component's border.

- **Transparent**

Check this option to make the component's border transparent, that is to turn off the display.

- **Style**

From the **Style** list box, select the line style for the component's border.

- **Width**

From the **Width** list box, select the line width for the component's border.

Position and Size

On the **Position and Size** page of the Component Properties dialog, you have access to properties giving information about the position and size of the component. It is only shown when a single component is selected. The position and size is shown using the current measurement unit.

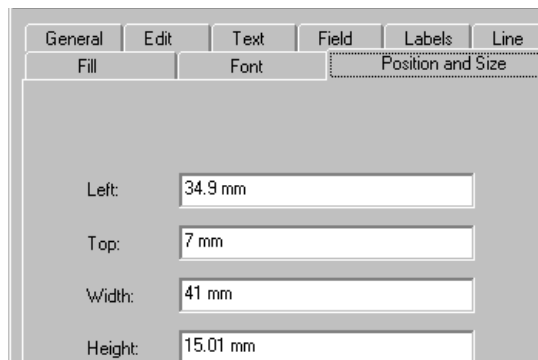


Figure 219. Component Properties Dialog: Position and Size Page

You can also edit the displayed position and size to manually adjust or align components.

This page contains the following property items:

- [Left](#)
- [Top](#)
- [Width](#)
- [Height](#)
- [Width](#)

- **Left**
Displays the component's position from the left border in the unit defined in [Measurements and Size](#) dialog.
- **Top**
Displays the component's position from the top border in the unit defined in [Measurements and Size](#) dialog.
- **Width**
Displays the component's width in the unit defined in [Measurements and Size](#) dialog.
- **Height**
Displays the component's height in the unit defined in [Measurements and Size](#) dialog.

Show / Hide

The **Show/Hide** page is similar to the **Field** page of the Component Properties dialog. You define document and aspect property references in order to show or hide the selected graphic component or primitive.

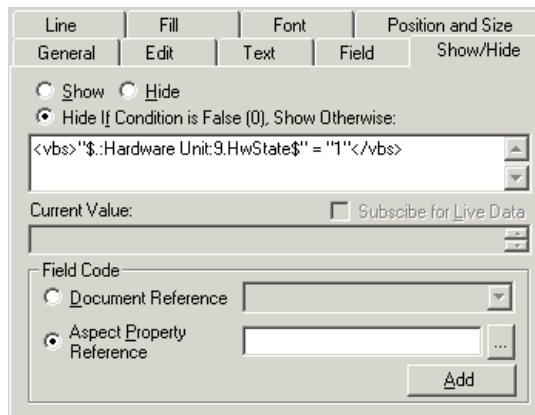


Figure 220. Component Properties Dialog: Show/Hide Page

This page contains the following property items:

- [Left](#)

- [Top](#)

- [Width](#)

- [Height](#)

- [Width](#)

- **Show**

Check this option to always show the graphic component.

- [Left](#)

- [Top](#)

- [Width](#)

- [Height](#)

- [Width](#)

- **Hide**

Check this option to always hide the graphic component.

- [Left](#)

- [Top](#)

- [Width](#)

- [Height](#)

- [Width](#)

- **Hide if Condition is False(0), Show Otherwise**

Check this option to show or hide the graphic component based on a condition. You can enter a Visual Basic script enclosed into `<vbs>` and `</vbs>`. And you can combine that script with any document or aspect property reference, see [Field](#).

- **Current Value**

This field displays the current value of the **Condition**.

- **Subscribe for Live Data**

Check **Subscribe for Live Data** if the **Condition** contains references to properties retrieved from an OPC server, i.e. on-line values. This setting just prepares the field for display of on-line values. To activate/start on-line display for all such prepared fields, use command [Online > Subscribe for Live Data](#).

- **Field Code**

Groups the fields for setting the **Document Reference** or **Aspect Property Reference**.

- **Document Reference**

Choose a predefined document reference from the drop-down list box, see [Document References as Field Parameters](#).

- **Aspect Property Reference**

Edit an aspect property reference, e.g. '\$.:Name:Name\$'. Or press '...' to open the Aspect Property Reference dialog.

- **Add**

Click this button to add the selected Document/Aspect Property Reference to the **Condition** field.

Text

On the **Text** page of the Component Properties dialog, you have access to properties allowing the user to enter a string value and set properties that determine how the text is displayed and edited. You can use a check box to toggle between single line and multiple line text. If multi-line text is enabled, the **Word Break** check box

allows you to toggle automatic word break on and off. Horizontal alignment can be set to left, center, or right. Vertical alignment can be set to top, center, or bottom.

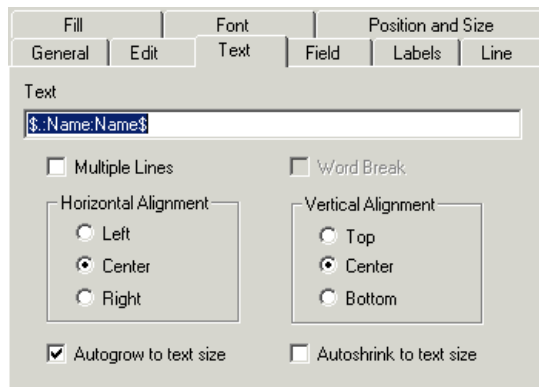


Figure 221. Component Properties Dialog: Text Page

- This page contains the following property items:
- [Text](#)
- [Multiple Lines](#)
- [Word Break](#)
- [Left](#)
- [Center \(horizontal\)](#)
- [Right](#)
- [Top](#)
- [Center \(vertical\)](#)
- [Bottom](#)
- [Autogrow to text size](#)
- [Autoshrink to text size](#)**Text**

In the **Text** field, enter the text, displayed within the component.

- **Multiple Lines**

Check this option to prepare the text location for multiple lines. If this option is set, you can skip to the next line by pressing the ENTER key.

- **Word Break**

Check this option to retain the component's width on writing text of any length. New text is automatically written in a new line, if the right border is reached.

Unmark this option, if text shall only be written in a new line after pressing the ENTER key.

- **Left**

Check this option of the **Horizontal Alignment** group box to align the component's text left.

- **Center (horizontal)**

Check this option of the **Horizontal Alignment** group box to align the component's text horizontally centered.

- **Right**

Check this option of the **Horizontal Alignment** group box to align the component's text right.

- **Top**

Check this option of the **Vertical Alignment** group box to align the component's text vertically on top.



This option is only accessible and can be set if the **Multiple Lines** option is not set.

- **Center (vertical)**

Check this option of the **Vertical Alignment** group box to align the component's text vertically centered.



This option is only accessible and can be set if the **Multiple Lines** option is not set.

- **Bottom**

Check this option of the **Vertical Alignment** group box to align the component's text vertically on bottom.



This option is only accessible and can be set if the **Multiple Lines** option is not set.

- **Auto grow to text size**

Check this option for automatically adjusting the component's width to the length of the entered text.

- **Auto shrink to text size**

Check this option for automatically shrinking the component's width to the length of the entered text.

Color

Use the **Color** dialog to define a color for the previously selected component(s).

Access this dialog, for example, from the **Font** page of the **Component Properties** dialog by clicking on the **Other** button of the **Text Color** drop-down list box. Such color drop-down lists are also offered in some other dialogs.

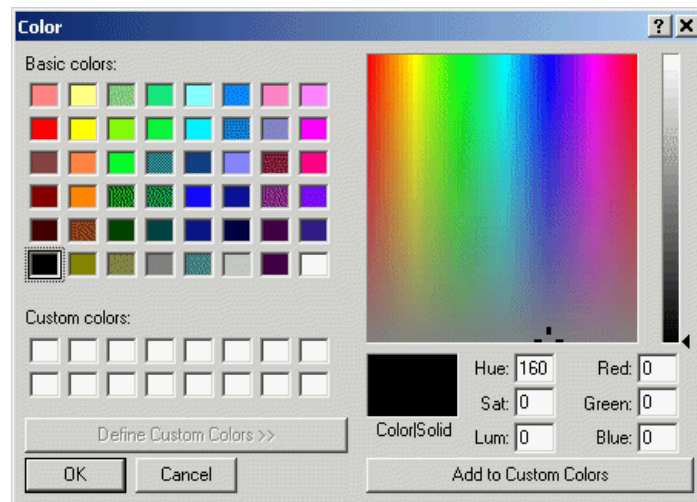


Figure 222. Color Dialog

The **Color** dialog contains the following property items:

- Basic Colors
- Custom Colors
- Define Custom Colors
- Color | Solid
- Hue
- Sat
- Lum
- Red
- Green

- [Blue](#)
- [Add to Custom Colors](#)
- [OK](#)
- [Cancel](#)
- **Basic Colors**

Displays the basic colors available. You can choose the wanted color by clicking on it.
- **Custom Colors**

User defined colors added from the color matrix. You can choose the wanted color by clicking on it.
- **Define Custom Colors**

Not active.
- **Color | Solid**

Displays a preview of the color selected from the color matrix.
- **Hue**

Specifies the hue of the color selected from the color matrix.
- **Sat**

Specifies the saturation of the color selected from the color matrix.
- **Lum**

Specifies the luminosity of the color selected from the color matrix.
- **Red**

Specifies the Red value in the sense of a RGB color value.
- **Green**

Specifies the Green value in the sense of a RGB color value.
- **Blue**

Specifies the Blue value in the sense of a RGB color value.

- **Add to Custom Colors**

Click this button to add the color selected from the color matrix to the **Custom colors**.

- **OK**

Click **OK** to save your settings and to close the dialog.

- **Cancel**

Click **Cancel** to quit the dialog without saving your setting modifications.

Components

Use the **Components** dialog to navigate to components, to delete components, or to get access to the properties of components.

Open the **Components** dialog via the **Edit > Components** menu command.

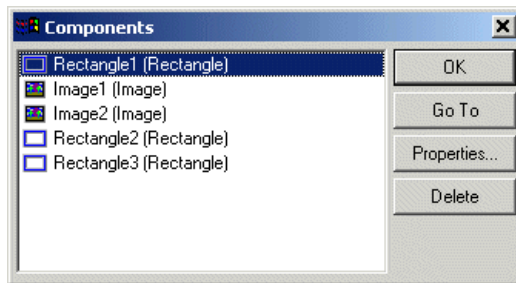


Figure 223. Components Dialog

- **OK**

Click this button to close the window.

- **Go To**

Click this button to navigate to the selected component(s)



The navigation will result in displaying the selected component in the center of the document window.

If you have selected one component, it will be selected in the document window.

If you have selected several components, the focus of the window will be the center of the area built by the selected components.

- **Properties**

Click this button to open the [Component Properties](#) dialog.

- **Delete**

Click this button to delete the component from the diagram.

Setting Component Properties

To set the properties of one or more components, do as follows:

1. Select the required component(s) as described in [Selecting Components](#).
2. Open the Component Properties dialog by choosing one of the following methods:
 - Right-click the selected components on the Function Designer window.
 - Choose the menu command **Edit > Component Properties**.
 - Click the appropriate toolbar button.
3. Click on the appropriate tab, to open the page containing the wanted properties.
4. Set the properties according to your wishes.
5. Click **OK**.

Alternatively, you can open the **Properties** dialog for getting access to the selected components' properties, do as follows:

1. Open the [Components](#) dialog.
2. Select the wanted component(s) from the list box. See [Component Properties](#).
3. Click **Properties**. See [Component Properties](#).
4. Click **OK** in the [Components](#) dialog after having finished the work in the **Properties** dialog.

Default Properties

In this dialog, you can define default properties for graphic components. The default properties are automatically used by designer for newly created components. Default properties are not component-specific.

You can define

- default edit properties, e.g. allow scale
- default label properties, e.g. enable orientation
- default line properties, e.g. line width
- default fill properties, e.g. fill color
- default font properties, e.g. font size.

To access the **Default Properties** dialog, choose the **Edit > Default Properties** menu command.

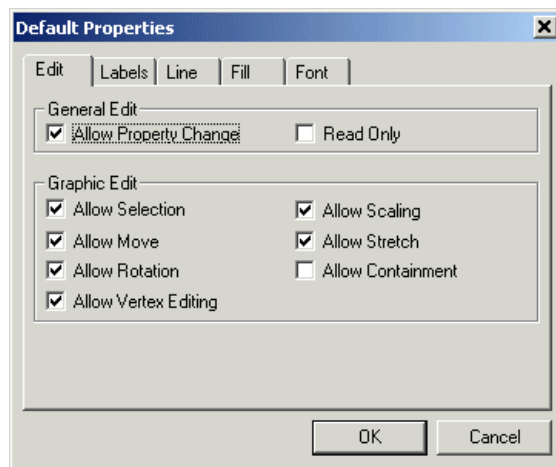


Figure 224. Default Properties Dialog

This dialog contains the following tabs and buttons common for all dialog pages:

- [Edit](#)

- [Fill](#)
- [Font](#)
- [Labels](#)
- [Line](#)
- [OK](#)
- [Cancel](#)
- [Apply](#)
- **Edit**

Click on the **Edit** tab to access the properties on the [Edit](#) page.

- **Fill**

Click on the **Fill** tab to access the properties on the [Fill](#) page.

- **Font**

Click on the **Font** tab to access the properties on the [Font](#) page.

- **Labels**

Click on the **Labels** tab to access the properties on the [Labels](#) page.

- **Line**

Click on the **Line** tab to access the properties on the [Line](#) page.

- **OK**

Click **OK** to save your settings and to close the dialog.

- **Cancel**

Click **Cancel** to quit the dialog without saving your setting modifications.

- **Apply**

Click **Apply** to save your settings and leave the dialog open.

Edit

This page is identical with the [Edit](#) page of the [Component Properties](#) dialog.

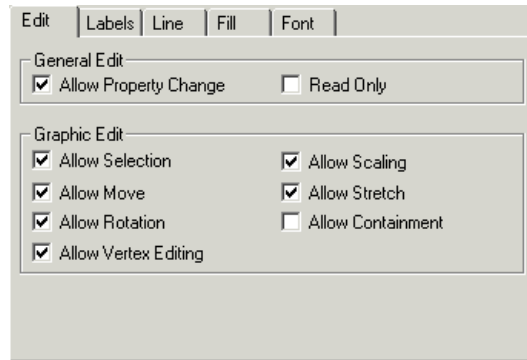


Figure 225. Default Properties Dialog: Edit Page

Fill

This page is identical with the [Fill](#) page of the [Component Properties](#) dialog.

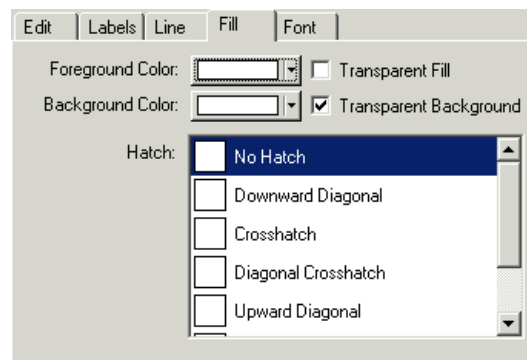


Figure 226. Default Properties Dialog: Fill Page

Font

This page is identical with the [Font](#) page of the [Component Properties](#) dialog.

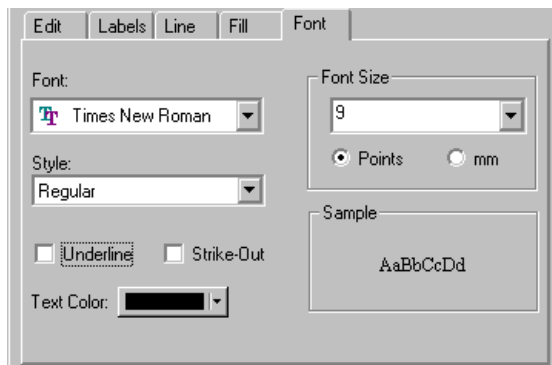


Figure 227. Default Properties Dialog: Font Page



If the font size is changed in **Default Properties**, the new font size is applicable till the Function Diagram is open. Once the Function Diagram is closed and reopened, the font size retains the default font, that is seven (7) points.



In the **Default Properties** to save a font size of a desired choice, e.g. ten (10) points, make changes in the other properties in the Function Diagram such as Edit, Fill etc. and save. Close the Function Diagram and reopen it. The saved changes are reflected and are used automatically by Function Designer while creating a component.

Labels

This page is identical with the [Labels](#) page of the [Component Properties](#) dialog.

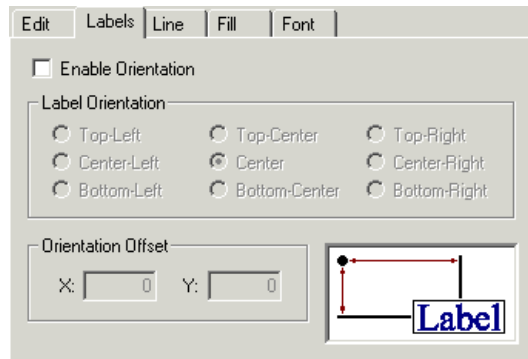


Figure 228. Default Properties Dialog: Labels Page

Line

This page is identical with the [Line](#) page of the [Component Properties](#) dialog.

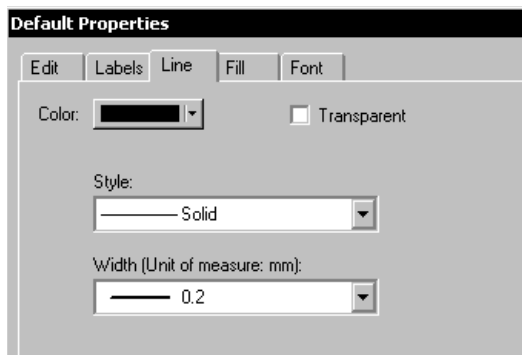


Figure 229. Default Properties Dialog: Line Page

Layers

The designer, Diagram view, allows you to design your diagram distributed on in several layers:

- Logic Layer
- Master Page Layer
- Background Layer
- Annotation Layer

For each layer, you can set properties defining the display of the layer in the diagram, if the layer is active or inactive. Use the [Layer Properties](#) dialog to set the properties for each layer.

Layer Properties

The **Layer Properties** dialog allows you to set the properties for each layer of the designer.

Access this dialog via the **View > Layers > Properties** menu command.

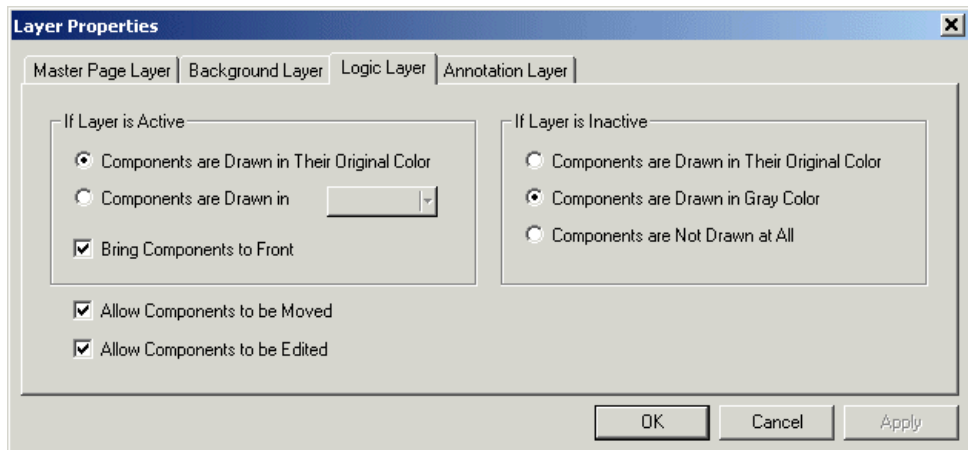


Figure 230. Layer Properties Dialog

- **OK**
Click **OK** to save your settings and to close the dialog.
- **Cancel**
Click **Cancel** to quit the dialog without saving your setting modifications.
- **Apply**
Click **Apply** to save your settings and leave the dialog open.
- **Master Page Layer**
Click this tab to access properties for the **Master Page Layer**.
- **Background Layer**
Click this tab to access properties for the **Background Layer**.
- **Logic Layer**
Click this tab to access properties for the **Logic Layer**.
- **Annotation Layer**
Click this tab to access properties for the **Annotation Layer**.
- **If Layer is Active**
Groups the properties defining the display of the active layer.
- **Components are drawn in their original color**
Check this option in group **If Layer is Active** to draw the layer's components in their original color - if the layer is active.

Check this option in group **If Layer is Inactive** to draw the layer's components in their original color - if the layer is inactive.
- **Components are drawn in**
Check this option to draw the layer's components in a user-defined color - if the layer is active.

Select a user-defined color from the color drop-down list box. For more colors, click the **Color** button to open the [Color](#) dialog.

- **Bring Components to Front**

Check this check box to bring components of the active layer to front.

- **If Layer is inactive**

Groups the properties defining the layer's display - if it is inactive, that is another layer is active.

- **Components are drawn in grey color**

Check this option to draw the layer's components in grey color - if the layer is inactive.

- **Components are Not Drawn at All**

Check this option to prevent drawing of the layer's components - if the layer is inactive.

- **Allow Components to be Moved**

Check this option to allow moving components on the layer.

- **Allow Components to be Edited**

Check this option to allow editing components on the layer. If unchecked, the layer is read-only.

Pages

In Diagram view, a diagram consists of several drawing pages of fixed page size. If a page is full, a new page can be inserted. Then, connections between components on different pages are automatically referenced via off-page-connectors. Size and orientation of a drawing page are defined via [Edit > Measurements and Size](#) dialog, whereas size and orientation of the printer page is defined via [Page Setup](#) dialog.

In Component view, a component or component template definition consists of one page only.

Keep in mind:

- The page you see on the screen is called the **drawing page**.
- You print the drawing on the **printed page**, which is the paper in the printer.

Typically the size and orientation of the printer page equals the drawing page. However, they can differ in the following cases:

- Your designer diagram is drawn on a drawing page A4 portrait (21.0 cm * 29.7 cm), but you print it on A3 landscape (42.0 cm * 29.7 cm) or Legal (8.5 cm * 11 inch) paper. If the printer paper size is less than the drawing page size, the drawing gets clipped.
- Your function diagram is drawn on a drawing page A3 landscape (42.0 cm * 29.7 cm), but you print it on A4 landscape (29.7 cm * 21 cm). You can zoom each page of the diagram to fit to the printer page size, either by a percentage value or on a Fit to number of printer sheets basis, see [Edit > Measurements and Size](#).
- Your function diagram is scaled, e.g. uses a predefined or user-defined drawing scale like 1 : 2 or 5 : 1. In that case, the drawing units differ from the page units. Thus you can print a A3 drawing also on A4 paper. Or you can insert more function component symbols on a page. Assume your symbols are drawn in 1 : 1 scale. Using 1 : 2 drawing scale, you can insert four times the number of symbols on a single diagram page. The drawing units are defined together with the drawing page size by [Edit > Measurements and Size](#).

In any case you can check the print results before actually printing via [File > Print Preview](#). It is recommended to check menu command [View > Printable Area](#) to

outline the printable area in print preview, i.e. the printer paper size reduced by print margins.



Blocks and reference in Function Diagram may exceed boundary on performing copy/paste of Function Diagram across the pages. To avoid this user has to manually move blocks and reference inside the boundary.

Measurements and Size

The **Measurements and Size** dialog allows you to change the units of measure (drawing units), the drawing scale, and the drawing page size.

The dialog is accessible via the **Edit > Measurements and Size** menu command.

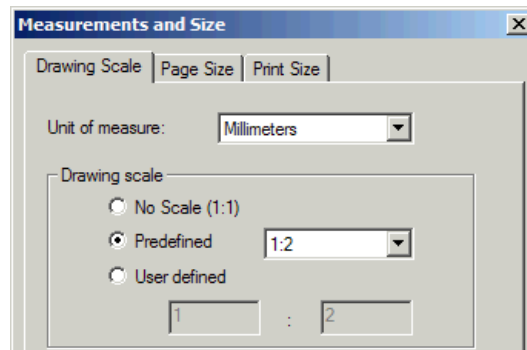


Figure 231. Measurement and Size Dialog

The dialog contains the following pages and buttons accessible from all pages:

- **Drawing Scale**
Click this tab to get access to the properties of the [Drawing Scale](#) page.
- **Page Size**
Click this tab to get access to the properties of the [Page Size](#) page.
- **Print Size**
Click this tab to get access to the properties of the [Print Size](#) page.

- **OK**

Click **OK** to save your settings and to close the dialog.

- **Cancel**

Click **Cancel** to quit the dialog without saving your setting modifications.

Drawing Scale

The **Drawing Scale** page allows you to specify what a given real-world size on the drawing represents. It also influences how measurements are presented to you.

Suppose you defined symbols (object types) with default scale 1 : 1. Lets say, the symbol's size equals 2 cm * 8 cm in real-world coordinates. Instantiate the symbol on a diagram...

- with drawing scale 1 : 1. The drawing on the screen, the rulers, and the symbols component properties, position and size, will display a symbol of size 2 cm * 8 cm. Print the diagram.
On the printer you will get a symbol of size 2 cm * 8 cm.
- with drawing scale 1 : 2. The drawing on the screen, the rulers, and the symbols component properties, position and size, will still display a symbol of size 2 cm * 8 cm. Print the diagram.
On the printer you will get a symbol of size 1 cm * 4 cm.
- with drawing scale 2 : 1. The drawing on the screen, the rulers, and the symbols component properties, position and size, will still display a symbol of size 2 cm * 8 cm. Print the diagram.
On the printer you will get a symbol of size 4 cm * 16 cm.

The default drawing scale is 1 : 1.

- **Unit of Measure**

Select the value of the drawing units from the drop-down list box. If you change the value, no measurement is changed at all. The chosen value just influences the measurement unit(s) displayed in

- Rulers
- **View > Grid Properties**: Grid spacing.
- **Edit > Default Properties**: Position and size, Font Size, Line Width, Label Orientation Offset.
- **Edit > Ambient Properties**: Off-Page Connector and Snap port distance.
- **Edit > Options** > Autorouting

- **Drawing Scale**

Select No Scale (1 : 1), a predefined drawing scale, or define a drawing scale by yourself.

- **Module**

Defines a base unit for font size increments, nudge by keyboard, and pan by keyboard. It is recommended to always set the following measurements in multiples of the module (default 1 mm):

- font size, e.g. 1mm, 2mm, 3mm...
- 10 * line width, e.g. 0.1mm, 0.2mm, 0.3mm...
- grid spacing, e.g. 1mm, 2mm, 3mm...
- auto routing settings, e.g. 1mm, 2mm, 3mm...

Page Size

On the **Page Size** page, you define the size of the drawing page.

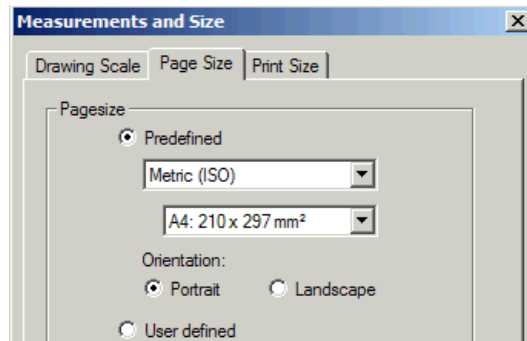


Figure 232. Measurement and Size Dialog: Page Size Page

- **Predefined**

The predefined drawing page sizes offered are filtered by the Metric (ISO) and English (US) setting. Select Metric (ISO) or English (US) and a drawing page size from the drop-down list.

If you choose Metric (ISO), all internal coordinates are based on 0.01 mm. If you choose English (US), all internal coordinates are based in 0.001 inch.

- **Orientation**

Select portrait or landscape.

- **User Defined**

Enter any size of the drawing page in the selected Units of Measure.

Print Size

Use this **Print Size** page if the diagram drawing page size does not fit the printer page size. E.g. your diagram page size equals A3 landscape, and you want to print on A4 landscape.

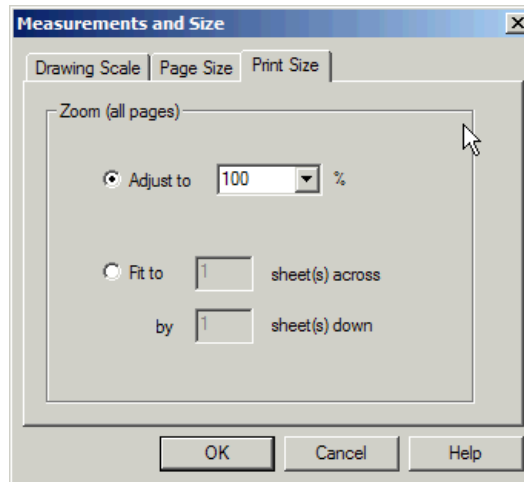


Figure 233. Measurement and Size Dialog: Print Size Page

- **Adjust to**

Reduces or enlarges a drawing page to a specified percentage. For example, to reduce a A3 drawing to A4, type 70.

- **Fit to**

Fits each drawing page on the number of sheets you specify across and down. Depending on the number of sheets you specify, the drawing is enlarged or reduced.

If the pages across and down are not proportional to the drawing's dimensions, only those sheets needed to maintain the drawing's proportions are used.

Fit to 1 sheet(s) across and 1 sheet(s) down guarantees that each diagram page fits on one sheet.

Page Setup

(Printer) page setup determine how the diagram / component is printed. The standard Windows **Page Setup** dialog allows you to select paper size, page orientation, margins for a specific printer. These settings are saved with your diagram, and the default printer is used to initialize the page setup for a new diagram.

Access the dialog via the **File > Page Setup** menu command to:

- Choose the **Paper Size** (A4, and so on) from drop-down list.
- Choose the **Paper Source** from the drop-down list.
- Choose either the **Portrait** or **Landscape** orientation.
- Choose the **Left, Right, Top, and Bottom Margins**.

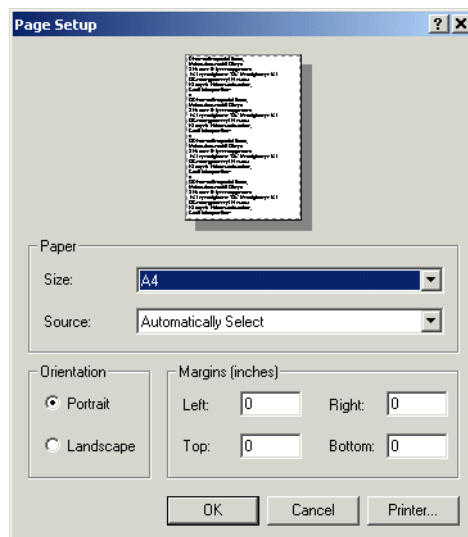


Figure 234. Page Setup Dialog

Select Master Page Template

The **Select Master Page Template** dialog lists predefined templates for diagrams. In effect, the selected master page template is referenced or copied into the active

diagram. The dialog is displayed when creating a new Function Diagram, see also *System 800xA Engineering, Engineering Studio Function Designer Getting Started (3BDS100968*)*, or by invoking the **File > Template** command.

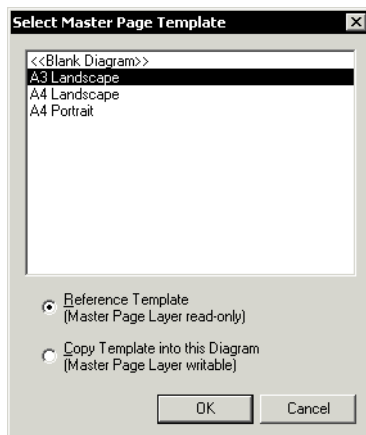


Figure 235. Select Master Page Template dialog

- **Template List**

Select a predefined diagram template. You can add additional templates or customize installed ones in Object Type Structure under Object Types\Functional Planning\Diagram Templates, see also [Figure 235](#).

- **Reference Template**

Select this option to reference the selected template in the active diagram. The diagram's master page layer gets read-only, but automatically updated when the template changes. Header/footer entries like author, document number, cannot be edited inside the diagram, but are referenced as aspect properties from a Function Diagram Document aspect of type Document.

- **Copy Template**

Select this option to copy the selected template into the active diagram. The diagram's master page layer gets writable, but not updated when the template changes. Header/footer entries like author, document number, can be edited

inside the diagram, but also referenced as aspect properties from a Function Diagram Document aspect of type Document.

Templates and Skeletons



The Templates and Skeletons dialog of Topology Designer corresponds to the Select Master Page Template dialog of Function Designer.

The **Templates and Skeletons** dialog allows to access predefined templates and skeletons for diagrams. In effect, the selected master page template is referenced or copied or the selected skeleton is copied into the active diagram. The dialog is displayed when creating a new Topology Diagram, or by invoking the **File > Template** command.

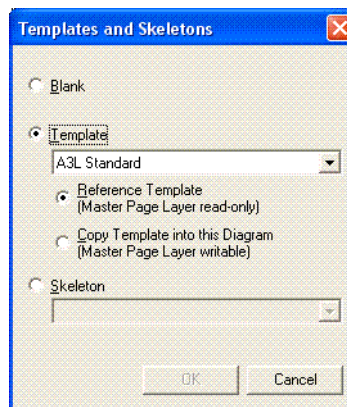


Figure 236. Topology Designer: Templates and Skeletons Dialog

You can add additional templates or customize installed ones in Object Type Structure under Object Types\Topology Planning\Diagram Templates.

You can add additional skeletons or customize installed ones in Object Type Structure under Object Types\Topology Planning\Topology Settings\Skeletons. See also Name.Description of these object type groups.

- **Reference Template**

Select this option to reference the template selected in the Template combo box in the active diagram. The diagram's master page layer gets read-only, but

automatically updated when the template changes. Header/footer entries like author, document number, cannot be edited inside the diagram, but are referenced as aspect properties from a Topology Diagram Document aspect of type Document.

- **Copy Template**

Select this option to copy the template selected in the Template combo box into the active diagram. The diagram's master page layer gets writable, but not updated when the template changes. Header/footer entries like author, document number, can be edited inside the diagram, but also referenced as aspect properties from a Topology Diagram Document aspect of type Document.


- **Skeleton**

Select this option to copy the skeleton selected in the Skeleton combo box into the active diagram. Skeletons behave like copied templates but contain additional prepared contents.

Insert/Delete Page

Insert Page

To insert a page in a designer diagram, choose one of the following methods:

- Click the **Page** toolbar button .
- Choose the **Insert > Page** menu command.
- Choose the **View > Pages > Insert New Page** menu command.
- Optionally enter a page comment, see [Figure 237](#).

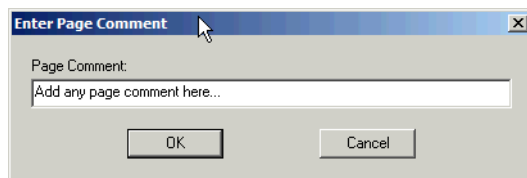


Figure 237. Dialog Enter Page Comment




For performance reasons a Function Diagram should not have more than 20 pages. A corresponding warning message box is displayed if you try to insert more pages.



In Diagram view, current page number is displayed in the first pane of the status bar and in the page toolbar.

Delete Page

To delete a page from a designer diagram, select the page to be deleted, and choose one of the following methods:

- Click the **Page** toolbar button  .
- Choose the **View > Pages > Delete Page** menu command

This function is enabled only if the according page is empty.

Page Comments

You can enter a page comment on **Insert Page** in the Enter Page Comment dialog, see [Figure 237](#). In addition, the Page comment command  from the **Page** toolbar or menu command **View > Pages > Page Comment** support

- modify/delete page comment
- copy page comment to different pages.

See [Figure 238](#).

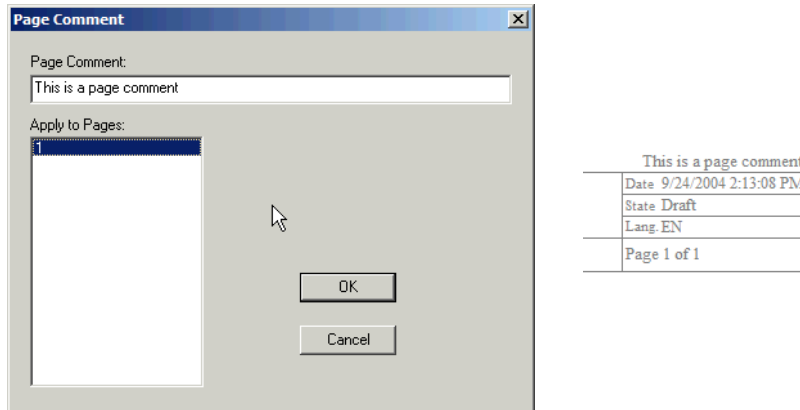


Figure 238. Page Comment Dialog

The page comment gets displayed

- on the lower-right corner of the page template, see [Figure 238](#).
- in the complete (detailed) contents diagram, created according to [Figure 240](#).


Page Navigation

To navigate to the wanted page, you have the following possibilities:

- [Navigation to the first page](#)
- [Navigation to the last page](#)
- [Navigation to the next page](#)
- [Navigation to the previous page](#)
- [Navigation to a certain page](#)

Navigation to the first page

To navigate to the first page, do one of the following:

- Click  of the Page toolbar
- Choose the **View > Pages > First Page** menu command.

Navigation to the last page

To navigate to the last page, do one of the following:

- Click  of the Page toolbar
- Choose the **View > Pages > Last** menu command.


Navigation to the next page

To navigate to the last page, do one of the following:

- Click  of the Page toolbar
- Choose the **View > Pages > Next** menu command.


Navigation to the previous page

To navigate to the last page, do one of the following:

- Click  of the Page toolbar
- Choose the **View > Pages > Previous** menu command.

Navigation to a certain page

To navigate to the last page, do one of the following:

- Select page number from combo box  in the Page toolbar.

- Choose the **View > Pages > Goto Page** menu command, enter the wanted page number (see [Figure 340](#)) and click **OK**.

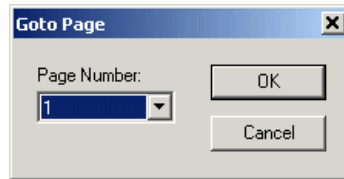


Figure 239. Goto Page Dialog

- Click directly in the wanted page if it is displayed in the diagram.

Printing

User can bulk print and create a contents diagram for each substructure by creating a dummy designer aspect at any level in designer structure. In order print and create a contents chapter for all diagrams in designer structure,

- create an empty designer aspect at the root object in designer structure
- open the (empty) Diagram view
- invoke the **File > Create Contents** command
- invoke the **File > Print** command.

The diagrams are listed and printed according to their level in alphabetical order.

Contents

User can create a contents diagram for

- a single diagram in Diagram view
- multiple diagrams in Diagram view (bulk printing).

User invoke creation or update of a contents diagram by the **File > Create Contents** command, see [Figure 240](#), and you open an already existing contents diagram by the **File > Open Contents** command.

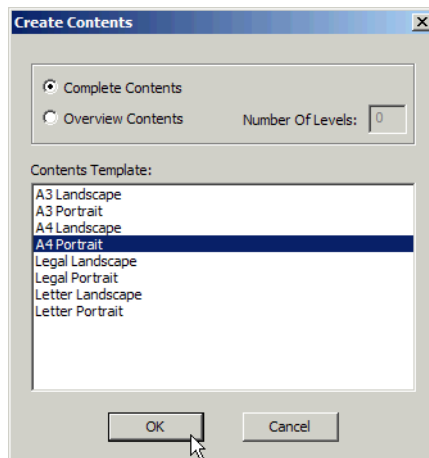


Figure 240. Create Contents Dialog

- **Complete Contents**

Select Complete contents to create detailed contents for each diagram and all its pages including functional designation (if any), name, description, page comment and page number.

- **Overview Contents**

Select Overview contents to create contents for all diagrams according to the selected number of levels. The overview includes functional designation, name, description, and number of pages. Child diagrams are indented according to their level in designer structure

- **Contents Template**

Select a contents template for the contents diagram. The contents diagram page size must not necessarily fit the diagram size. E.g. you can print the contents on A4 portrait, while your diagrams are printed on A4 landscape paper.

Print

User can print

- a single component in Component view
- a single diagram in Diagram view
- multiple diagrams in Diagram view (bulk printing).

User invoke printing by the **File > Print** command or by the corresponding toolbar button. In case there exist a contents diagram and/or sub- (child-) diagrams in designer structure, the dialog [Figure 241](#) gets displayed. Select the required options for the bulk print job.



It is recommended to split the bulk printing into smaller functional groups instead of printing as a single job. This helps to optimize available memory and avoids aborting of printing due to insufficient memory.

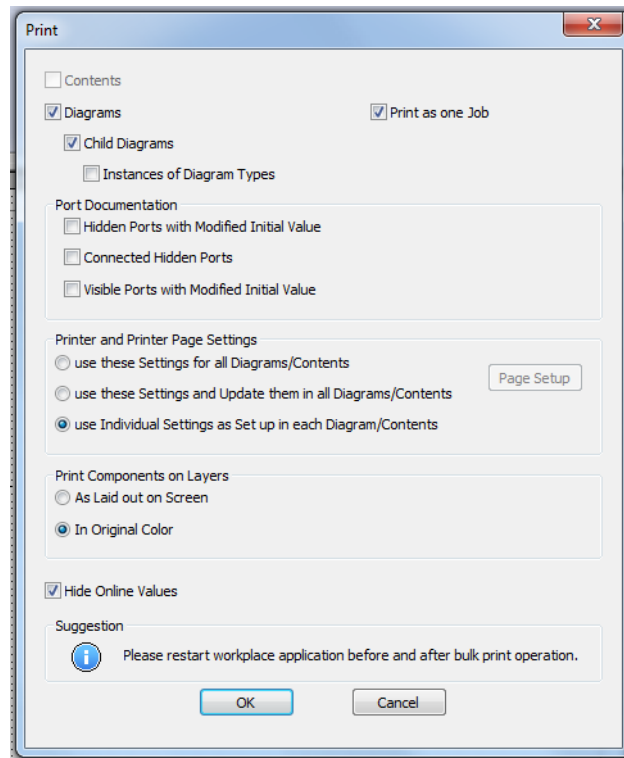


Figure 241. Bulk Print Dialog

- **Contents**
Check in order to print a contents diagram.
- **Diagrams**
Check in order to print diagram(s).
- **Child Diagrams**
Check in order to print child diagram(s).
- **Instances of Diagram Types**
Check in order to print instances of diagram types with their instance-specific appearance.

- **Print as one job**

Check in order to print everything as a single job.

- **Port Documentation**

If at least one of the check boxes is checked an additional document is generated with the corresponding information.

- Hidden Ports with Modified Initial Value
- Connected Hidden Ports
- Visible Ports with Modified Initial Value

- **Printer and Printer Page Settings**

Select an option

- to print the contents/diagrams with the same printer and printer page settings defined by this starting diagram (starting parent Function aspect). You can check and modify these settings by pressing the **Page Setup** button.
- to print and to update and save the contents/diagrams with the same printer and printer page settings defined by this starting diagram (starting parent designer aspect). You can check and modify these settings by pressing the **Page Setup** button.
- to print the contents/diagrams with individual printer and printer page settings, as setup and saved in each contents/diagram.

- **Print Components on Layers**

Select an option

- print the diagrams as laid out on the screen. In this case, components on different layers get grayed or not printed at all according to the [Layer Properties](#).
- print the diagrams in their original color.

- **Hide Online Values**

Select this option to print the diagrams without online values.

Click **OK**, the standard **Print** dialog [Figure 242](#) appears which allows the user to set further printer parameters:

- Select the **Printer Name** from a drop-down list.
- Access the printer-specific **Properties** dialog.
- Define the **Print range** by choosing one of the following options:
 - To print all pages, choose the **All** option.
 - To print a range of pages, define **from** which page **to** which page you want to print.
 - To print only the selected components, choose the **Selection** option.
- Define the **Number of copies** to be printed.

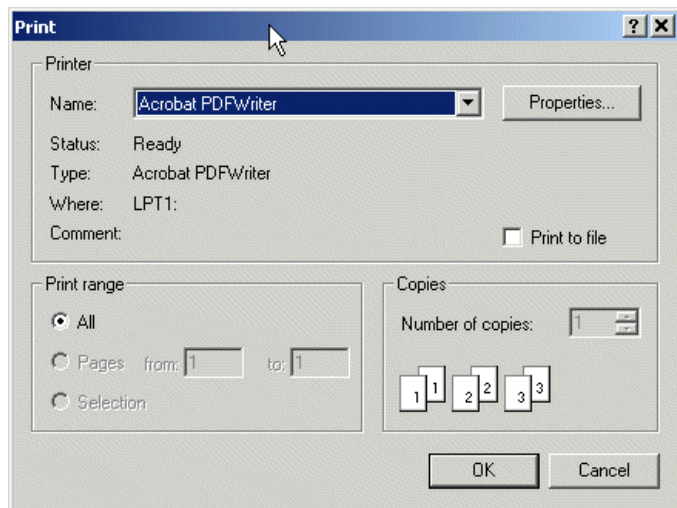


Figure 242. Print Dialog



The settings in [Figure 242](#) are valid for all diagrams printed in one job.

Click **OK**, a print working dialog indicating the current printed diagram appears. Click **Cancel** to cancel the print job.

The printed diagrams are printed according to their level in alphabetical order. They get listed in the output window of the starting parent designer aspect.



During the print operation of a Function Diagram, do not **minimize** the function diagram.

Print Preview

Use this preview to display the current diagram as it will be printed. Open the preview via the **File > Print Preview** menu command.

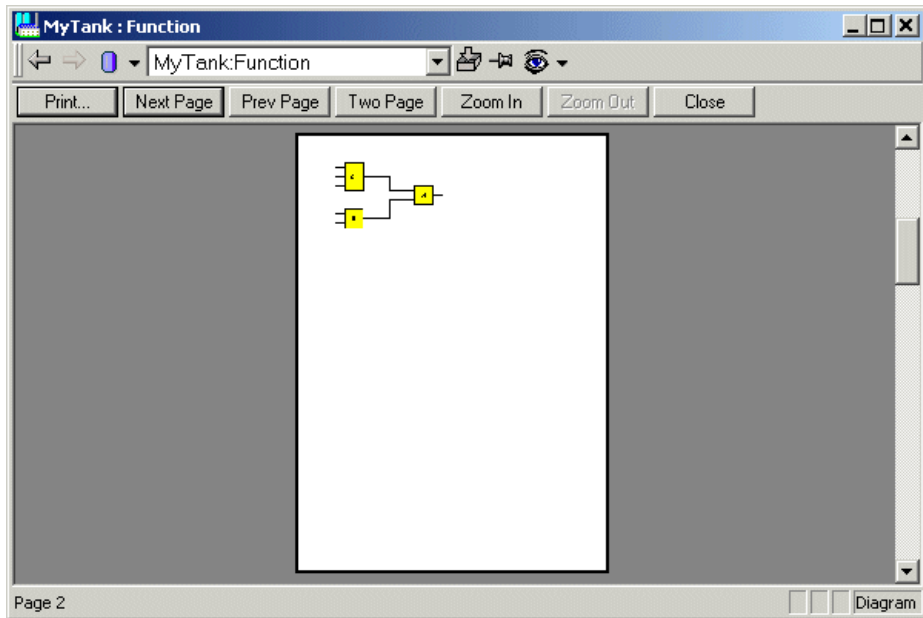


Figure 243. Print Preview

- **Print**

Click this button to open the [Print](#) dialog.

- **Next Page**

Click this button to navigate to the next page of the diagram.

- **Prev Page**
Click this button to navigate to the previous page of the diagram.
- **Two Page**
Click this button to display two pages side by side.
- **Zoom In**
Click this button to zoom in the selected area, that means to display this area enlarged.
- **Zoom Out**
Click this button to zoom out the selected area, that means to display this area reduced.
- **Close**
Click this button to close the preview.

Options

The **Options** dialog allows you to set various designer options

Access this dialog via the **Edit > Options** the menu command.

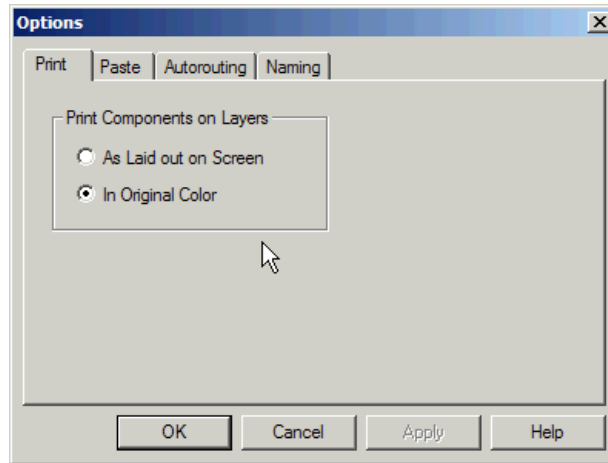


Figure 244. Options Dialog

The dialog contains the following pages and buttons accessible from all pages:

- **OK**
Click **OK** to save your settings and to close the dialog.
- **Cancel**
Click **Cancel** to quit the dialog without saving your setting modifications.
- **Apply**
Click **Apply** to save your settings and leave the dialog open.
- **Print**
Click this tab to access the options of the [Print](#) page.
- **Paste**
Click this tab to access the options of the [Paste](#) page.
- **Autorouting**
Click this tab to access the options of the [Autorouting](#) page.

- **Naming**

Note: Not available for Topology Designer.

Click this tab to access the options of the [Naming](#) page.

Print

The **Print** page of the Options dialog offers two options for printing components on layers:

- **As Laid out on Screen**

Check this option to print components as displayed on the screen. In that case layer properties [Figure 245](#) are taken into account, e.g. components on background layer are drawn greyed.

- **In Original Color**

Check this option to print all components on all layers in their original color.

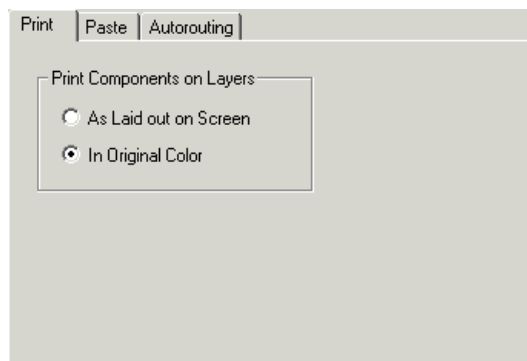


Figure 245. Options Dialog: Print Page

Paste

The **Paste** page of the Options dialog allows you to define the behavior on pasting copied components, see also [Paste Options](#):

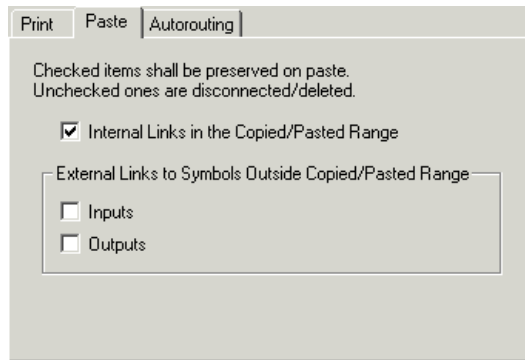


Figure 246. Options Dialog: Paste Page

- **Internal Links in the Copied/Pasted Range**

Check to paste links (connections) between copied/pasted symbols. By default, this setting is checked, see [Copying Designer Components including Internal Links](#), [Figure 197](#) and [Figure 198](#).

Uncheck to disconnect links (connections) between copied/pasted symbols.

- **Inputs**

Check to paste links (connections) between input ports of copied/pasted symbols and symbols outside the copied/pasted range (external inputs), see [Copying Designer Components including External Input Links](#), [Figure 199](#).

Uncheck to disconnect them.

- **Outputs**

Check to paste links (connections) between output ports of copied/pasted symbols and symbols outside the copied/pasted range (external outputs), see [Copying Designer Components including External Output Links](#), [Figure 200](#).

Uncheck to disconnect them.

Autorouting

Autorouting enables the capability of automatically routing all link components in a page or diagram. Link components get rerouted accordingly whenever:

- The linked symbols are moved.
- Show / Hidden state of the port(s) of an existing linked symbol is changed.

The **Autorouting** page of the Options dialog offers the possibility to set the following options:

- **Distance between Links**
- **Distance Link to Symbol**

For both options, you can set the horizontal and vertical value based on the unit defined in the [Measurements and Size](#) dialog.

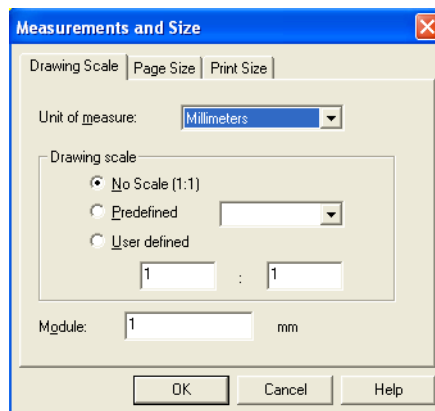


Figure 247. Options Dialog: Autorouting Page

Naming

Note: The Naming Page is not available for Topology Designer.

The **Naming** page of the Options dialog defines if the optional **New Component Name** prompt gets displayed on [Inserting Designer Components](#), see [Figure 249](#).

Check/uncheck the various component types in order to show/hide the **New Component Name** prompt. The settings get stored in the registry per user.

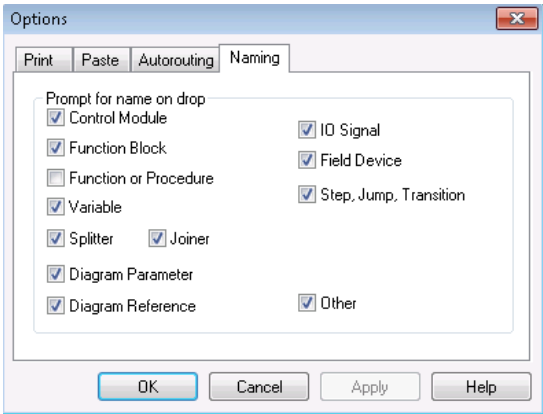


Figure 248. Options Dialog: Naming Page

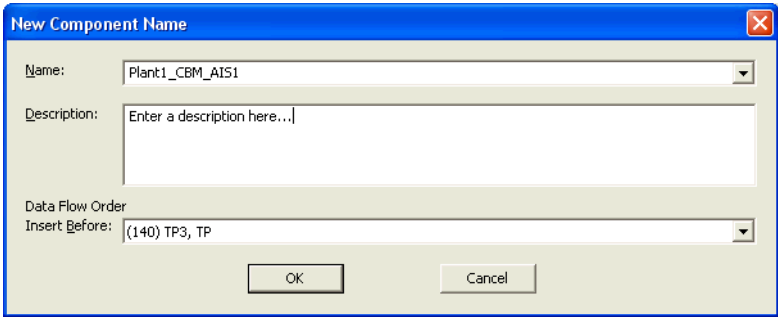


Figure 249. Dialog New Component Name

- **Name**
Accept or modify the default name proposed for the new instance. The default name itself can be configured in the Function Settings aspect. For aspect objects, the entered name equals the `..Name.Name` and the `..Function`

Parameters.ObjectName aspect properties. For symbol objects, the entered name equals `..Function Parameters.ObjectName`.



Function Designer supports displaying NLS (National Language Settings) variants for type name and instance name in Function Diagrams for blocks representing Aspect Objects (typically Function Blocks, Control Modules and CBM Signals).

For Aspect Objects the New Component Name dialog allows to insert a NLS variant of Name and Description according to the current Regional Settings of the Windows operating system.

For example, entering a name and a description under regional settings English (United States) results in a ENU NLS variant for Name and Description.

To add additional NLS variants of name and description use the Config View on the Name aspect of the object in Plant Explorer.

NLS variants are not supported for Symbol Objects.

- **Description**

Enter any description for the new instance. For aspect objects, the entered description equals the `..Name.Description` and the `..Function Parameters.ObjectDescription` aspect properties. For symbol objects, the entered name equals `..Function Parameters.ObjectDescription`.

- **Data Flow Order Insert Before**

The combo box **Data Flow Order Insert Before** lists all function components of the current allocatable group. Select the function component before which the new instance shall be located. Select **at the end of the Allocatable Group** if you want the new instance to be located at the end of the current allocatable group. If you want to change the data flow order of the instance later use the **Data Flow Order** dialog.

Import/Export

In addition to the Import/Export of .afw-files supported by the 800xA system, Function Designer supports export of diagrams and components into

- XML

- SVG (Scalable Vector Graphics)
- Bitmap
- Enhanced Metafile.

See [File Export](#).

and import of components from

- SVG (Scalable Vector Graphics).

See [File Import](#).

File Export

This dialog lets you export:

- The complete diagram with all pages and all layers as a set of XML and SVG files (*.xml, *.svg). The exported files contain graphics data only, but no other aspects nor aspect properties. To export the complete diagram with all its dependencies and substructure use System 800xA **Import Export** tool.
- The active layer as SVG file (*.svg). The exported SVG file contains graphics data only, but no other aspects nor aspect properties.
- The active page (canvas) as bitmap (*.bmp, *.dib)
- The active page (canvas) as enhanced metafile (*.emf)

This dialog appears when you choose the **File > SVG Export** menu command.

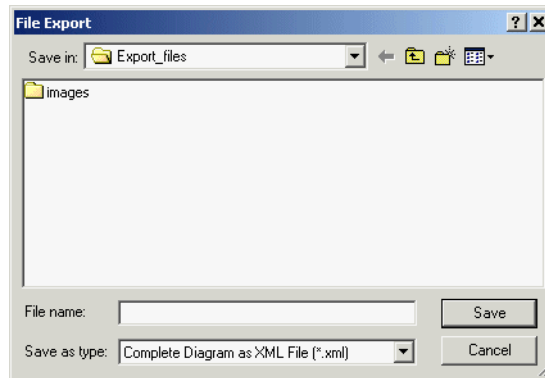


Figure 250. File Export Dialog

File Import

This dialog lets you import SVG files

- exported via the **File Export** dialog.
- exported via commercial graphic tools supporting SVG, e.g. Adobe Illustrator.

This dialog appears when you choose the **File > SVG Import** menu command.

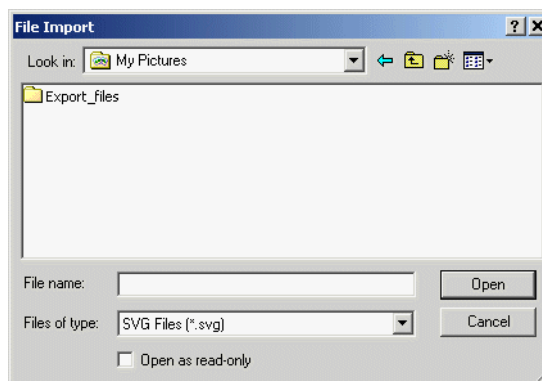


Figure 251. File Import Dialog

Export Diagram Data / Import Diagram Data

This feature enables the users to migrate the existing Single Control Modules (Function Diagrams created up to 800xA 5.0 SP2) to Diagrams.

Follow the steps to migrate the existing Single Control Modules to Diagrams.:

1. Launch the Plant Explorer.
2. Select the Functional Structure in the structure browser.
3. Select the required object in the Functional Structure and open the respective Function aspect.
4. Select **File > Export Diagram Data**.
5. Select a location and enter a file name in the **Save As** window.
6. Click **Save**. The diagram data is saved as a .xml file.
7. Create a blank Function Diagram. For more details refer to subsection Creating a new Function Diagram in *System 800xA Engineering, Engineering Studio Function Designer Getting Started (3BDS100968*)*.
8. Select **File > Import Diagram Data** in the created blank Function Diagram.
9. Select the exported .xml file in step 6 from the **Open File** window.
10. Click **Import**.

The exported diagram data is imported to the blank Function Diagram with all its Function Blocks, Control Modules, Signals, Variables, etc and their connected links.



Graphical position of Function Blocks, Control Modules, Signals, and Variables are maintained in the imported diagrams.

The diagram data of the multi page Function Diagrams are imported with the page details and interconnections.



After migrating the existing Single Control Modules to Diagrams, ensure the following before performing configuration data generation:

- a) Delete the existing connections between the Local Variables, CBM signals, and Diagram References and establish new connections.
- b) Explicit Splitter/Joiner blocks need to be used instead of variables.



To convert Function Diagrams having a parent and child structure:

1. Individually export the diagram data of the parent diagram.
2. Create a blank parent diagram similar to the exported Function Diagram structure.
3. Import the .xml files of the parent Function Diagram individually.
4. Individually export the diagram data of the child diagram.
5. Create a blank child diagram structure similar to the exported Function Diagram structure.
6. Import the .xml files of the child Function Diagram individually.



Newly created diagrams in system version 5.1 can also be exported and imported to the other newly created diagrams, using the above workflow. Ensure there are no missing connections within the imported diagrams.

Limitations of Export Diagram Data / Import Diagram Data

The following are not supported for import to a Function Diagram in the 800xA 5.1 system:

1. Diagram Parameters.
2. Global variables.
3. SFCs.
4. Text properties and text box properties.

Undo/Redo

With the [Undo](#) and [Redo](#) command you can restore the state of a diagram/component diagram before doing the last designer command.

It is also possible to undo/redo several commands step by step. The undo/redo stack is only limited by the computer's RAM.

Undo

Use the **Edit > Undo** menu command to undo the previous command. Alternatively, you can use the shortcut **CTRL+Z**. The following commands cannot be undone:

- **File > New**
- **File > Save, File > Save All, File > Skip Modifications**
- **File > Generate Configuration Data (Full Build)**
- **Allocation > Allocate Diagram**
- **Changing Type of Component Instances**
- **Changing Number of Inputs**
- Any modifications done outside Function Designer diagram view or component view. E.g. Plant Explorer create/delete/move/cut/paste aspect objects cannot be undone, nor Bulk Data Manager modifications.
- Any code-relevant changes modifying the state (e.g. generated or loaded) cannot be undone.

Redo

If the last Function Designer command is undone, user can restore the state to before undoing. Select **Edit > Redo** menu command or alternatively, the shortcut **CTRL+Y**.

Section 4 Error Messages

Function Designer checks the Channel/HSE HOST/Direction/Data Type attributes of the FF Signal. These attributes are mandatory for Function Designer to create the ports and generate the code. If any one of these attributes is missed during FF engineering, then FuD will prompt some error while allocating the FBAD to the FF Proxy object.

[Table 55](#) shows the list of error messages prompted by FuD while allocating the FBAD to the FF Proxy object with possible solution.

Table 55. Errors with solution

Serial No.	Error Description	Solution
1	If the "Channel" is not assigned for the signal in FF Builder, FuD says "Channel is not allocated for some signals" while allocating the FBAD to FF Proxy object.	Correct the FBAD Diagram and reallocate the FBAD.
2	If the "HSE HOST"/ "Direction" / "Data Type" is not assigned for the signal in FF Builder, FuD says "Channel information is not found for some signals" while allocating the FBAD to the FF Proxy object.	Correct the FBAD diagram and reallocate the FBAD.

Table 55. Errors with solution

Serial No.	Error Description	Solution
3	If the allocated FBAD do not have any FF signal, FuD says "Signal Information is not Found" while allocating the FBAD to FF Proxy object. This may happen if the FBAD diagram is not saved properly after modification.	Correct the FBAD and reallocate the FBAD to the FF Proxy object.
4	If the XML received from the FBAD is not proper, FuD says "XML Parsing failed".	Resave the FBAD diagram properly.
5	If any fault occurred during the processing of FBAD XML, FuD says "XML Error".	Check if all the signal names and their attributes are saved properly.
6	If the selected FBAD diagram do not have any XML data, FuD says "Failed to get the FF information".	Save the selected FBAD in FF builder properly.

Appendix A Restored Functionalities

This appendix provides information for specific Engineering Studio functionalities or features available in Function Diagrams restored to Engineering Studio 800xA 5.1.

Single Control Modules

Function Diagrams created upto 800xA 5.0 SP2 are available as Single Control Modules (SCM) in Control Builder M, as shown in [Figure 252](#).

Creation of an SCM based Function Diagram

Follow the steps to create Single Control Module (SCM) aspect based Function Diagram:

1. In the Plant Explorer, navigate to a parent object in the Functional Structure.
2. Right-click the parent object and click **New Object...** from the context menu to open the **New Object** dialog.
3. In the structure browser of this dialog, click **Generic Type**.
4. Enter a name in the **Name** field.
5. Enter a description in the **Object description** field.
6. Click **Create**.
7. Right-click this object and select **New Aspect...**
8. In the **Common** tab, select the **List presentation** check box.
9. Select the **Single Control Module** aspect.
10. Click **Create**.
11. In the **Common** tab, select the **List presentation** check box.
12. Select the **Function** aspect.

13. Click **Create**.
14. Double-click the Function aspect, and select **File > Template...** to assign the required template.
15. Allocate the Function Diagram to an application (see [Allocating Function Diagrams](#)) and use it as a SCM based diagram.

Alternatively, an existing SCM based Function Diagram can also be copy / pasted. This Function Diagram can be modified and used as a SCM diagram.

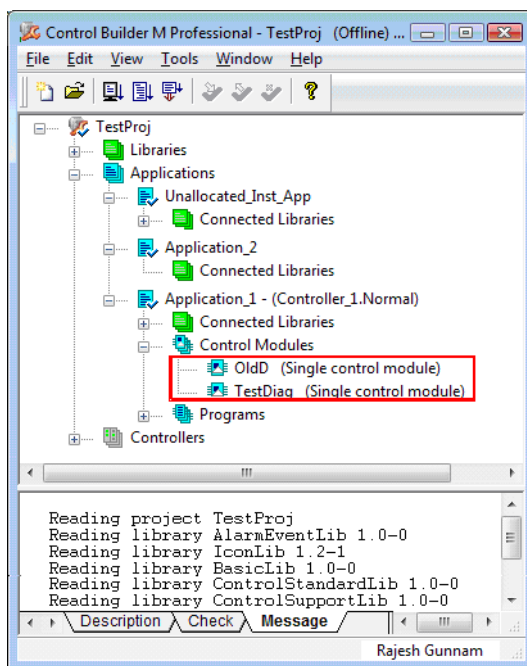


Figure 252. Function Diagram Available as Single Control Module



From 800xA 5.1 onwards the newly created Function Diagrams are available within the **Diagram** folder in the Control Builder and Control Structure as shown in [Figure 2](#).



Function Diagrams restored to system version 5.1, requires relatively more time to open for the first time.

Implicit Casts

Function Designer supports implicit casts of data types / connections. These include:

- Insertion of upcast function in CBM (e.g. DINT_TO_REAL)
- Insertion of downcast function in CBM (e.g. REAL_TO_DINT) with warning
- Conversion of real constants to CBM format (2 -> 2.0)
- Selection of proper element of structured data type (e.g. Value of RealIO)

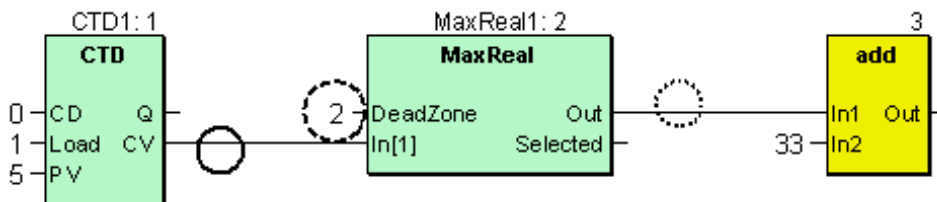


Figure 253. Example Function Diagram for Implicit Casts

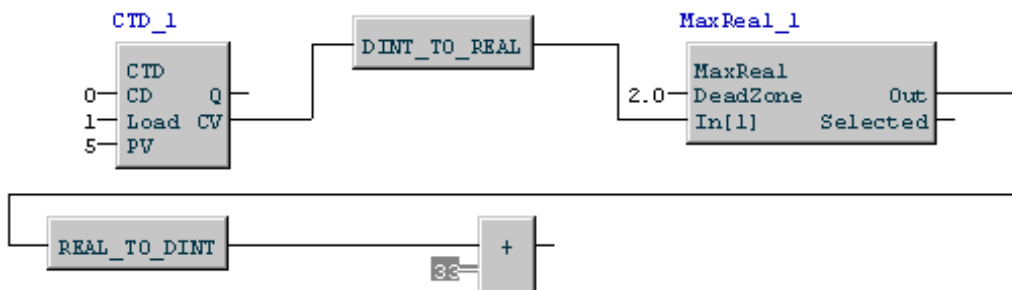


Figure 254. Generated CBM Code with Implicit Casts

Casted Data Types

Table 56. Support Implicit Casts

		Destination								
Source		BOOL	INT	DINT	UINT	WORD	DWORD	REAL	TIME	STRING
	BOOL		+	+	+	+	+	+	-	+
	INT	+w		+	+w	+w	+w	+	-	+
	DINT	+w	+w		+w	+w	+w	+	+	+
	UINT	+w	+w	+		+w	+w	+	-	+
	WORD	+w	+	+	+		+	-	-	+
	DWORD	+w	+w	+	+w	+w		-	-	+
	REAL	+w	+w	+w	+w	-	-		+	+
	TIME	-	-	+	-	-	-	+		+
	STRING	+x	+x	+x	+x	+x	+x	+x	+x	

Legend:

Same data type (no action needed)

- + Conversion inserted
- +w Conversion inserted, warning given (possible effects:
 - Truncation, change of sign)
- +x String with wrong format sets the destination value to 0 / False at runtime (see also CBM documentation)
- not supported

The used conversion functions of CBM are named
<Source Data Type>_TO_<Destination Data Type>
e.g. INT_TO_REAL, TIME_TO_STRING.

There is only one cast function used (i.e. no cascading of implicit casts).

A conversion function (e.g. INT_TO_REAL) always belongs to the input that needs a conversion, i.e. networks may need more than one conversion function. E.g. an integer output gets connected to three real inputs will use three INT_TO_REAL functions.

Ports with data type ANY (e.g. MOVE) are not checked.

Structured Data Types

If a basic data type (e.g. real) is connected to a structured one, the main element of the structure is connected. (e.g. '.Value' of RealIO); see also table below.

Table 57. Used Elements of Structured Data Types

Type Name	Usage	Connected element
RealIO	Connection to process IO	RealIO.Value
BoolIO	Connection to process IO	BoolIO.Value
ControlConnection	Two-way communication between control modules	ControlConnection.Forward.Value
HWStatus	IO Diagnostics	HWStatus.HwState
DwordIO	Connection to process IO	DwordIO.Value
DintIO	Connection to process IO	DintIO.Value

Constant Connections, Literals

This sub chapter only describes additions that are done by the code generator for CBM to produce valid and better readable code.

- Data Type REAL

- If the decimal separator is missing, '. 0' is added
(e.g. 4711 -> 4711.0)

- Data Type INT, DINT, UINT, WORD, DWORD

- Numbers to bases other than 10 are represented in base 2, 8 or 16 (prefix 2#, 8# or 16#)
- Data Type BOOL
 - Valid constants for BOOL are 0, 1, FALSE, TRUE
 - If written in lowercase, they are converted to upper case
- Data Type TIME
 - If the prefix T# or TIME# is missing, it is added (e.g. 39m1s -> T#39m1s)
- Data Type DATE_AND_TIME
 - If the prefix DT# is missing, it is added (e.g. 2003-12-31-21:35:07 -> DT#2003-12-31-21:35:07)

Checks for Connections

Certain combinations of unequal data types are accepted and lead to implicit casts (see below):

- Input ports and inout ports must not be connected more than once.
- Output ports and inout ports must not be connected to direct constants or variables with the flag 'constant'.
- Corresponding inout ports must not be connected to each other.



More than one connection to InOut Port: Create Variable object and connect Variable to InOutPort (right side); right side of Variable can be connected several times to other symbols.

Connection/Networks

Consider a **network** having a set of M:N connections between ports. A connection network has 1...M sources and 1...N sinks, and is defined by its unique network name, that equals the connect string of type variable. All network components, i.e. all links and off-page connectors, share the variable/network name as component name property. The variable/network name is also accessible by a label attached to network links. Automatically created links get the default variable/network names link, link1, link2...

Consider the following examples:

- In Figure 255, S4:Out was connected to variable abc. A $M : N$ network, $M = 2$, $N = 2$, gets automatically created.

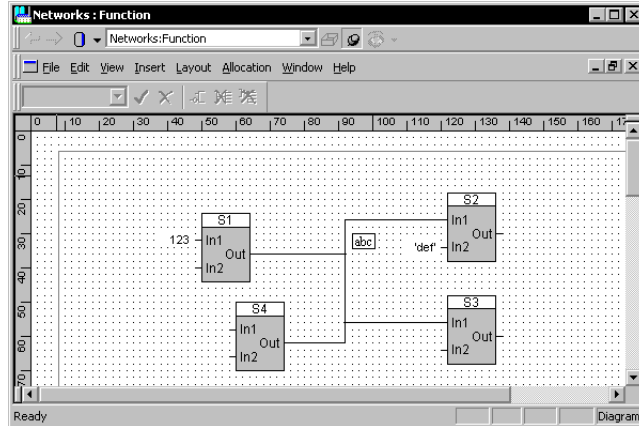


Figure 255. $M : N$ Network ($M = 2$, $N = 2$)

In Figure 256, S2 was moved from page 1 to page 2. Off-page connectors with page references get automatically created. Page referencing is not reflected in the aspect properties nor in Bulk Data Manager, i.e. S1:Out and S2:In still hold the connect string abc. In other words, off-page connectors are also automatically created when connecting symbols on different pages by updating aspect properties, e.g. in Bulk Data Manager.



Behavior of page references:

- Page references to symbols without name (functions) display the name of the connection instead.
- There may be multiple page references on a page to the same source or sink.

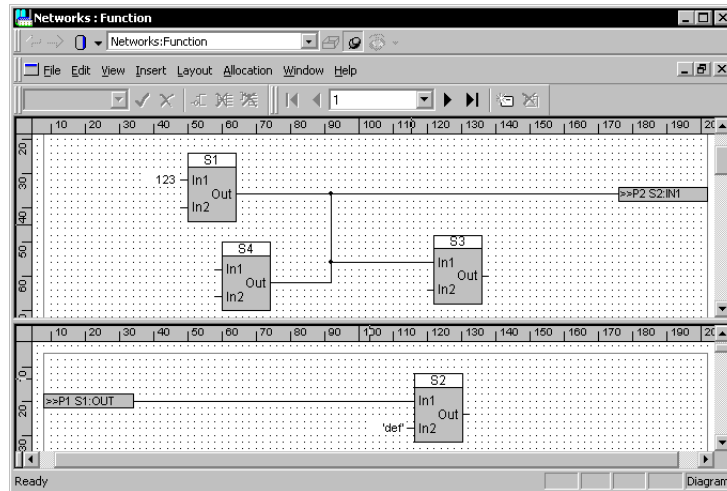


Figure 256. $M : N$ Network with Off-Page Connectors

Connection Mapping

Use the function **Connection Mapping** to connect ports of different structured data type. This function is implicitly invoked for connections that cannot be automatically cast. User can also explicitly invoke the Connection Mapping command from a link or a variable component. In both cases, user can select which substructures shall be interconnected. Necessary variables and links get automatically created.

Suppose you want to connect a port of data type 'TenRealIO' to a port of data type 'RealIO', see [Figure 257](#). 'TenRealIO' is a library-defined structured data type and consists of 10 record elements of type RealIO.

The connection checks display an error message indicating that there is no implicit conversion possible. Now you can either

- insert a connection mapping. In that case, the **Connection Mapping** dialog gets displayed where you define a mapping between individual sub-structures or record elements of the source data type and the sink data type. On OK, necessary variable(s) and links get automatically created. Use command Show/Hide ports in order to connect the rest of the IO.

- ignore the error. This will result in an erroneous connection link drawn in red color.
- cancel the connection.

In [Figure 258](#), two variables and corresponding links get created to map the data types 'TenRealIO' (structured data type with 10 record elements RealIO) and 'FiveRealIO' (structured data type with 5 record elements RealIO). The automatically created variables are named <SourceComponent>_<SourcePort> and <SinkComponent>_<SinkPort>.

In order to modify a connection mapping, select such a mapping variable and invoke its **Connection Mapping**.

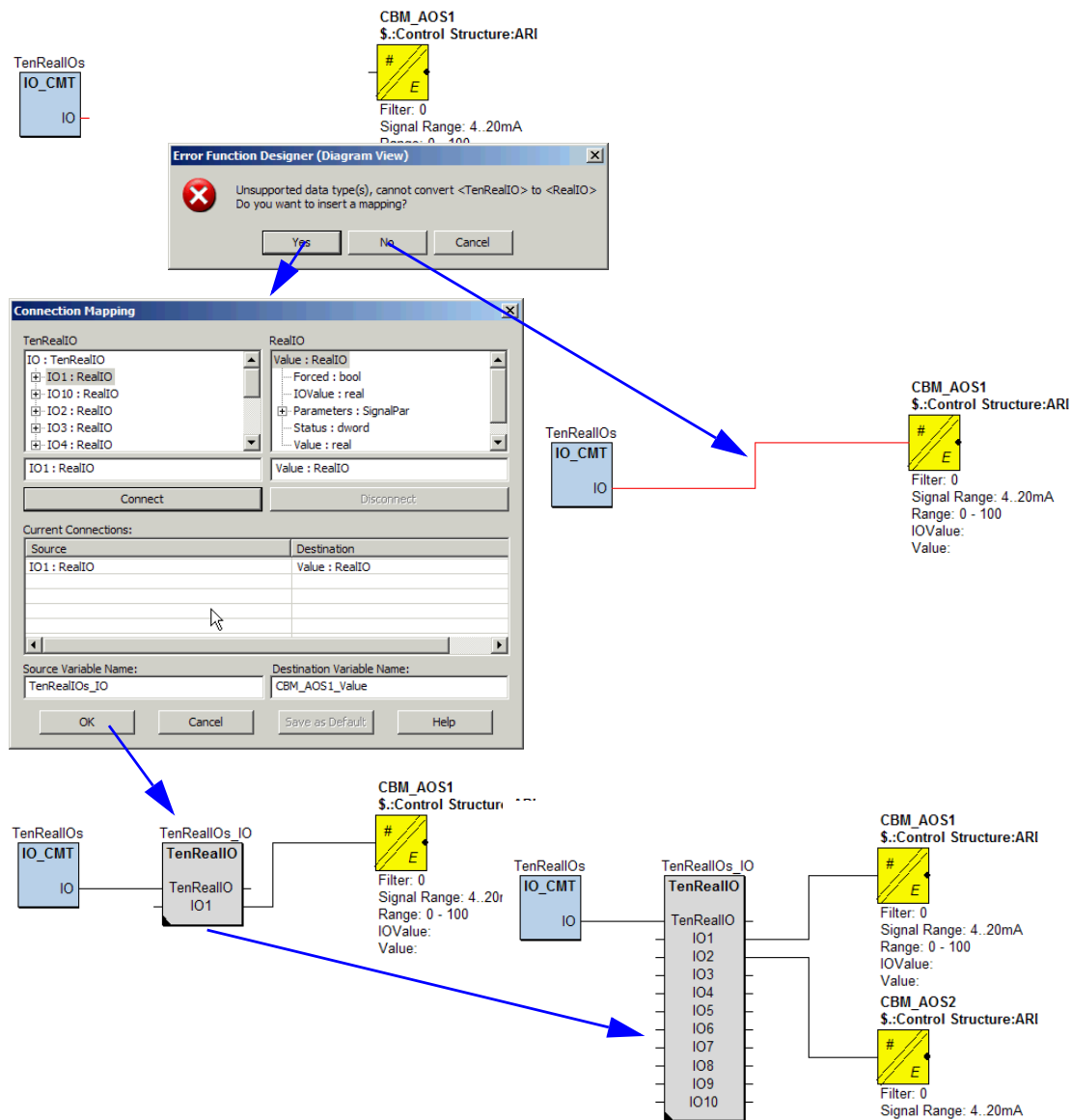


Figure 257. Connection Mapping TenRealIO to RealIO

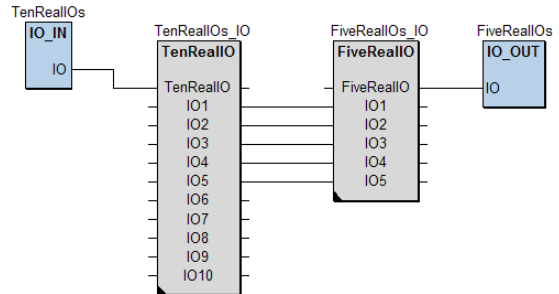
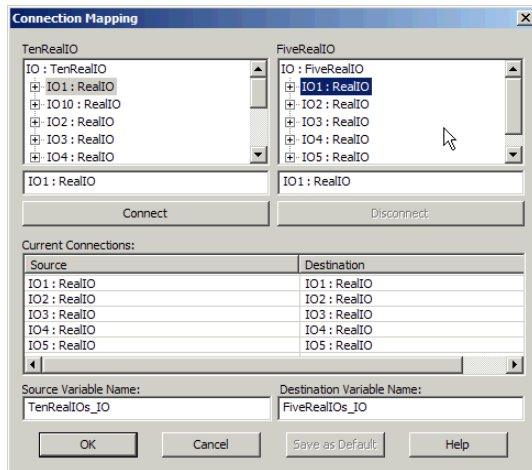


Figure 258. Connection Mapping TenRealIO to FiveRealIO

Code Relevant Changes

Code relevant changes are modifications that affect the configuration data that Function Designer generates. These changes are causing transitions between the states.

You can make these kind of changes in:

- Function Designer (FD)
- Control Builder M (CBM)
- Plant Explorer (PE)
- Bulk Data Manager (BDM)
- or any other environment that modifies the Function Designer's Aspect Objects).



Diagram State Modifications cannot be Undone. If you do code-relevant modifications ‘Generated’ or ‘Loaded’ Function Designer displays a warning in order to cancel the modification. If you ignore this warning, the diagram gets ‘Modified’ state. ‘Undo’ or ‘File > Skip Modifications’ will restore the diagram data, but not the diagram state. Regenerate configuration data and download the diagram.



If you connect an Application object to a new version of a user defined library containing data types with changed definition. The related function diagram with instances (variables) of these data types should become ‘red’.The diagram state remains ‘yellow’, however.

Table 58. Modifications of Function Designer Generated Code

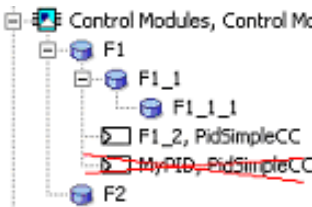

Case Id	Modification	Source	State Change	Description
AO1§	<p>You remove a Control Module or a Function Block that is a child of a Function Designer generated Single Control Module from Control Structure.</p> 	PE	<p>F1:</p> 	<p>If the object ‘MyPID’ has a parent object in the Functional Structure, this object will undergo the same state change as the parent object in the Control Structure.</p> <p>Any open Function Diagram is updated.</p> <p>Note: if the object that has been removed is a Control Module, this has no effect on the code blocks or variable definitions of the parent Single Control Module. However, this is not treated in a different way, because it is expected that the parent object in the Control Structure is also the parent object in the Functional Structure in most cases.</p>

Table 58. Modifications of Function Designer Generated Code (Continued)

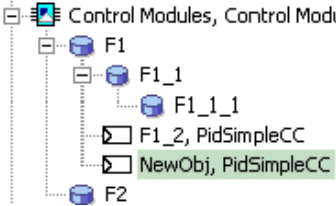

Case Id	Modification	Source	State Change	Description
AO2	<p>You insert a Control Module or a Function Block as a child of a Function Designer generated Single Control Module in the Control Structure.</p> 	PE	<p>F1:</p> 	<p>The parent propagates its own state change to all its child objects in the Control Structure. If the newly inserted object is not placed in the Functional Structure, the parent object in the Control Structure will also become the parent object in the Functional Structure.</p> <p>Any open Function Diagram is updated.</p> <p>Note: if the Function aspect is inherited, the object cannot be in state 'Modified'.</p> <p>If the object 'NewObj' has a parent object in the Functional Structure, this object will undergo the same state change as the parent object in the Control Structure, but the state change is not propagated to the child objects in the Functional Structure.</p>

Table 58. Modifications of Function Designer Generated Code (Continued)

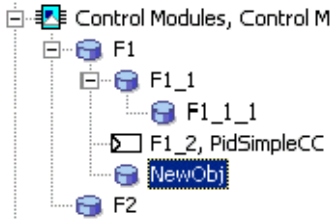


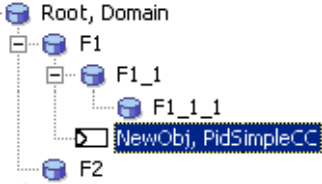

Case Id	Modification	Source	State Change	Description
AO3	<p>You insert an object with a Function aspect as a child of a Function Designer generated Single Control Module in the Control Structure. Note: 'NewObj' is a child diagram.</p> 	PE	<p>F1:</p>  <p>NewObj:</p> 	<p>The code generator automatically starts for object 'NewObj' during the insertion of the object into the Control Structure. If the code generator runs without error, 'NewObj' ends in state 'Generated'.</p> <p>Any open Function Diagram is updated.</p>
AO4	<p>You insert a Control Module or a Function Block as a child of a Function in the Functional Structure.</p> 	PE FD	<p>F1:</p> 	<p>Any open Function Diagram is updated.</p>

Table 58. Modifications of Function Designer Generated Code (Continued)

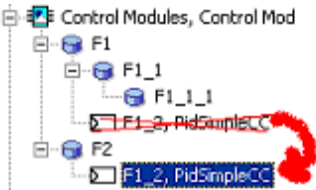
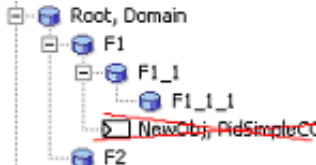



Case Id	Modification	Source	State Change	Description
AO5	<p>You move a Function Designer generated Control Module or a Function Block to a new location in the Control Structure.</p> 	PE	<p>F1: Like in AO1</p> <p>F2: Like in AO3</p>	Any open Function Diagram is updated.
AO6	<p>You remove a Function Block or a Control Module from a Function in the Functional Structure</p> 	PE FD	<p>F1:</p> 	Any open Function Diagram is updated.
AO7	<p>You rename a Function Designer generated Function Block or Control Module</p> 	PE FD	<p>F1:</p> 	<p>The object (F1_2) detects the modification and notifies its parent in Control Structure and the parent in Functional Structure.</p> <p>Any open Function Diagram is updated.</p>

Table 58. Modifications of Function Designer Generated Code (Continued)

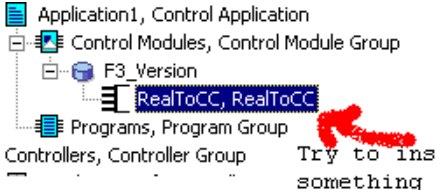
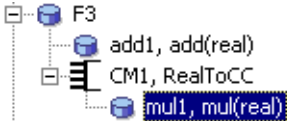

Case Id	Modification	Source	State Change	Description
AO8	Copy & Paste in Functional Structure	FD PE		See AO4 for the receiving parent diagram. The source object or its parent is not affected.
AO9	Copy & Paste in Control Structure	PE		See AO2 for the receiving diagram.
AO10	Cut & Paste in Control Structure	PE		See AO1 for the source object. See AO2 for the sink.
AO11	Cut & Paste in Functional Structure	FD PE		See AO6 for the source. See AO4 for the sink.
AO12	You insert a new Control Module or Function Block as a child to an existing Control Module in Control Structure 	PE	None	This action is rejected by CBM. It is not possible to add a new child object to an instance of a Control Module type. This would be a type change.
AO13	You insert a new Control Module or Function Block as a child to an existing Control Module in the Functional Structure 	PE	F3: 	The Function Diagram of 'F3' is updated with the new object. Any open Function Diagram is updated.

Table 58. Modifications of Function Designer Generated Code (Continued)

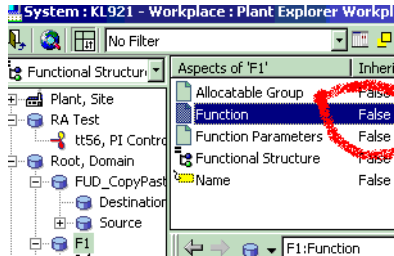



Case Id	Modification	Source	State Change	Description
AO14	You change the type of an object. (Supported only for symbol objects).	FD		The effected parent object in Functional Structure (diagram or allocatable group) change their state to 'Modified'.
AO15	You override an inherited Function aspect 	PE	F1: 	
CS1	You change the connect string of a Function Designer generated Control Module. 	PE FD	F1: 	<p>The object F1_2 detects the modification and notifies its parent in Control Structure and the parent in Functional Structure.</p> <p>If a diagram that shows the connection is open, the diagram gets updated.</p> <p>If you modify a connect string in Function Designer, the changes are written to the Control Properties of the effected Control Module immediately.</p>

Table 58. Modifications of Function Designer Generated Code (Continued)

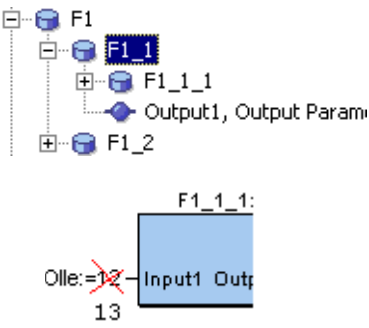

Case Id	Modification	Source	State Change	Description
CS2	<p>You modify the initial value of an internal variable of a Function Designer generated Single Control Module</p> 	PE FD	<p>F1_1:</p> 	If the modification in done using the Control Properties aspect, the change is reflected in Diagram.
CS3	<p>You modify the initial value of an internal variable of a Control Module that is part of a Function Diagram.</p>	PE		The containing Single Control Module (parent in the Control Structure) changes its state to 'modified'. If such a Single Control Module does not exist, the containing application changes its state to 'modified'.

Table 58. Modifications of Function Designer Generated Code (Continued)

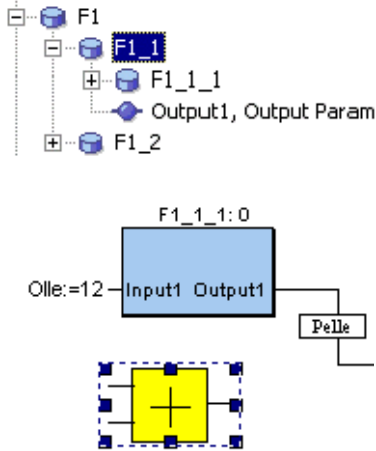

Case Id	Modification	Source	State Change	Description
SO1	<p>You add an IEC 61131-3 function (a none Aspect Object one) to a Function Diagram.</p> 	FD	<p>F1_1:</p> 	<p>The state of the diagram that contains the new IEC 61131-3 function is changed to 'modified'.</p>

Table 58. Modifications of Function Designer Generated Code (Continued)

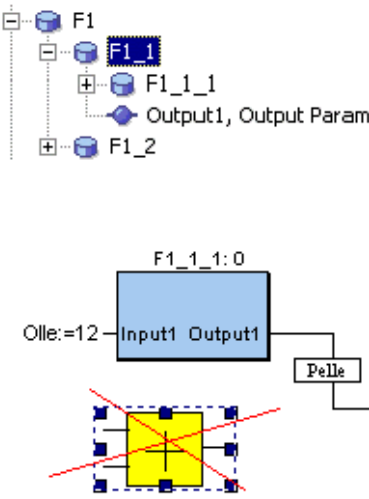

Case Id	Modification	Source	State Change	Description
SO2	<p>You remove a Function (a none Aspect Object one) from a Function Diagram</p> 	FD	<p>F1:</p> 	<p>The state of the diagram that contains the new IEC 61131-3 function is changed to 'modified'.</p>

Table 58. Modifications of Function Designer Generated Code (Continued)

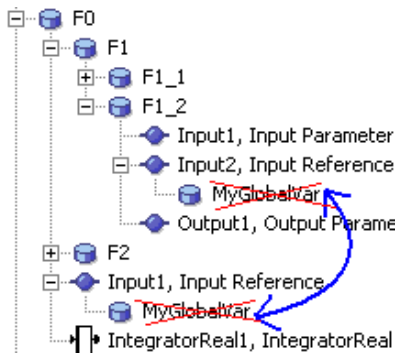

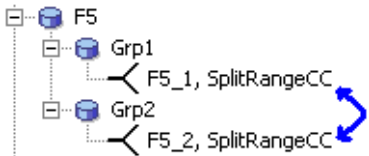
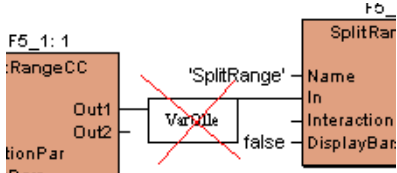

Case Id	Modification	Source	State Change	Description
GV1	<p>You delete a global variable that has been generated by the Function Designer.</p> 	<p>CBM FD PE</p>	<p>F1_2, F2:</p> 	<p>All Functions reading or writing the effected global variable are set to the state 'Modified'.</p>
GV2	 	<p>CBM FD</p>	<p>F5:</p> 	<p>The global variable 'VarOle' is generated by Function Designer to connect the two Control Modules. The two Control Modules are defined in two different allocatable groups.</p> <p>If the modification is done in the Control Builder, the change is not reflected in the diagram. The variable is re-created the next time we generate configuration data.</p>

Table 58. Modifications of Function Designer Generated Code (Continued)

Case Id	Modification	Source	State Change	Description
GV3	You rename a global variable that has been generated by the Function Designer.	CBM FD		See GV1
GV4	You change the initial value of a global variable.	CBM FD PE		All Functions using the effected global variable are set to the state 'Modified'. For any open Function Diagram the display is updated.
GV5	You change the attributes of a global variable.	CBM FD		Changes done in CBM are synchronized to attributes stored in diagram variables and diagram references. The state of effected diagrams is <u>not</u> changed. Changes done directly in diagram variables or diagram references set all effected diagrams to state 'Modified'.
GV7	You add a diagram reference 	FD PE	F5: 	Any open Function Diagram is updated.

Table 58. Modifications of Function Designer Generated Code (Continued)



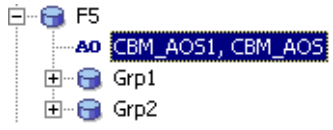

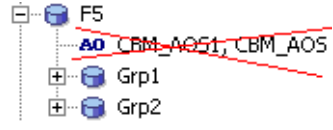

Case Id	Modification	Source	State Change	Description
GV8	<p>You connect a diagram variable to a diagram reference.</p> 	FD PE	<p>F5:</p> 	Any open Function Diagram is updated.
IO1	<p>You add a new IO signal as a child of an object with a Function in the Functional Structure.</p> 	PE FD	<p>F5:</p> 	Any open Function Diagram is updated.
IO2	<p>You remove an IO signal from a Function.</p> 	PE FD	<p>F5:</p> 	Any open Function Diagram is updated.
IO3	You rename an IO signal	PE FD		See AO7
IO4	You change the allocation of an IO signal.	PE		Any open Function Diagram is updated.

Table 58. Modifications of Function Designer Generated Code (Continued)

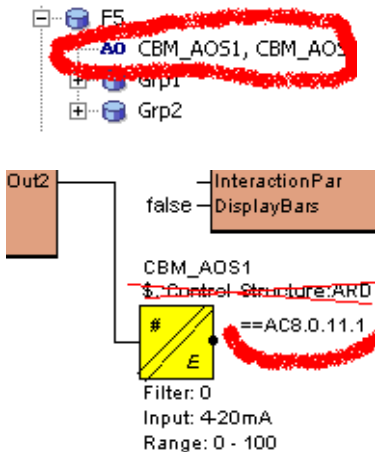

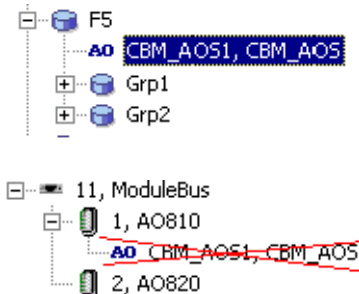

Case Id	Modification	Source	State Change	Description
IO5	<p>You move an unallocated IO signal to a Hardware Unit.</p> 	PE	F5: 	<p>The IO signal object detects the structure change. It notifies all its parent objects in the Functional Structure.</p> <p>You can now re-generate the configuration data. This will result in the correct assignment of the (global) variable to the IO channels.</p>
IO6	<p>You remove an allocated IO signal from its Hardware Unit.</p> 	PE	F5: 	

Table 58. Modifications of Function Designer Generated Code (Continued)

Case Id	Modification	Source	State Change	Description
IO7	<p>You move an IO signal from one Hardware Unit to another one.</p>	PE	F5: 	Any 'Loaded' diagram that contains the signal is changed to the state 'Generated' to indicate to the user that the diagram data does not correspond to the running application anymore.

Table 58. Modifications of Function Designer Generated Code (Continued)

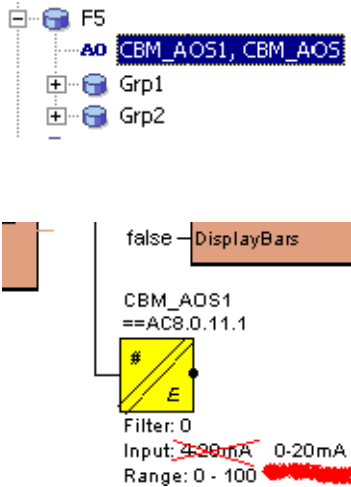
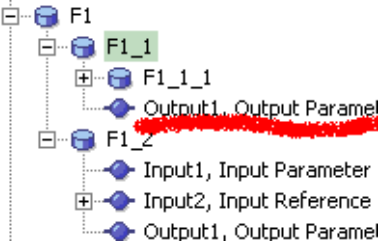
Case Id	Modification	Source	State Change	Description
IO8	<p>You change one of the parameters of an IO signal</p> 	PE	None	<p>The diagram state is not modified on signal parameter changes. No new Function Designer configuration data generation is necessary. However, in order to update the signal parameter values in Control Builder, you should use the IO Allocation function Write Allocation to CBM of Engineering Workplace.</p> <p>Be aware that change of signal name (case IO3) does change the state, AO7 applies.</p>
PA1	<p>You add a diagram parameter</p> 	FD PE	F1_1:	<p>The Single Control Module that represents the diagram changes its state to 'Modified'. The containing diagram is not affected.</p> <p>Any open Function Diagram is updated.</p>

Table 58. Modifications of Function Designer Generated Code (Continued)

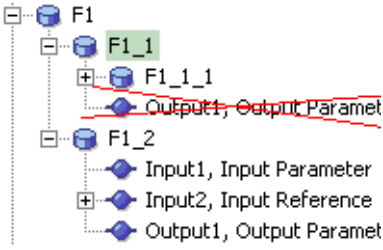


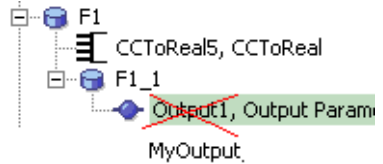


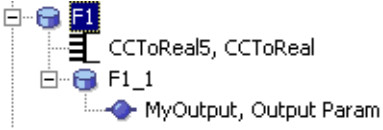


Case Id	Modification	Source	State Change	Description
PA2	You remove a diagram parameter 	FD PE	F1_1:  F1: 	The Single Control Module that represents the diagram changes its state to 'Modified'. The containing diagram also changes its state to 'Modified', because any connection to the removed parameter is also deleted. Any open Function Diagram is updated.
PA3	You rename a diagram parameter 	FD PE	F1_1:  F1: 	The affected Single Control module changes the state to 'Modified'. The parent diagram (Functional Structure) also changes its state to 'Modified'. Any open Function Diagram is updated.
AL1	You select 'Unallocate' 	FD PE	F1, F1_1: 	The effected diagram and all its children in the Control Structure are changing their state to 'Unallocated'.
AL2	You select 'Allocate'	FD PE		The effected diagram and all unallocated children in the Functional Structure are allocated. Their state is change to 'Modified' or 'Generated'.

Table 58. Modifications of Function Designer Generated Code (Continued)

Case Id	Modification	Source	State Change	Description
AL3	You allocate an IO signal using the IO allocation tool.			See IO5
CB1	You try to alter the code block of a Function Designer generated Single Control Module in the CBM.	CBM	None	The Function Designer has ‘locked’ the Single Control Module. CBM will reject any modifications.
CB2	You add a new variable or a new Function Block to a Single Control Module using the variable pane of the Control Builder.	CBM	None	The Function Designer has ‘locked’ the Single Control Module. CBM will reject any modifications.
FD1	You alter the data flow order	FD		The state of the diagram that contains the elements changed to ‘modified’.

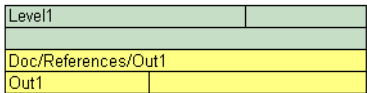
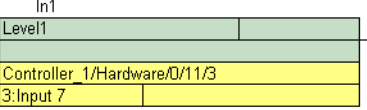
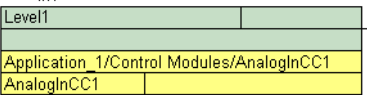
The above table lists the different code relevant changes and their consequence.

Cross References

The global variables that the Function Designer creates in Control Builder can also be used by parts of the application that are not generated by Function Designer through cross references.

[Table 59](#) shows, how Function Designer shows the different usages of the variables that are created by the Function Designer.

Table 59. Cross References

Variable Usage	Display	Description
Usage in a different diagram	Reference string in Functional Structure. 	This reference is always displayed.
IO Connection	Reference string in Control Structure plus IO channel name. 	The reference is displayed in diagram states Allocated, Modified, or Loaded.
Usage in a Control Module	Reference string in Control Structure. 	The reference is displayed in diagram states Allocated, Modified, or Loaded.
Usage in a Program	Reference string in Control Structure.	The reference is displayed in diagram states Allocated, Modified, or Loaded.

Cross-references to other Function diagrams are updated when the reference gets modified. Cross-references to I/O, into control modules or into programs are updated when the Function diagram gets opened.

You can navigate to any cross-reference by the command **Goto Reference**. In case of cross-references to I/O, into control modules, or into programs you can navigate directly into various Control Builder editors:

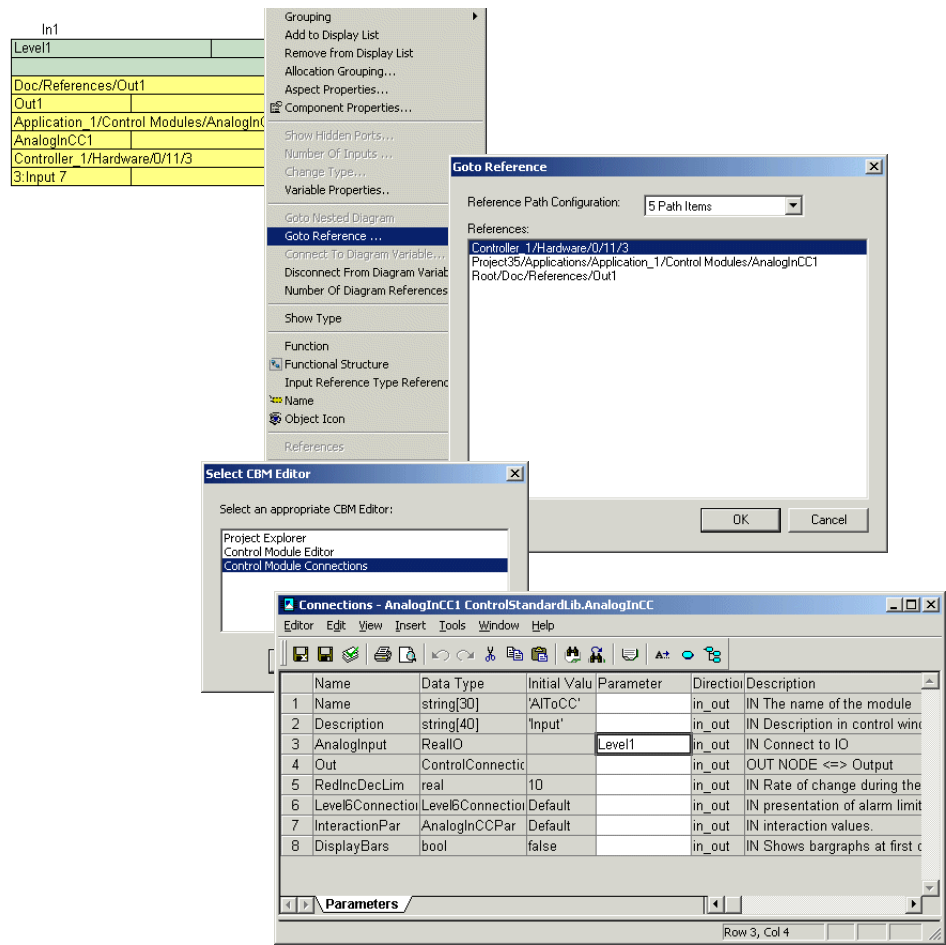


Figure 259. Goto Reference Inside Control Builder

Control Application Allocation Change

MMS communication: There is no message when control application allocation is changed.



It is possible to download code even though the cross communication is not consistent. Create cross communication once more after allocation changes.

MMS Cross Communication

To establish communication between different applications of one AC 800M controller or of several AC 800M controllers MMS cross communication links have to be configured. Access variables need to be created and both on the sender side and on the receiver side some logic needs to be configured using function blocks / control modules of MMSCCommLib library.

Function Designer provides a MMS cross communication generator. It supports configuring MMS cross communication links by automatically generating all required access variables, their data types and the necessary logic included in sender and receiver diagrams.

For details on configuring the communication between a restored Function Diagram and a newly created 800xA 5.1 Function Diagram, refer to [Connection between Restored Diagrams \(created upto 800xA 5.0 SP2\) and New Diagrams \(created from 800xA 5.1 onwards\)](#) on page 102.



From 800xA 5.1 onwards, an unavailable **Create FDCodeBlock** check box in the **Allocatable Group** window indicates whether the Function Diagrams are available as **Diagram** or **Single Control Module** in Control Builder and Control Structure.

By default, if the unavailable **Create FDCodeBlock** is in selected state, then the Function Diagrams are available as **Diagram** in the Control Builder and Control Structure (only for 800xA 5.1 systems and systems restored to 800xA 5.1).

By default, if the unavailable **Create FDCodeBlock** is not in selected state, then the Function Diagrams are available as **Single Control Module** in the Control Builder and Control Structure (only for systems prior to 800xA 5.1).

MMS Communication Overview

Suppose there exist two diagrams which are connected by two diagram references Variable1 and Variable2: SenderDiagram1 and ReceiverDiagram1.

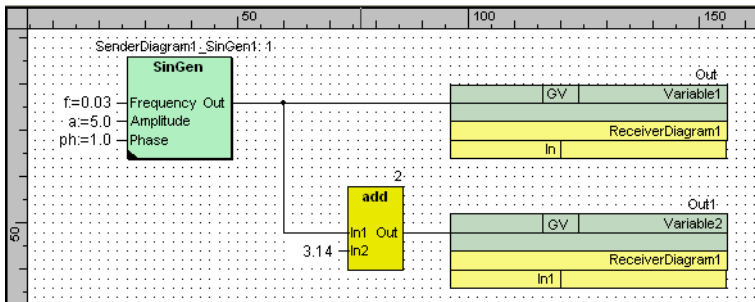


Figure 260. SenderDiagram1

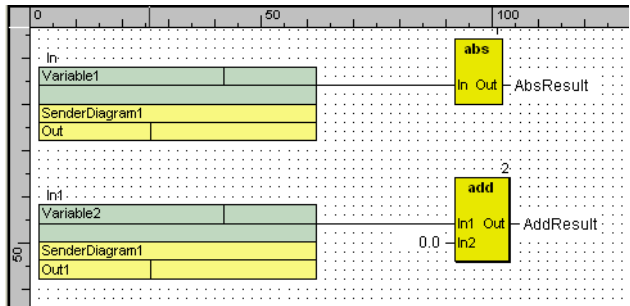


Figure 261. ReceiverDiagram1

If these diagrams later on are allocated in different applications, there exists a hint on the receiver side that MMS communication is required.

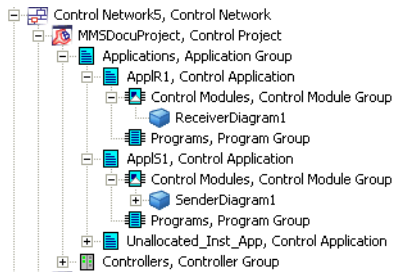


Figure 262. Allocation of Sender and Receiver Diagram in Different Applications

An additional field in blue is added to the diagram reference symbol and its CrossCommunication aspect property is set to 'MMS':

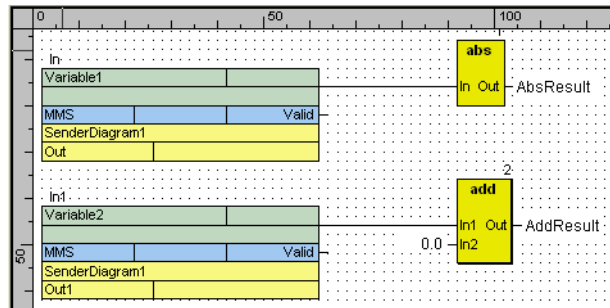


Figure 263. ReceiverDiagram1 with MMS Diagram References and Valid Port

In order to establish MMS communication some additional configuration is required. An access variable has to be created.

- Variables Variable1 and Variable2 need to be connected to this access variable on the sender side.
- Variables Variable1 and Variable2 need to be read on the receiver side by the function blocks 'MMSConnect' and 'MMSReadCyc' (or control module MMSToCC in case of data type ControlConnection) from the 'MMSCommLib' library.

On request, MMS CrossCommunication generator of Function Designer can create the required additional access variables and function blocks / control modules by creating some additional diagrams, one diagram for the sender of a cross communication and diagram(s) for the receivers of this sender (one diagram for each application where receivers occur).

Cross communication using variables of data type ControlConnection is supported using one fixed access variable per ControlConnection variable connection.

User can also take advantage of the boolean Valid port displayed at the MMS diagram reference: Here user will get a validation flag indicating that the corresponding MMSReadCyc function block or MMSToCC control module was able to read the required signal from the sender side. User can connect the Valid port to some glue logic to evaluate this condition within the loop.

If user does not want to get automatic MMS cross communication for the variables, user can set the CrossCommunication aspect property of these diagram references to 'manual'. Then user has to take care for cross communication configuration for these variables.

Additional basic information on MMS communication is given in

System 800xA Control, AC 800M Configuration (3BSE035980) and AC 800M Communication Protocols (3BSE035982*).*

Prerequisites

Before generating MMS cross communication the following prerequisites need to be fulfilled:

- All diagrams containing MMS diagram references need to be allocated to applications.
- Generation of configuration data need to be finished for these diagrams.
- Applications running these diagrams need to be connected to controllers and to tasks.

Reasons:

- The function block MMSReadCyc or control module MMSToCC reads a variable from a sender. Therefore the IP address of the controller of the sender side has to be known.
- The function block MMSReadCyc or control module MMSToCC has a cycle time for reading the variable. A default for that cycle time is calculated according to the scheme:
 - Take the interval time of the task of the sender diagram and take the interval time of the task of the receiver diagram. Variables shall be read as double as fast as the slowest interval time.
 - That default can be modified by [MMS Editor](#) on page 553.

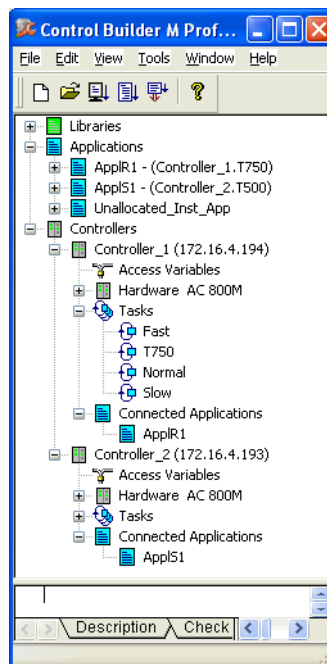


Figure 264. Prerequisites for Automatic Cross Communication Fulfilled: Applications Connected to Controllers and Tasks; IP Addresses Configured.



The length of a MMS cross communication message is limited to 957 bytes. This message size is defined by a MMSReadCyc function block. The length of this message is calculated by the data types of the access variables of all input terminals of this MMSReadCyc block.

Function Designer's MMS CrossCommunication generator is considering this message size: In a sender diagram, global variables are collected into one access variable as long as the message size is not reached. If the message size is reached (regarding some overhead message load), a new access variable is created and following variables are collected into this new one. So one access variable, composed of a lot of global variables, is limited by the maximum message size. Therefore only MMSReadCyc[1] function blocks with only one input parameter are used.

When using the ControlConnection data type and MMSToCC control module only one global variable is handled by one access variable.



1. Consider the following while naming manually created cross communication Function Diagrams that configure cross communication between different applications, projects, and control networks:

a. Do not use **S_MMS** or **R_MMS** in the naming convention. If they are used, the Function Diagrams will get deleted while performing **Clear Generated Data** context verb on the MMS cross communication aspect.

b. Diagram names must be unique.

2. Save, manually created Function Diagrams that configure cross communication between different applications, projects, and control networks using MMS blocks in a separate folder. Do not save in Function Designer created CrossCommunication_<project name> folder within the Functional Structure.

MMS Cross Communication Generation Principle.

The example in this subsection explains the basic steps how MMS cross communication diagrams are generated and how adding and deleting a diagram variable is handled.

1. Create three Function Diagrams, each one allocated to another Application. The diagram variable Variable1 is source for the diagram reference in

FunctionDiagram1 and sink for the diagram references in FunctionDiagram2 and FunctionDiagram3.

2. Perform Generate Configuration Data for the three Function Diagrams. The result is that the Single Control Module for each Function Diagram is created in the Application and the global variable Variable1 is created.
3. Perform Analyze and Create MMS, using the MMS CrossCommunication generator.

For each Application the same Access Variable datatype (in [Figure 265](#) sketched as DataType1) gets created with the same diagram variable e.g. Variable1.

And for each Application a Send or Receive Diagram is created, each represented by a corresponding Single Control Module in Control Builder M.

In the Send Diagram a packer variable of DataType1 with connection to the variable Variable1 gets generated. This is connected to an MMSDefAccVar Function Block.

In the Receive Diagram an MMSReadCyc Function Block gets connected to a unpacker variable of DataType1 with connection to Variable1.

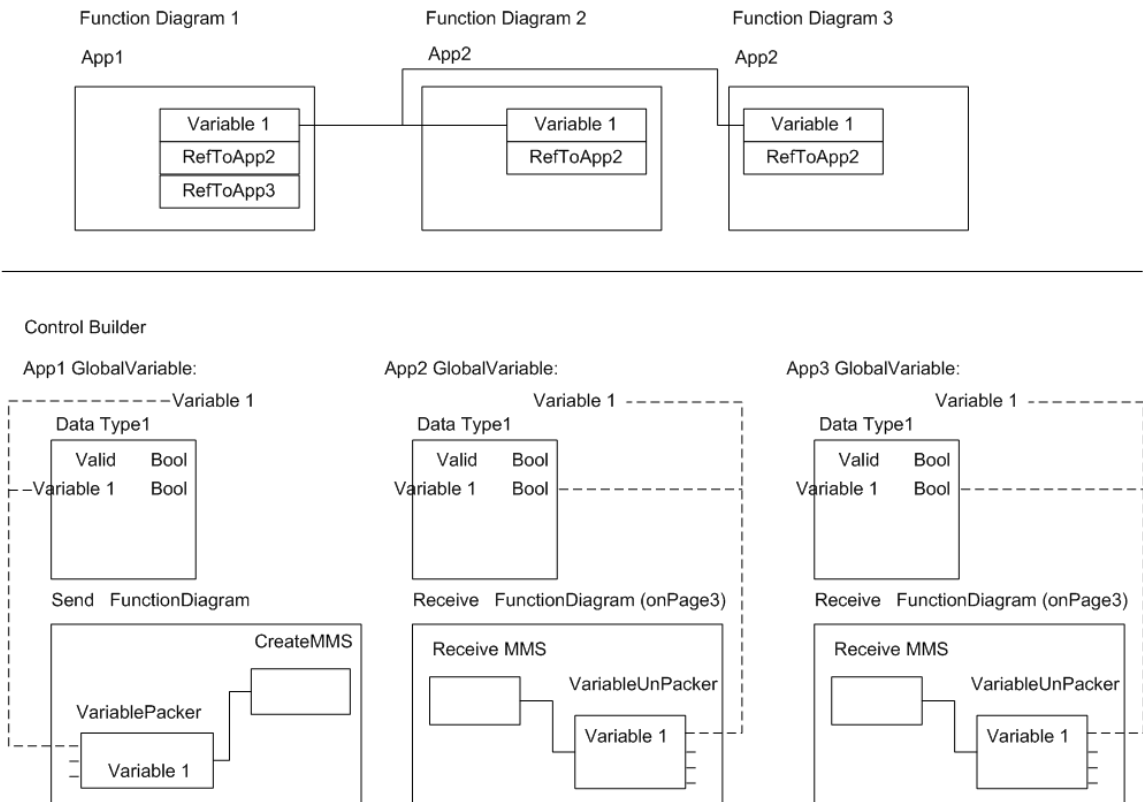


Figure 265. MMS Cross Communication Principle

Adding a Diagram Variable

For an existing MMS cross communication link add a diagram variable.

1. Create an additional output diagram reference connected to a newly created diagram variable Variable2 in source FunctionDiagram1 and only one sink (input diagram reference connected to the variable Variable2) in FunctionDiagram3.

2. Now perform Generate Configuration Data for FunctionDiagram1 and Function Diagram3. The result is that the Single Control Modules for both Function Diagrams are re-created in the Application, now including the global variables Variable1 and Variable2.
3. Then use MMS CrossCommunication generator and perform Analyze and then Create Differences. The changes are taken care in MMS cross communication configuration as follows:
 - The access variable data type (DataType1) is updated in all three applications to hold Valid, Variable1 and Variable2.
 - The Send Diagram is re-created to also pack Variable2 and to re-create the access variable according to changed DataType1.
 - Both Receive Diagrams are re-created to also unpack Variable2 according to changed DataType1.



If only the Sender application is downloaded first and the Receiver Applications are not yet loaded then the variables in FunctionDiagram2 and FunctionDiagram3 according to Figure 118 (which are in Receiver Applications) will hold the last Retain Values. MMSReadCyc Function Block will set communication invalid with error code -463.

The same behavior is observed in cases where Function Designer is not involved (i.e.) engineering is done in CBM for MMS communication.

Deleting a Diagram Variable or Changing the Name

If the user deletes the diagram reference with variable Variable2 in source FunctionDiagram1 or rename the diagram variable, the same steps as described for the adding of the diagram variable apply. As result we have changed FunctionDiagram1, FunctionDiagram3, the Send Diagram and the Receive Diagrams

Regarding download the same information as given in the above warning box applies.

How to Start Generation

The MMS CrossCommunication generator is started by the object verb

Advanced > Cross Communication on

- Control Project,
- Control Network or
- Control Structure Root level.

In case you configure cross communication across several Control Projects or networks, use the verb on Control Network or Root level. If you configure cross communication within a single Control Project, use the verb on project level.



A CrossCommunication object is only allowed as child of one level:

Either parallel objects on project level (in case of several projects) or parallel objects on Network level (in case of several Networks) or only one object on Root level.



If Cross Communication is performed from a generic object, the following message is displayed: **Please select a CBM project.**

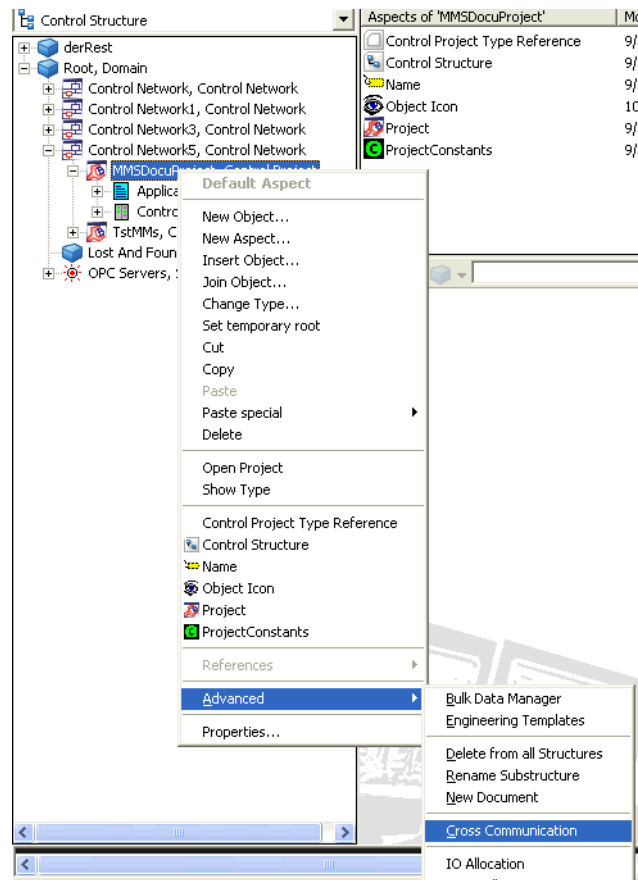


Figure 266. Starting CrossCommunication by Object Verb



The Advanced menu is available in Engineering Workplace, but not in Plant Explorer Workplace.

After starting MMS CrossCommunication generator a generic object named '**CrossCommunication_<startingObjectName>**' gets created. It is placed

- inside Functional Structure on Root level

- inside Control Structure below the object where the object verb was started.
- A **CrossCommunication** aspect is added.

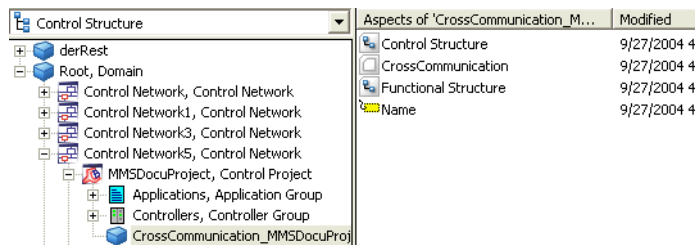


Figure 267. New Object with CrossCommunication Aspect

The CrossCommunication aspect offers as main view an editor where you can start and influence the generation of MMS cross communication:

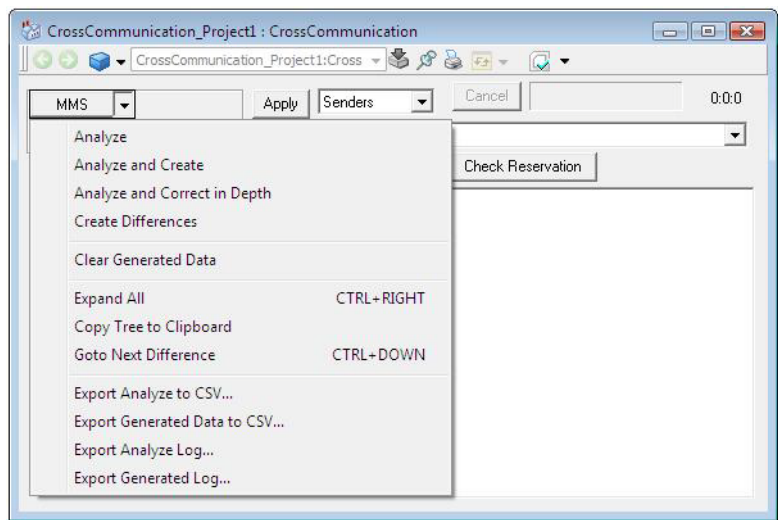


Figure 268. Main View of Aspect CrossCommunication

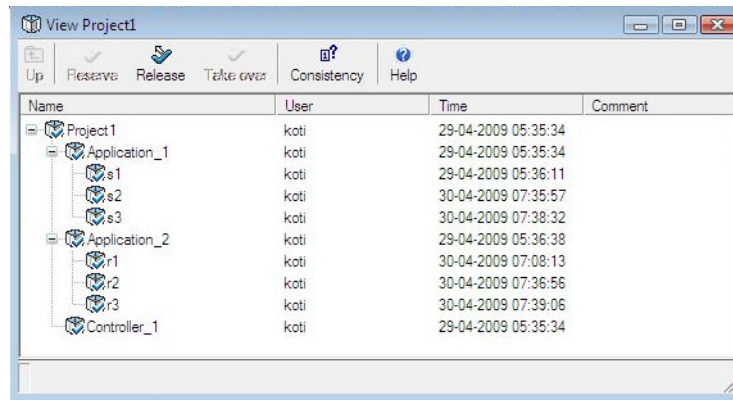


Figure 269. Reserve / Release Dialog

Workflow of Generation

The automatic generation of MMS communication requires the following workflow:

1. Make sure that the prerequisites for using the MMS CrossCommunication generator are fulfilled. See [Prerequisites](#) on page 540.
2. Start up the MMS CrossCommunication generator on the appropriate Control Structure level. See [How to Start Generation](#) on page 545.
3. Click **Check Reservation**.
 - The reservation dialog window appears with reservation details of the project, applications, and all the diagrams under that application (if any of the MMS Cross Communication established project or application or diagrams available in that application is reserved by another user, then **MMS > Analyze** fails at the end of the analyze operation). User can modify the reservation status.



If the cross communication object is available in the root or control network level, then the reservation dialog box opens in sequence.



It is advisable to perform **Check Reservation** before executing **MMS > Analyze** or **MMS > Analyze and Create** or **MMS > Analyze and Correct in Depth** menu commands.

4. Click **MMS > Analyze** to perform the analysis:
 - All diagrams below the object containing this CrossCommunication aspect (e.g. of the whole Control Project or even of a whole system) are examined for cross communication
 - All global variables of diagram references found on diagrams belonging to different applications are collected into access variables on the sender side
 - They are read from these access variables on receiver side with an interval time, calculated as mentioned above.

The analysis result is presented in different views for sender side, receiver side or variables. See [MMS Analysis](#) on page 552.



In some situations (after import/copy application or after import project) the CrossCommunication aspect property of a diagram reference may set wrong such as:

- CrossCommunication aspect is set to "None" instead of "MMS" and the receiver diagram reference do not show the presence of global variable across applications by the additional blue color field.
- CrossCommunication aspect is set to "MMS" instead of "None" and the receiver diagram reference shows additional blue color field even though there is no global variable present across applications.

MMS > Analyze and Correct in Depth option can be performed in this situation to check the correctness of CrossCommunication aspect property of all diagram references and afterwards to execute the **Analyze** command.

Alternatively, the **Consistency Check** tool or the **Consistency Checker** aspect will also perform the same action.



Analyze and Correct in Depth command is slower than **Analyze** command.

5. If required edit modifications in the MMS editor's tree window (in view Senders / modifier Actual Analyze only):

After analysis, optionally some modifications can be done within the MMS editor, for example:

- Rename access variables
- Move variables to other Access Variable (except for ControlConnection variables).
- Change calculated cycle time.



Note that modifications edited in the analysis tree are overwritten by a next analysis!

See [MMS Editor](#) on page 553.

6. Analyze and Create Diagrams.

- a. Click **MMS > Analyze and Create** to analyze (refer to [Step 4](#) for details about the analysis) and create all additional diagrams. After analysis and optional modification, all additional diagrams for sender and receiver side of cross communication can be created.
All variables are collected into one or more access variables of a new structured data type. This data type gets created within the required applications.

See [Analyze and Create](#) on page 559.

- b. Or, if prerequisites listed below are fulfilled: Click **MMS > Create Differences** to re-generate just those additional diagrams covering the differences to the previous generation.

Prerequisites for **Create Differences** are:

- A first complete generation of diagrams for sender and receiver side of cross communication has already been done.
- Then you have done changes in parts of the cross communication configuration.
- You have performed a new analysis before Create Differences.

Optional modifications done in the differences after the analysis are taken care.

See [Create Differences](#) on page 566.

- 7. Optional: Verify generated diagrams, access variables and data types. You find them in the Functional Structure under an object named CrossCommunication_<startingObjectName> in the generated cross communication objects below this object.
- 8. Download Control Project(s) to the controllers, go online and test the in connections to verify the communication functionality of the downloaded generated code.

See [Download and go Online](#) on page 568 for some test possibilities.



Do not configure Function Diagrams connected by diagram references requiring MMS communication while MMS analysis (Analyze) and corresponding MMS generation (Analyze and Create MMS or Generate Difference) is in progress.

MMS Analysis

All diagrams are searched for necessary (non-manual) cross communication. The result are presented in the MMS editor. In case there are diagrams in state ‘modified, red’ which need cross communication, an error log is displayed.

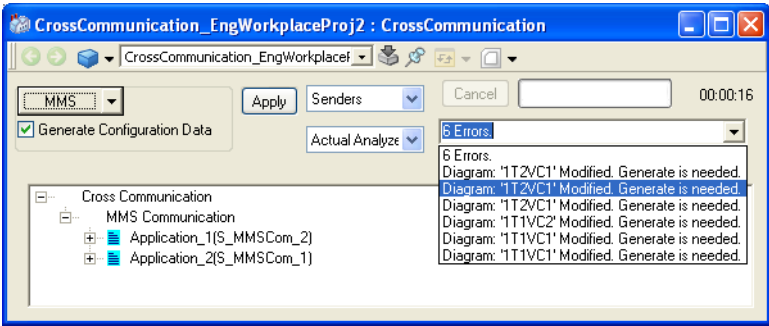


Figure 270. Errors During Analysis Phase

Only diagram references with CrossCommunication aspect property set to ‘None’ or to ‘MMS’ are regarded. Unallocated diagrams assigned to Unallocated_Inst_App application are not regarded at all.

If the analysis is done without errors, create MMS configuration data continues. After creation, the user can check the analyzed and created data.

MMS Editor

The CrossCommunication aspect displays the MMS Editor (see also [Figure 268](#)) in its main view.

MMS Menu

- **Analyze** - Analyses the Function Diagrams for MMS communication, builds up the Analyze result tree and displays the results in the Editor Tree view.
- **Analyze and Create** - Always analyzes and creates additional MMS communication diagrams newly according to the Actual Analyze tree and deletes old diagrams.
- **Create Differences** - Always only re-generates those additional MMS communication diagrams that have to be changed. Only active if differences are available.
- **Copy Tree to Clipboard** - Copies the Editor Tree view to the clipboard.
- **Goto Next Difference** - Selects the next difference in the Editor Tree view.
- **Export Analyse to CSV...** - Exports the analysis tree result to a .CSV file.
- **Export generated data to CSV...** - Exports the generated data tree result to .CSV file.
- **Export Analyse Log...** - Exports the analysis (**Analyze**) error logger results to a text file.
- **Export Generated Log...** - Exports the generated (**Analyze and Create**, **Create Differences**) error log to a text file.

Buttons

Apply - Saves the changes (optional modifications) done in Editor Tree view, Analyze tree

Cancel - Cancels the current action.

Check Reservation - Opens the reservation dialog window with reservation details of project, applications, and diagrams.

Check Box

- **Generate Configuration Data** - Automatically starts the configuration data generation if **Analyze and Create** / **Create Differences** menu items are activated.

Views and Modifiers

The result of the analysis is presented by a tree view in three different views:

- Senders (default), supported modifiers: Actual Analyze (default), Differences, Generated Data.
- Receivers, supported modifiers: Actual Analyze (default), Differences, Generated Data.
- Variables, supported modifiers: Actual Analyze (default), Generated Data.

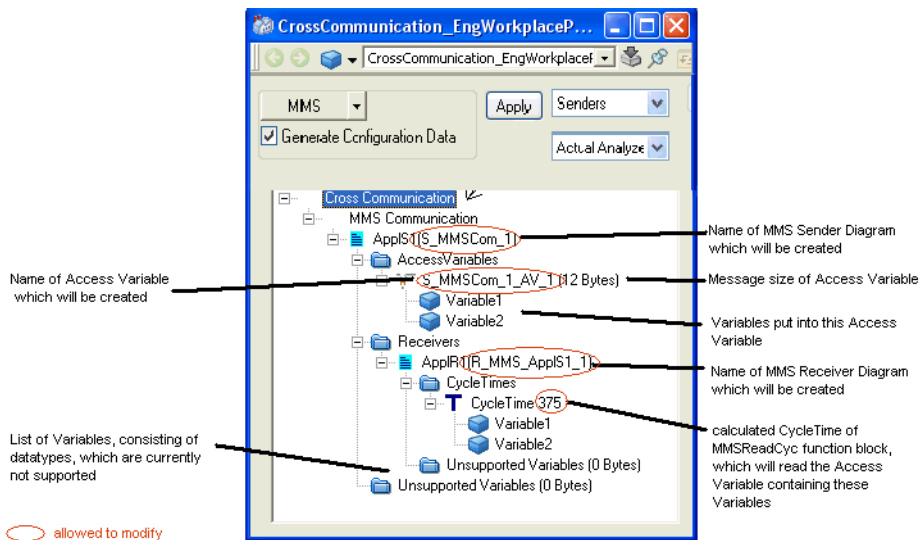


Figure 271. MMS Editor Displaying Sender View

The default **Sender view** lists the applications which contain diagrams with output diagram references (as senders of cross communication). An additional sender diagram gets created and named as displayed in parenthesis:

- Below this new sender diagram the created access variables are displayed together with the collected global variables. The current message size of this access variable that is based on the data types of the global variables gets displayed, too. Later during creation of MMS diagrams the required data types of these Access Variable are created inside the required applications as application defined data types.
- Furthermore the MMS receiver applications belonging to a MMS sender are listed. These are the applications which contain diagrams with input diagram references (as *receivers* of cross communication data). An additional receiver diagram gets created and named as displayed in parenthesis. Its children are the global variables read by this receiver diagram and corresponding access variable. The global variables read are sorted by the calculated default cycle time (calculation depends on connected tasks of original sender and receiver diagram, default can be changed, see below).

The other views for receivers and variables are displayed in a similar way, starting with receiver diagrams or variables respectively.

It is recommended to work with the default sender view.

The views can be modified using the modifiers:

- **Generated Data**
Shows the result of the analysis tree of the last creation of MMS communication. Editing in the tree is not allowed.
- **Actual Analyze**
Shows the result of the current analysis tree. Editing in the tree is allowed.
- **Differences**
Shows the differences between the already generated data and the current analysis tree data.

Icons in the tree are marked if modified, deleted or additionally appearing, see [Figure 272](#). Editing in the tree is not allowed.

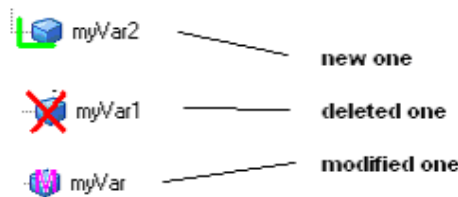


Figure 272. Icons Within the Differences Tree

Tool tips get displayed for applications or global variables:

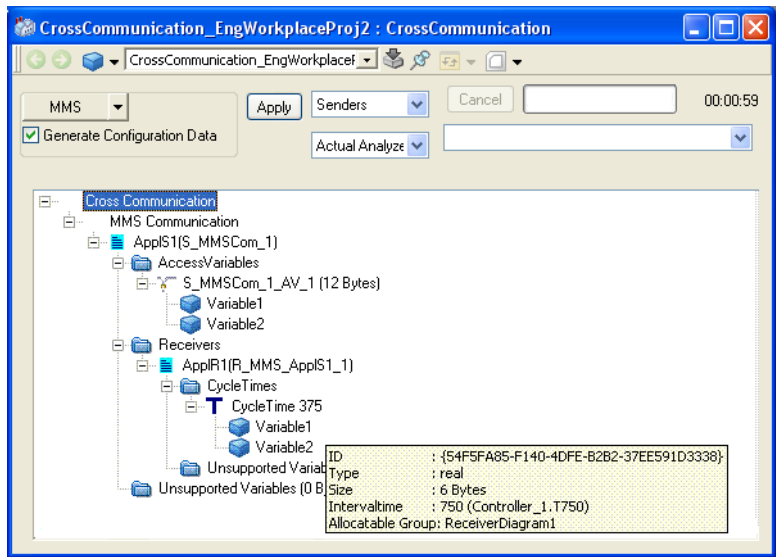


Figure 273. Tool Tip for a Global Variable

Editing in the Actual Analyze Result Tree



Modifications edited in the analysis result tree are overwritten by a next analysis and have to be re-edited afterwards if they are still needed.



Only an experienced user should do any modifications at all.

Inside the MMS editor the following modifications are supported if modifier Actual Analyze is switched on:

- Rename
 - MMS sender diagram(s)
 - MMS receiver diagram(s)
 - Access variable(s).

Press the F2 key or left click in the name field or right click on parent tree node.



Do not re-name access variables in MMS Editor, and give name manually created access variables according to the default scheme S_MMS.

- **Change cycle time.** Press the F2 key or left click on the Cycle Time tree node field.



Following are the recommendations for optimal MMS communication:

- The task interval on the receiver side (generated R_MMS diagram) should be at least twice as fast as (half of) the cycle time parameter on the MMSReadCyc block.
- The task interval of the sender side (generated S_MMS diagram) should be double of the receiver side task interval, which means equal to the Cycle Time parameter for MMSReadCyc block.
- **Add new access variable** nodes by right mouse click on parent tree node.

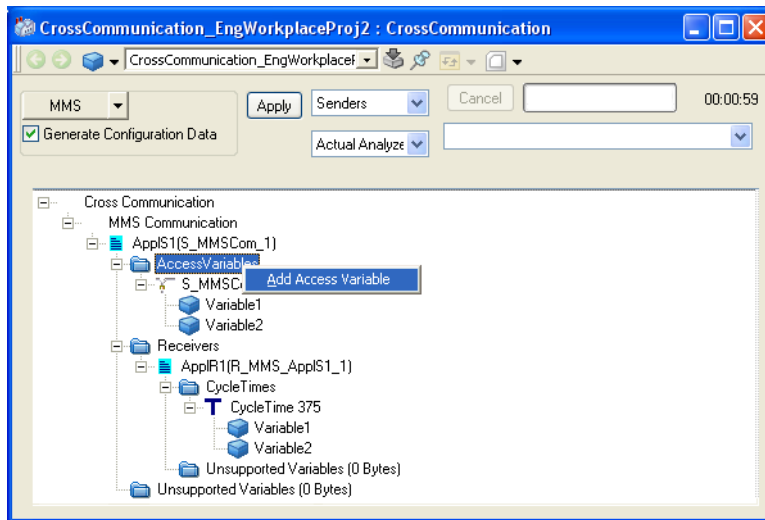


Figure 274. Adding new Access Variable in MMS Editor

A new access variable is empty (message size 0 bytes). The message size is calculated when variables to be transferred via this access variable are moved below it, see Move variables described below.

- **Add new cycle time** nodes by right mouse click on parent tree node.
- **Move variables** between access variables or between cycle times. The new message sizes are shown at the changed access variables.



Create MMS results in an error if an access variable becomes empty after moving its variables to some other access variables.



It is not possible to move variables using ControlConnection variables.

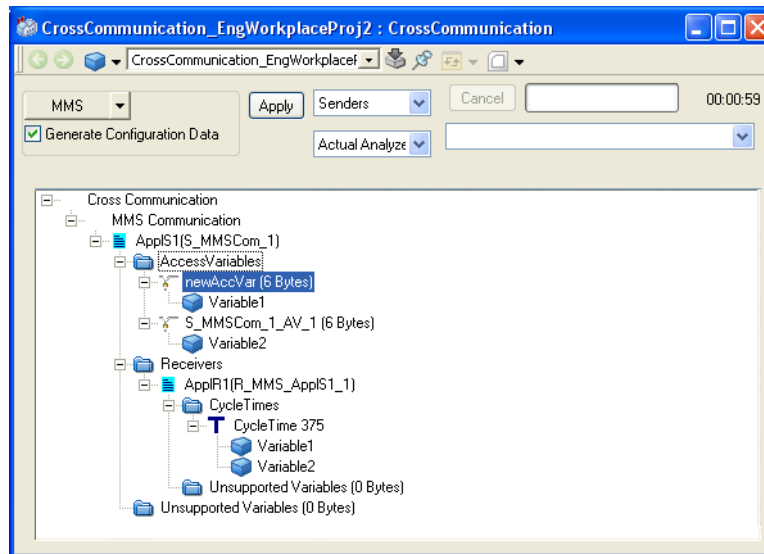


Figure 275. Moving a Variable From one Access Variable to Another one

- **Delete** empty access variables or empty cycle times. Right click on tree node.
- **Print** the tree: The tree gets printed by pressing 'CTRL+F5' keys.
- **Expand** the tree by selecting a node and pressing the 'CTRL+ right arrow' keys. Collapse the tree by pressing the 'CTRL+ left arrow' keys.
- **Copy the tree into the clipboard** (as *.emf format): Press 'CTRL+F6' keys. (Advantage of the *.emf format: Text inside the picture can be edited.)

Click **Apply** to save the modifications.

Analyze and Create

All required MMS sender and receiver diagrams, access variables, and necessary data types of access variables are generated after analysis.

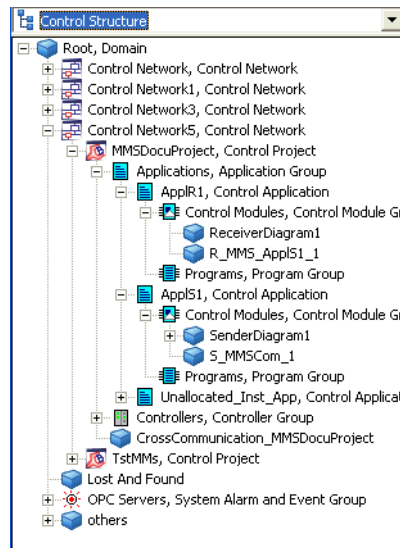


Figure 276. Control Structure and Functional Structure after MMS Creation

The additional created diagrams are placed below the object which contains the CrossCommunication aspect in the Functional Structure and below the corresponding applications in Control Structure.

Default naming is done according to [Table 60](#).

Table 60. Default Naming done by Create MMS

Item	Sender Side	Receiver Side	Remark
Top Object	CrossCommunication_<startingObjectName>		Object with CrossCommunication aspect in Functional Structure, also placed in Control Structure
Object	S_MMSSCom_<sn>	R_MMS_<applicationname>_<rn>	sn Sender Number rn Receiver Number
Diagram	S_MMSSCom_<sn>	R_MMS_<applicationname>_<rn>	
Access Variable	S_MMSSCom_<sn>_AV_<avn>		avn Access Variable Number
Data Type of Access Variable	S_MMSSCom_<sn>_AV_<avn>_DT		application specific
Variable for Valid Flags		R_MMS_<applicationname>_<rn>_V	
Data Type of Variable for Valid Flags		R_MMS_<applicationname>_<rn>DT	application specific

Maximum Limit: User needs to ensure that the maximum number of characters including the default name (listed in [Table 60](#)) that get appended for Diagrams and Access Variables are as provided below:

- Access Variable Name <= 21 Characters
- Sender Diagram Name <= 28 Characters
- Receiver Diagram Name <= 28 Characters

If the maximum limit exceeds, then the following warning message appears:

Diagram Names are exceeding the limit of 28 characters: <Diagram Name> Please check.

The actual names of the created diagrams are related to the names configured in the MMS editor after analysis and optional modifications. For each access variable an application defined data type is created with substructures composed of the data types of the global variables collected into that access variable:

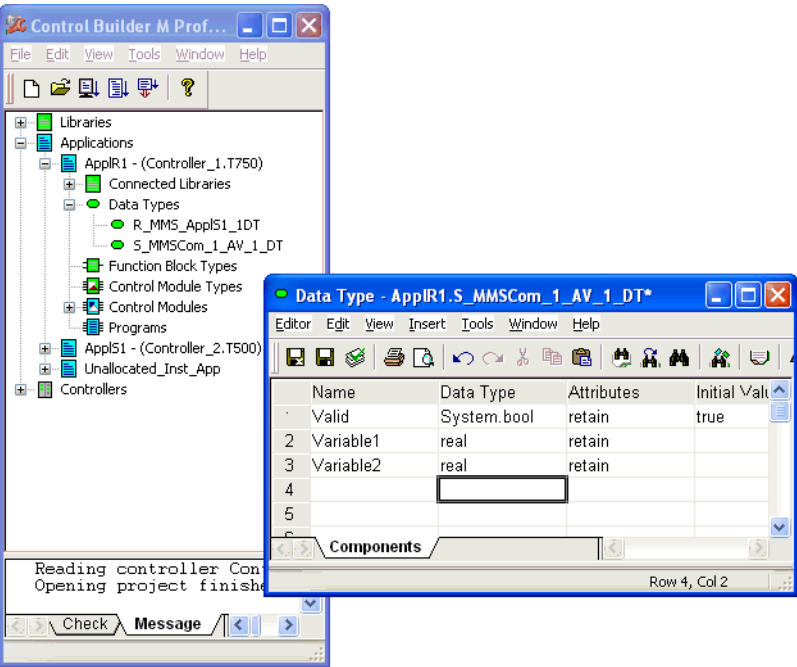


Figure 277. Created Data Type for new Access Variable



The following explanations are based on the use of function blocks MMSTConnect and MMSTReadCyc. ControlConnection variables are handled using control module MMSToCC accordingly.



Analyze and Create MMS / Generate Configuration Data generates code that creates access variables by means of `MMSDefAccVar`. Therefore the access variables are not visible in the Access Variables editor of Control Builder M Professional. But it is visible only in the MMS variables window shown by Show MMS Variables as described in [Download and go Online](#) on page 568.

An access variable is read by a `MMSReadCyc` function block. This function block contains a boolean `Valid` parameter indicating the result of the reading of that variable (the same port exist at the `MMSCConnect` function block).

These both boolean values are combined and sent to the `Valid` port of the input diagram reference of the original receiver diagram. In order to do that, a structured variable is created that collects all these valid flags in order to get only one global variable per each receiver diagram. Thus this new variable of structured data type collects all single boolean valid parameters of all `MMSCConnect` and `MMSReadCyc` function blocks (one per receiver diagram).

This new data type is also created within the required application:

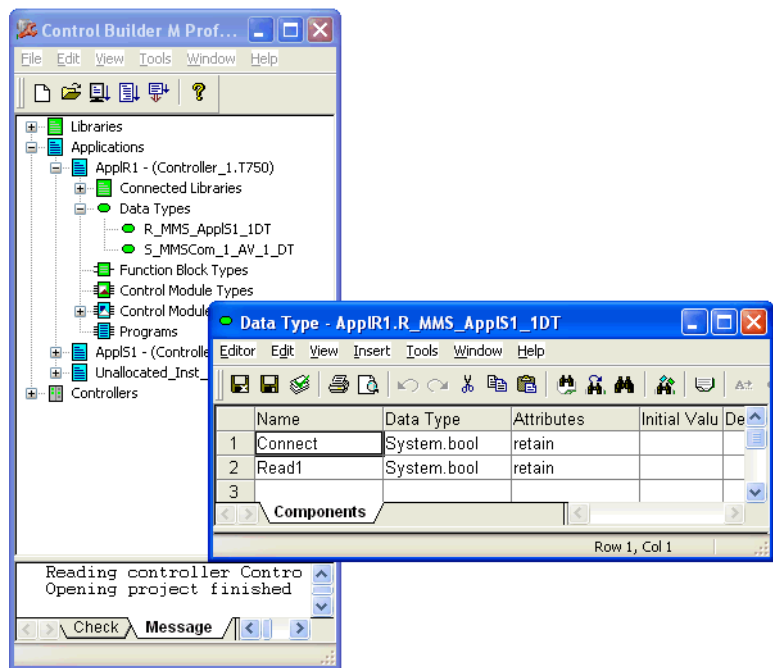


Figure 278. New Structured Data Type Containing Valid Flags

In the example of ReceiverDiagram1 there exist only one MMSReadCyc function block and one valid parameter which is collected into a structured data type and sent to the two input diagram references Valid ports.

Alarm and Events: Additionally the valid flags of the MMSConnect and MMSReadCyc function blocks are send to an AlarmCond function block in order to get communication errors in the Alarm and Event List.



If the MMS Control Modules are modified due to MMS Create or Create Difference operation and the changes are downloaded to the controller, the following problems exist for those alarms:

1. The status of the active alarms does not change in the alarm list, when the alarm condition recovers to normal.
2. The existing unacknowledged MMS alarms cannot be acknowledged.

However, the new alarms appear normally along with the existing list. These alarms need to be manually deleted from the alarm list.

To prevent this situation, ensure that there are no alarms related to MMS, before performing the MMS Create or Create Difference operation.

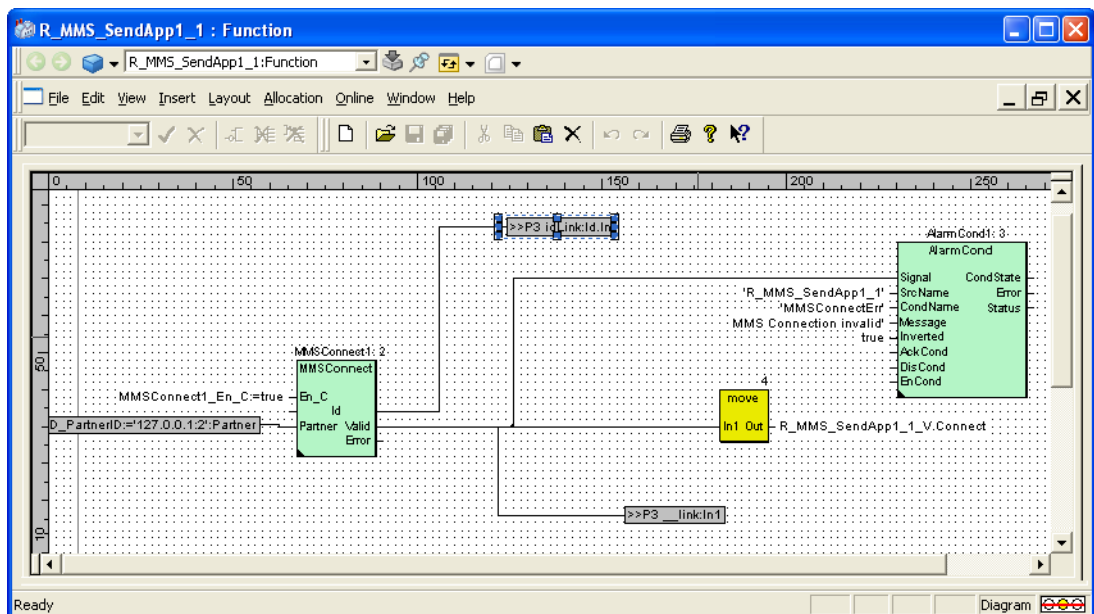


Figure 279. Page 2 of a Generated Receiver Diagram Containing an AlarmCond

After the MMS creation the diagram reference symbol(s) on the original receiver diagram ReceiverDiagram1 show the current cycle time, see [Figure 282](#).

Click **Cancel** to stop the creation process. Then the current transaction is finished and MMS creation is stopped. Diagrams already created are valid, but not all required ones are maybe available. Therefore a complete rerun is required afterwards in order to have a complete MMS communication generation.

Create Differences

Execute **MMS > Analyze and Create** command to analyze and create the MMS CrossCommunication.

After successful execution of **MMS > Analyze and Create**, if the existing diagram is modified (e.g. adding / deleting cross references, changing data types, modifying Cycle Times or IP addresses), always execute **MMS > Analyze** before performing **MMS > Create Differences**. It is also possible to generate the differences by executing **MMS > Analyze and Create** command. However, this will take more execution time.

Deletion of Obsolete MMS Diagrams

Altering the MMS configuration by deleting diagram references or disconnecting the diagram variables may result in obsolete MMS generated diagrams and datatypes.

These diagrams and datatypes are not deleted by **MMS > Analyze** or **Analyze and Correct in Depth** followed by **MMS > Analyze and Create** or **MMS > Create Differences**.

Perform the following to delete the obsolete diagrams and datatypes:

- If **MMS > Analyze and Create** is used:
Execute **MMS > Clear Generated Data** before executing **MMS > Analyze and Create** or **MMS > Analyze and Correct in Depth** followed by **MMS > Analyze and Create**.
- If **MMS > Create Differences** is used, perform either of the following depending on the scenario:
 - a. In case, cross communication happens between a sender diagram and receiver diagram across applications, and the respective sender / receiver diagrams are not involved in communication with any other diagrams. A tool tip (see [Figure 280](#)) on the CrossCommunication object in the Sender

diagram's **Differences** view provides the list of applications and variables that contain the obsolete MMS generated diagrams. User must navigate to the respective application and manually identify and delete the obsolete diagrams and datatypes individually.

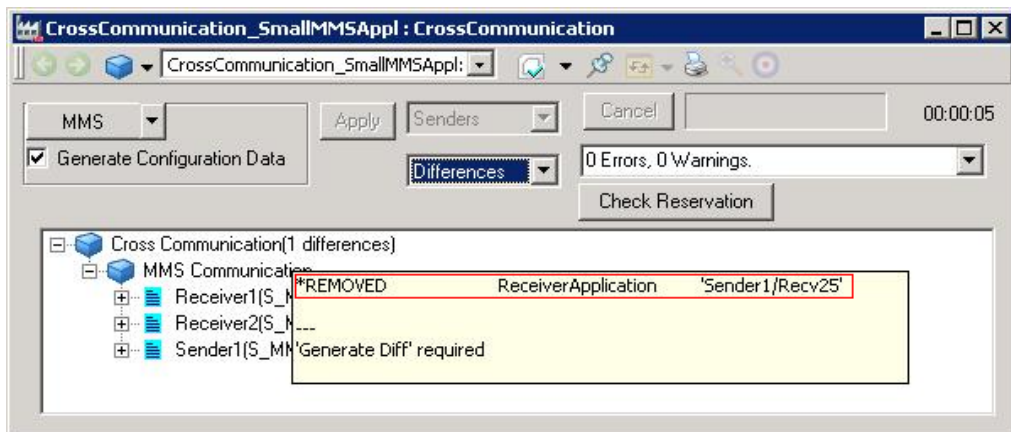


Figure 280. CrossCommunication between Single Sender and Receiver Diagram

- b. Incase, cross communication occurs between multiple sender and receiver diagrams across multiple applications, and the each sender / receiver diagram are involved in communication with other diagrams. The Sender diagram's **Differences** view provides the list of applications and variables including the names of the obsolete MMS generated diagrams (see [Figure 281](#)). User must navigate to the respective application and manually delete the obsolete diagrams and datatypes individually.

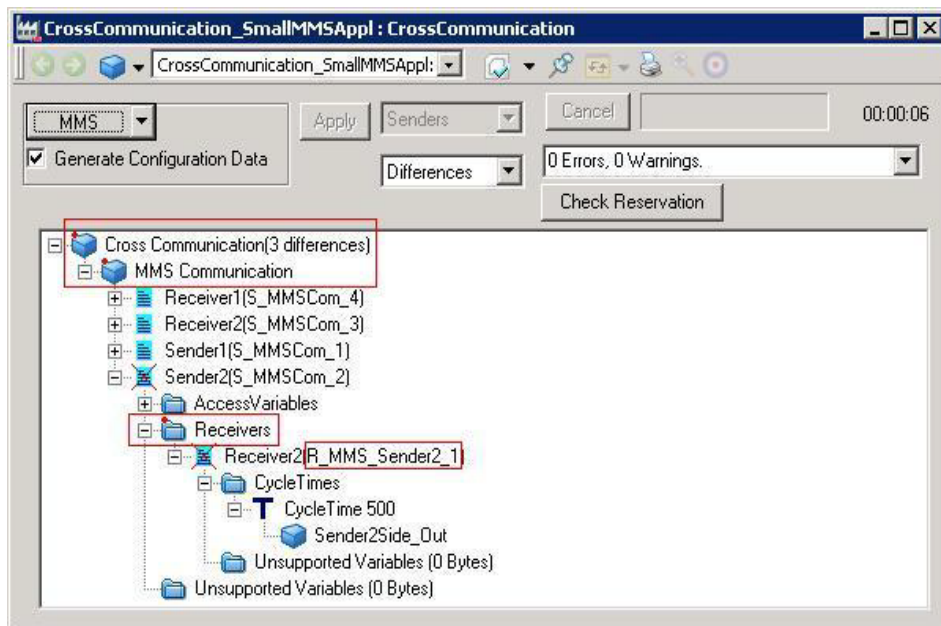


Figure 281. CrossCommunication between Multiple Sender and Receiver Diagrams

Download and go Online

After downloading the project data into the controller(s) you can check the Valid ports at the MMS diagram references. Subscribe for live data and select the Valid ports for display, see chapter [On-line Display](#) on page 184.

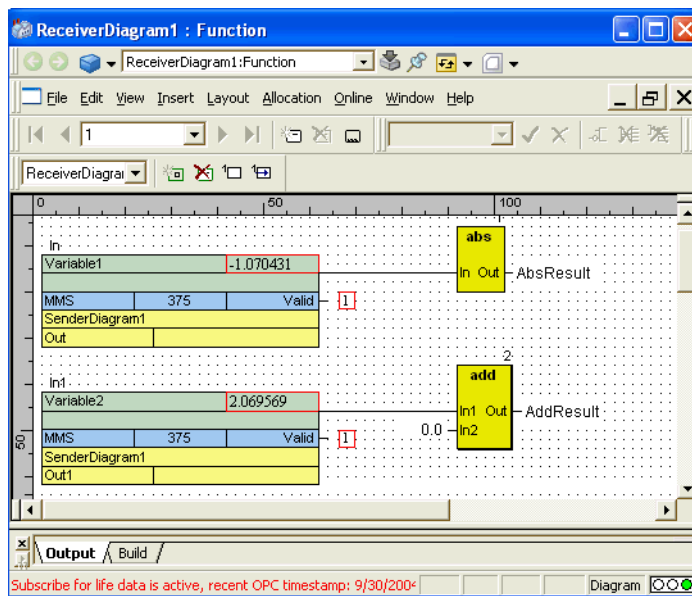


Figure 282. Diagram Reference Shows 375 ms Cycle Time for MMS

Follow the steps to check the access variables in CBM:

1. Right-click the required controller and select **Remote System**.
2. Click **Show MMS Variables**.

All MMS access variables corresponding to the selected controller are displayed in the **MMS Variables** window (refer to [Figure 283](#)).

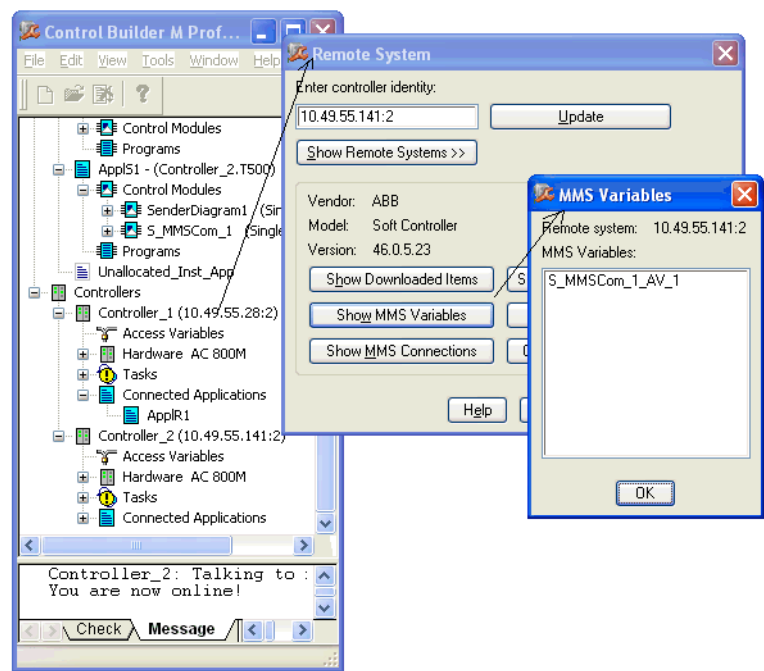


Figure 283. MMS Access Variables of Controller_2

By adding an **Alarm and Event List** aspect to your project or by using an existing one you can also verify correct behavior of MMS communication. In case of communication failures alarms indicating the corresponding receiver and sender gets generated.

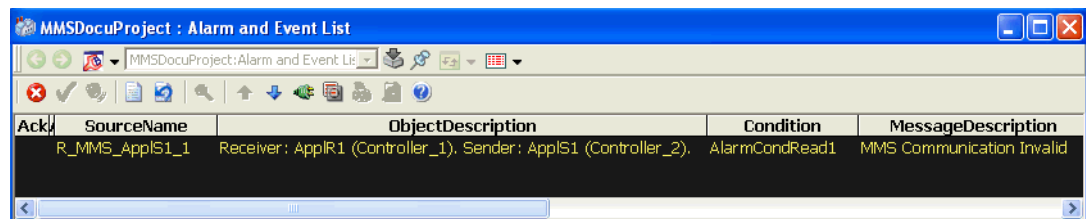


Figure 284. Alarm and Event List in Case of MMS Communication Failures

Aspect Verbs of CrossCommunication Aspect

The CrossCommunication aspect supports additional aspect verbs:

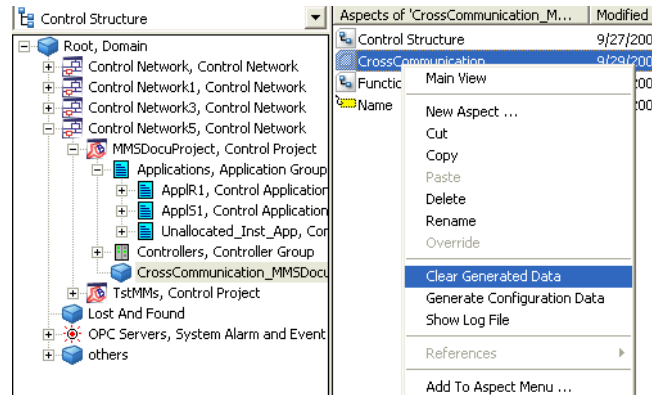


Figure 285. Aspect Verbs of CrossCommunication Aspect

- **Clear Generated Data** - All diagrams, global variables and application defined data types, created by the MMS CrossCommunication generator, will be cleaned. This will also be done automatically if you restart the generator for a second run.



Before performing **Clear Generated Data**, check the reservation details of all the MMS related application / Function Diagrams using the **Reserved by** option in the **Find Tool**. Release the relevant reservations performed by other users and then perform **Clear Generated Data**.

- **Generate Configuration Data** - If the user started creation by **MMS > Analyze and Create** without checking the check box **Generate Configuration Data**, then generation can be started by this aspect verb for all generated MMS diagrams (This step is necessary in order to transfer all configuration data into CBM for download).
- **Show Log File** - During the analysis or during the MMS creation step, you can see what is actually going on inside the error logger list box. When the run is finished, this list box shows the errors, but the content of this box is not saved.

If you want to see the errors again, you can open the XML log file for the whole process and search for errors.



Click "CTRL + F5" keys to print the Cross Communication Aspect window.

The results of **Analyze** and generation can be exported to .csv files, read into Excel, and also printed in tabular layout.

Current Limitations

- Generation of MMS cross communication is supported only for diagram references between diagrams. Connections between different allocatable groups allocated to different applications on one diagram are not supported.
- Only MMS communication is supported, other protocols have to be configured manually.
- Only function blocks MMSDefAccVar, MMSConnect and MMSReadCyc and control modules CCToMMS and MMSToCC from the MMSCommLib are actually supported.
- Connection to multiple output diagram references in different applications having same diagram variable name is not supported.
- If user is configuring the Root level Cross Communication object, there should not be any cross communication objects in Control Network and Project level. Otherwise this might lead to some inconsistency while creating the MMS Diagram and its variables.
- Do not use S_MMS or R_MMS in the naming convention. Manually created cross communication data types and diagrams under cross communication object having name starting with S_MMS, R_MMS will get deleted while generating cross communication using MMS Cross Communication aspect.
- Bidirectional MMS CrossCommunication is limited to ControlConnection data type. Function Designer does not support CrossCommunication using diagram references having data types that has InOut directions.

Outdated MMS Cross Communication Data

If you have finished MMS cross communication generation, and change afterwards the original sender or receiver diagram with a MMS relevant change, for example

delete a diagram reference, then the generated MMS diagrams are marked **outdated**, which can be seen within the system status viewer (for example added on Control Project level). The outdated diagram state is also indicated in the diagram's status bar, see [Diagram States and Transitions](#) on page 146.



In Function Designer, if the user changes any MMS relevant data within the hardware tree after finishing the generation of MMS cross communication, for example, connect application to another controller, change an interval time, change connected tasks, or change the IP addresses, MMS CrossCommunication must be regenerated through **MMS > Analyze and Create**. The system does not inform the user that CrossCommunication must be regenerated.

Objects	Status	Time	Description
MMSDocuProject, Control Project			
Applications, Application Group			
AppIR1, Control Application			
Control Modules, Control M...			
R_MM5_ApplS1_1	!	9/30/200...	Loaded, but outdated data. Rerun MMSCommunication.
ReceiverDiagram1	●	9/29/200...	Loaded (running).
AppIS1, Control Application			
Control Modules, Control M...			
S_MM5Com_1	!	9/30/200...	Loaded, but outdated data. Rerun MMSCommunication.
SenderDiagram1	×	9/29/200...	Modified. Generate is needed.
Controllers, Controller Group			

Figure 286. System Status Viewer After Deletion of a Diagram Reference in Sender Diagram

Template Settings

It is possible to change the template settings (from e.g. A4 portrait to Letter portrait) for the diagrams which will be generated for MMS communication: This can be done by opening the Function MMS Receiver Template aspect of the MMS Receiver Diagram object in Object Type Structure (same for MMS Sender Diagram Object), see [Figure 287](#).



Other changes must not be done in the **Function MMS Receiver Template** aspect and the **Function MMS SenderTemplate** aspect, which will affect the MMS CrossCommunication generator.

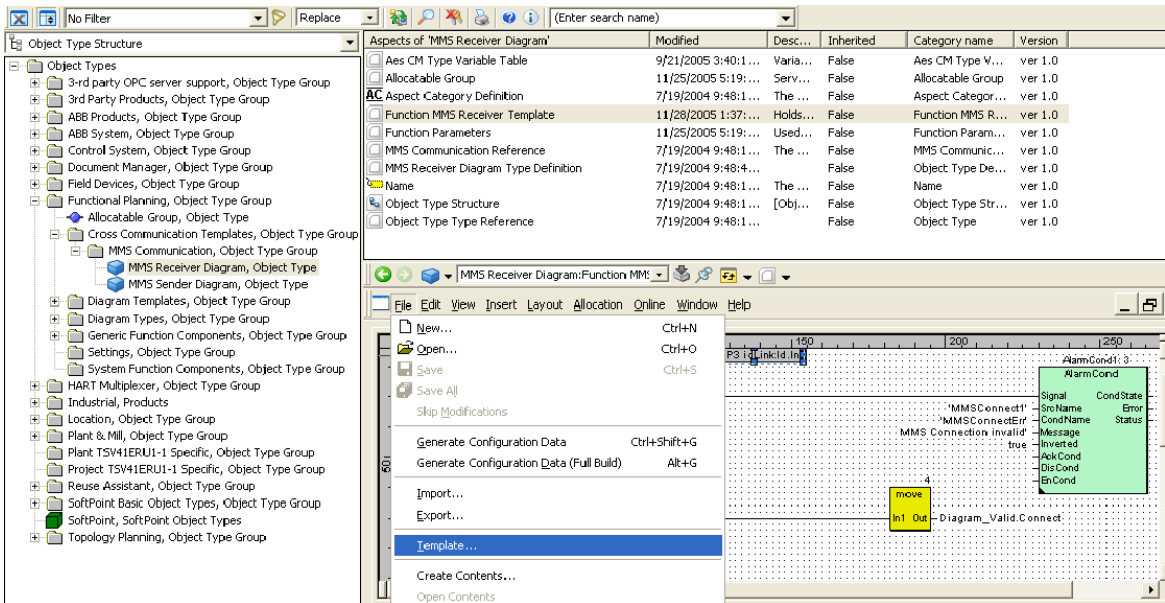


Figure 287. MMS Templates in Object Type Structure

Number of Input Ports

MMS_Read/Write Function Blocks have an extensible number of inputs. If the Function Block is created with the max. value (32), the symbol does not fit on an A4 portrait page. Add a component template with smaller fonts and less port distance to the MMS_Read/Write object types.

Control Application Allocation Change

MMS communication: There is no message when control application allocation is changed.



It is possible to download code even though the cross communication is not consistent. Create cross communication again after allocation changes.

Export / Import of MMS Generated Function Diagrams from Engineering System to Production System

The following export / import procedure assumes:

- that the dependencies of the Function Diagram are already fulfilled except for new placements to be done on the target system: All required system extensions are loaded, all user defined libraries and object types are available, the target Control Project is available and the needed libraries are connected to the target Control Application.
- that MMS Cross Communication already exists in both the Engineering system and Production system.

Follow the steps to import the additional MMS generated diagrams from Engineering system to Production system:

1. In Engineering system, click **Launch Import Export** tool.
2. Drag and drop the CrossCommunication object or MMS Receiver and MMS Sender diagrams to the **Import / Export** window.
3. Select **Include dependencies** check box in the appeared **Add Items** window.
4. Click **OK**.

The **Adding items...** window appears with a progress bar that indicates the process. The **Finished adding items** window appears when all the items have been added.

5. Click **Done** in the appeared **Finished adding items** window.
6. Click **Save** and provide a path and file name in the **Save import/export file** window.
7. Click **Done** in the appeared **Finished saving file** window.
8. In the **Import / Export - <file name>.afw** window, select **Production** from the drop-down list.
9. Click **File > Open**.

10. In the **Open import/export file** window, browse to and select the file saved in [Step 6](#).
11. Click **Done** in the appeared **Finished loading file** window.
12. Click **Differences**.
13. In the **Show Differences** dialog box, select **Production** in the **Environment** drop-down list.

Ensure that all the differences between the Engineering system and the Production system are listed.

14. Click **Import All** to import all the objects and aspects available in the Function Diagram.
15. In the **Import objects and aspects** dialog box, select **Production** in the **Environment** drop-down list.
16. Click **Finish**.

The **Importing <file path>\<file name>.afw...** window appears with a progress bar that indicates the process.

17. In the **Replace Existing Function Block Type** window, select the **Apply to all** check box and click **Yes** that appears during the import.
18. Click **Done**.



It is recommended to always perform **MMS > Analyze** to ensure that all the additional MMS generated diagrams are imported from the Engineering system to the Production system.

Multiple Allocatable Groups and Multiple Code Blocks

Mapping Aspect Objects

Each allocatable group can belong to only one Function Diagram. It is not possible to assign logic of different Function Diagrams into the same allocatable group. A Function Diagram can have nested components with its own Function Diagrams (and own allocatable group aspects). If function component objects of different Function Diagrams have to be located in the same task on the same controller, they

are allocated each of them in an own allocatable group. It is possible to allocate several allocatable groups to the same task.

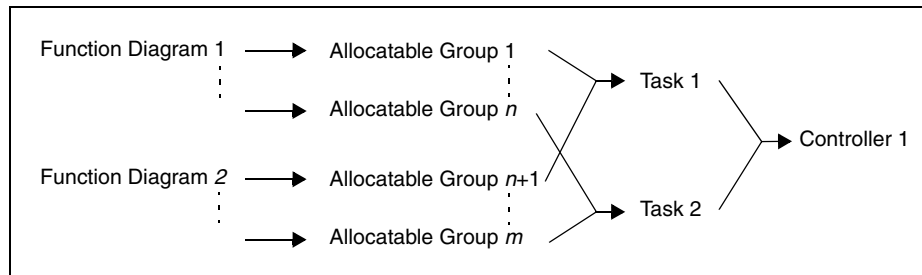


Figure 288. Mapping of Allocatable Groups

For AC 800M controllers, each allocatable group creates a single control module (SCM) or an I/O signal.

Grouping Function Components into Allocatable Groups

If the user wants to split the logic of one function diagram, in order to distribute the function components into different locations later on, it is possible to create additional allocatable groups, which will lead to generic objects with an allocatable group aspect. This allocatable group aspect will act as folder object. These folder objects appear as child objects of function diagrams in the plant explorer's Functional Structure.

The function components in a diagram can be split into several allocatable groups:

- **Splitting Function Diagram Logic to Several Allocatable Groups:**

If the user wants to split the logic in order to allocate it to several tasks, controllers or applications, user has the possibility to create allocatable groups for assigning the different parts of the function diagram to different allocatable groups.

Creation of allocatable groups means creation of generic objects having an allocatable group aspect. These generic objects appear as child objects below the object containing the function diagram and they act as folder objects.

If the control logic of a function diagram is to be assigned (split) to several allocatable groups in order to allocate the control logic parts to different tasks, applications, or controllers, do the following:

1. [Create a new allocatable group inside Function Designer](#)
2. [Grouping function components inside Function Designer](#)



From 800xA 5.1 onwards, newly created Function Diagrams do not support multiple allocatable groups.

Create a new allocatable group inside *Function Designer*

1. Open the Function Diagram whose logic shall be split to several allocatable groups.
2. Choose the menu command **Allocation > New Allocatable Group**, or the corresponding button in the Allocation toolbar. The **Create Allocatable Group** dialog is opened.
3. In the [Create Allocatable Group Dialog](#), enter a name for the new allocatable group in the **Name** field and a short name in the **Short Name** field. Optionally, you can also choose a color for indicating all function components assigned to this new allocatable group.
4. Click **OK**.

The created allocated object is located as child object below the function diagram object in Functional Structure. It can be seen as folder object to separate parts of the logic of a function diagram.

Create Allocatable Group

Use the **Create Allocatable Group** dialog to create a new, additional allocatable groups. Access this dialog via the **Allocation > New Allocatable Group** menu command, or via the Allocation toolbar.

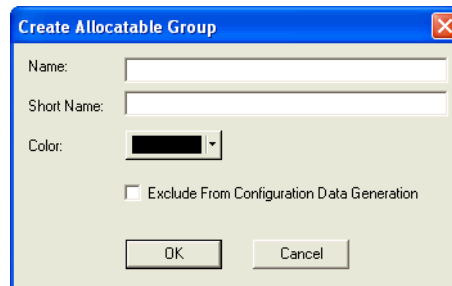


Figure 289. Create Allocatable Group Dialog

- **Name**
Enter a name for the new allocatable group object. This name equals the aspect objects Name aspect.
- **Short Name**
Enter a short name for the new allocatable group. If any, the short name gets displayed in the Allocation toolbar, in the Data Flow Order dialog, and in the diagram when showing allocation details via the document reference \$AllocGroup.
- **Color**
Click on the arrow down to open the color palette and choose the wanted color by clicking on it.



If there is not offered an appropriate color, click **Other** to open the **Color** dialog offering the whole color palette of the system. The selected color gets optionally displayed in the diagram when showing allocation details.

- **Exclude from Configuration Data Generation**
This allocatable group is to be excluded from configuration data generation.

- **OK**

Click **OK** to save your settings and to close the dialog.

- **Cancel**

Click **Cancel** to quit the dialog without saving your setting modifications.

Grouping function components inside *Function Designer*

Follow the steps to assign function components to an allocatable group inside *Function Designer*:

1. Select one or more function component(s) inside the function diagram.
2. Choose main menu item **Allocation > Allocation Grouping** or the component **Allocation Grouping**.
3. The dialog **Destination Group** is opened. Select an allocatable group and click **OK**.

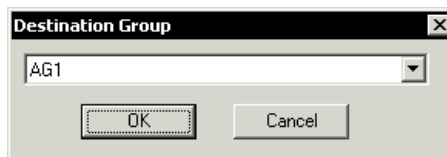


Figure 290. Destination Group Dialog

4. Selected function component aspect objects get automatically moved in Functional Structure and located as child objects of the corresponding allocatable group, see [Figure 291](#).

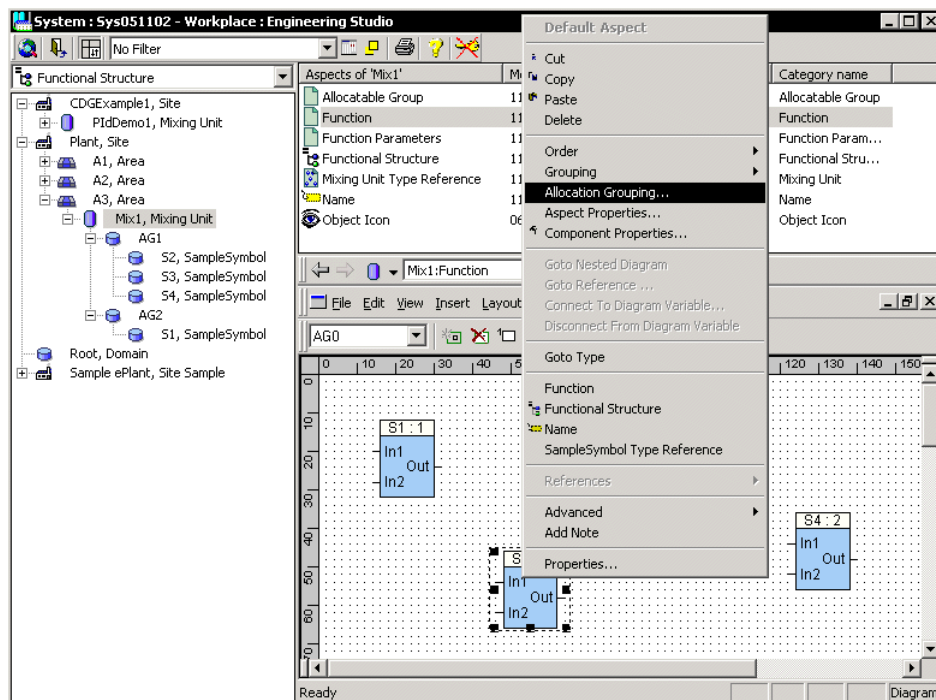


Figure 291. Group into Allocatable Groups by Context Menu

Assigning Function Components of Different Function Diagrams to One Task

An allocatable group belongs to one function diagram only. Therefore, it is not possible to group function components belonging to different function diagrams into the same allocatable group folder in order to assign them to the same task in the same application.

If you want to get such a kind of allocation, you have to create separate allocatable groups within all these function diagrams and allocate all of them later on to the

same task. The following figures displays an example for wrong and correct grouping of function components:

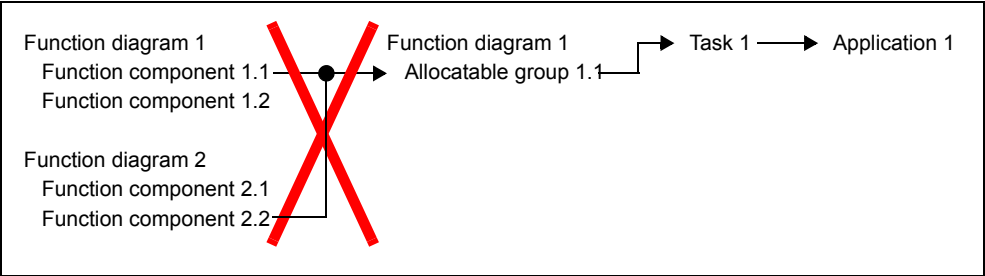


Figure 292. Wrong Grouping of Function Components

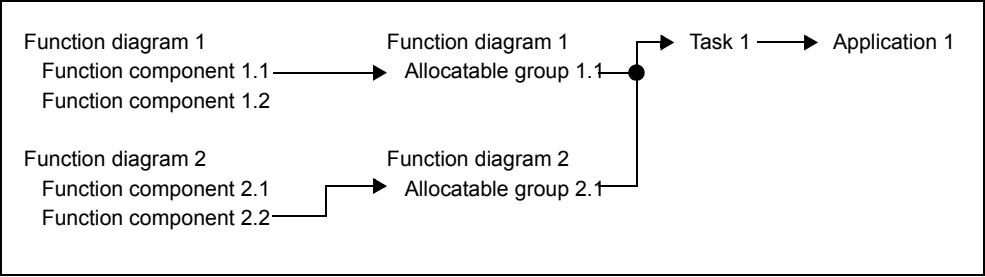


Figure 293. Correct Grouping of Function Components

Grouping Child Function Diagrams

Figure 294 shows an example of a function diagram, whose function component **Flow Control** represents a child function diagram. The child function diagram is shown in Figure 295:

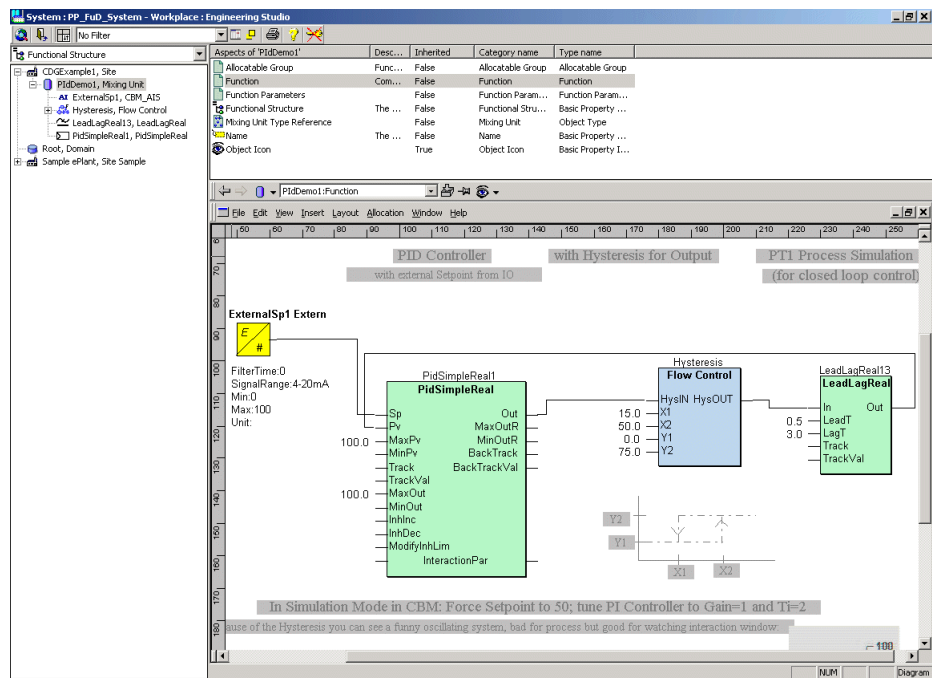


Figure 294. Function Diagram

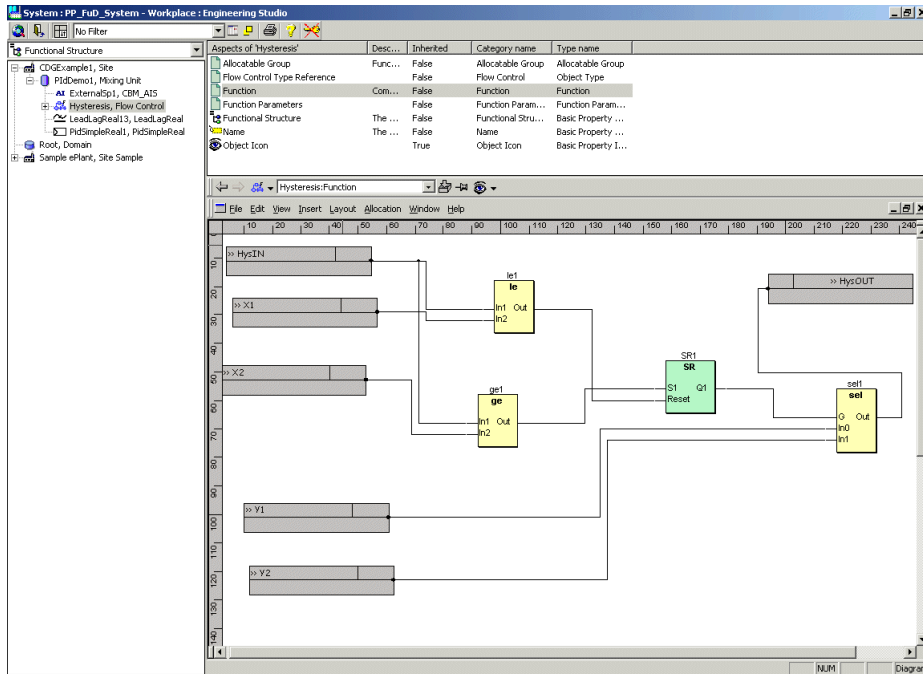


Figure 295. Child Function Diagram

See [Allocating Child Function Diagrams](#) on page 605.

Copying Function Components and Allocatable Groups

Note the following features on copying function components (Aspect Objects and/or symbol objects):

- If you copy and paste a function component Aspect Object inside the Functional Structure and the parent of the target object has a function diagram, this function component is automatically inserted into this function diagram
- If you copy and paste a complete allocatable group, you will get a new allocatable group with all child function components (Aspect Objects and symbol objects). These objects are automatically inserted into the corresponding function diagram.

- If you copy and paste function components inside a function diagram (Diagram view), the function components are assigned to the active allocatable group (see [Figure 296](#))

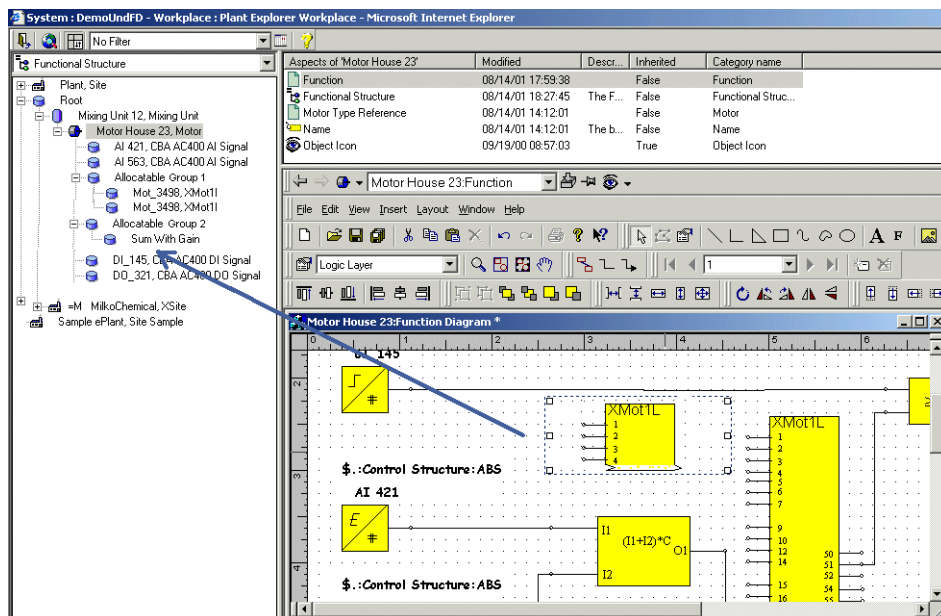


Figure 296. Copying Inside Function Designer

Code Block Generation

Function Designer generates at least one code block for each Function Diagram. In case of Function Diagrams with Control Modules, the code blocks are created as explained below:

1. Based on the assigned data flow order, add the first function / function block to a code block and include additional functions / function blocks until the occurrence of a Control Module.
2. Create a new code block for the Control Module.

Table 61. Code Block Splitting Details

Data Flow Order	Function Block / Control Module	Code Block	Reason
6	Control Module B	Code Block D	Data flow order 6 is assigned to a Control Module (Control Module B). Hence, a new Code Block is created.
7	Function E	Code Block E	
8	Function Block C	Code Block E	
9	Function Block D	Code Block E	

Function Designer generates a new aspect property named **NewCodeBlock** for every element which could generate a new code block.

Creation of New Code Blocks

In Diagram view, choose **Allocation > Data Flow Order** to open the **Data Flow Order in Allocatable Group** dialog. For Function Diagrams created prior to 800xA 5.1 and for diagrams restored to 800xA 5.1, right-click the required component and select **New Code Block**. Provide a **Code Block Name** in the **Code Block** dialog, and click **OK**. The selected component is assigned to the new code block.

Edit Existing Code Blocks

Right-click the component which is already assigned to a code block and select **Edit Code Block**. Edit the **Code Block Name** in the **Code Block** dialog, and click **OK**. The selected component is assigned to the newly named code block.

Delete Existing Code Block

Right-click the component which is already assigned to a code block and select **Delete Code Block** to delete the required code block.

Code Block Sorting

To get an optimal execution order, code blocks created by the Function Designer are sorted by the Control Builder compiler while downloading them to the controller.

For more details on code block sorting, refer to *System 800xA Control 5.1 AC 800M Planning (3BSE043732*)*.

Loops in Control Modules

The compiler sorts the execution order of code blocks based on the data flow between code blocks. Circular data flow between code blocks form a loop.

When a loop is detected between a set of code blocks, those code blocks are sorted in a random order. Downloading such randomly sorted code blocks to the controller along with the loop may cause undesired execution behavior in the controller.

Compiler Switches

To detect the loops before downloading them to the controller, set the compiler switch **Loops in Control Modules** to either **Warning** or **Error** for all control projects. This provides a text file with the information of the code blocks that form the loop.

If the compiler switch is set to:

- **Warning** - control projects are downloaded to the controller even when a loop is detected.
- **Error** - control projects are not downloaded to the controller when a loop is detected.

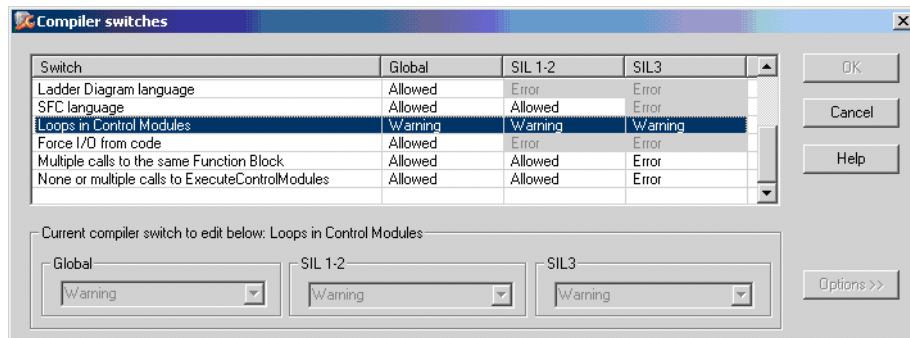


Figure 298. Compiler Switches

Resolving Loops

The following scenarios can cause loops in Single Control Modules:

- Two code blocks read and write values to the same variable.
- Circular data flow between code blocks.
- User defined Control Module type is created in Control Builder and a port (For example, Out port) is configured to read and write from a connected Control Module.

Scenario 1: Two Code Blocks Read and Write Values to the Same Variable

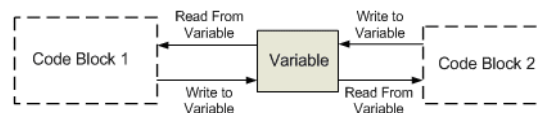


Figure 299. Two Code Blocks Read and Write to the Same Variable

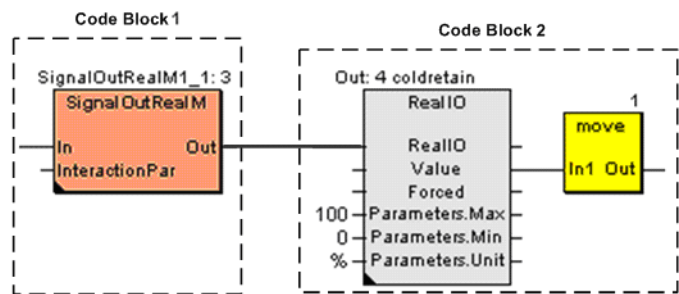


Figure 300. Two Code Blocks Read and Write to the Same Variable

In [Figure 300](#), code block 1 reads Max, Min, and Unit from the variable and writes Value to the variable in code block 2. Code block 2 writes Max, Min, and Unit to variable and reads Value from the variable. A loop is created due to the circular data flow between the code blocks, as shown in [Figure 301](#).



Variables do not belong to any code blocks.



Figure 301. Loops Created from the Circular Dataflow between Code Blocks

To resolve this loop, create a new code block from the Move block as shown in the [Figure 302](#):

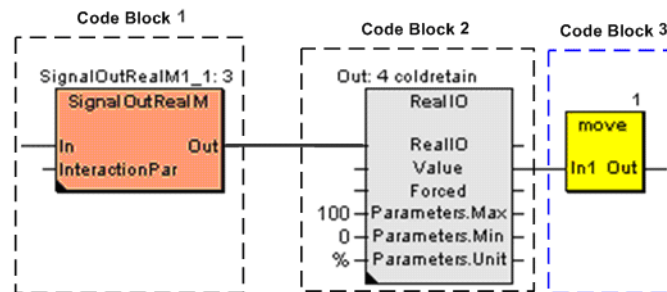


Figure 302. Resolving the Loop

Follow the steps to create a new code block:

1. Launch the **Data Flow Order in Allocatable Group** dialog window.
2. Right-click the respective block (**Move** block) from the list and select **New Code Block**.
3. Enter the new code block name.
4. Click **OK**.
5. Click **Apply**.

The newly created code block resolves the loop formation as the read-write operation is now handled by two different code blocks, as indicated in [Figure 303](#).

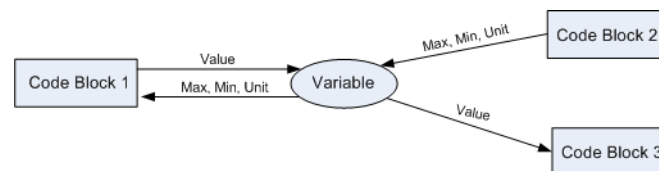


Figure 303. New Code Block Resolves the Loop Formation

Scenario 2: Circular dataflow between codeblocks



Figure 304. Circular Dataflow

In [Figure 305](#), circular dataflow occurs between code block 1 and code block 2. In this scenario, the code blocks are formed due to the presence of the AnalogInCC control module between the **OR** and **TOn** blocks.

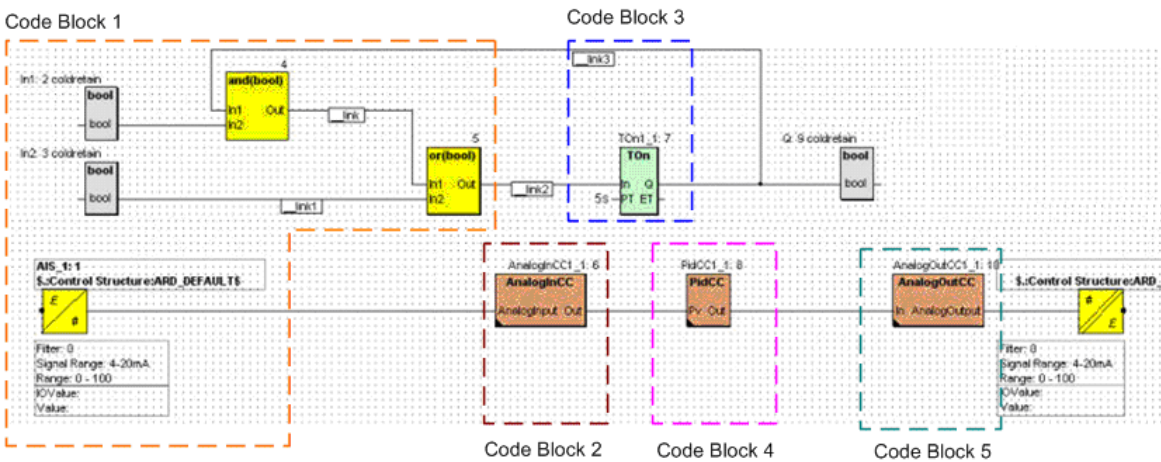


Figure 305. Circular Dataflow Between CodeBlocks

This can be resolved by moving the AnalogInCC control module towards right (as shown in [Figure 306](#)) and generating the data flow order. This change in logic combines code block 1 and code block 3 (refer to [Figure 306](#)) and eliminates the loop in the logic.

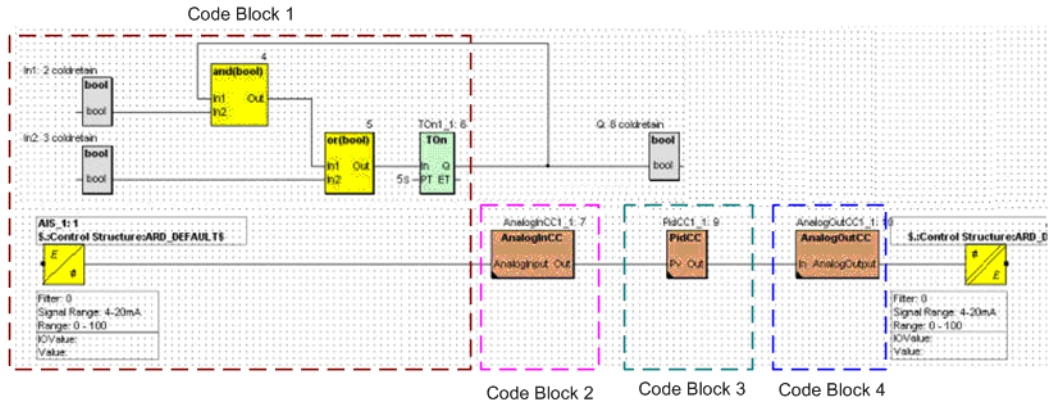


Figure 306. Resolving Circular Dataflow Between Code Blocks

Scenario 3: Forward and Backward Communication between Control Modules

In some instances, user defined control modules are configured to write and read from another control module through the same port, as indicated in [Figure 307](#). Each control module forms a code block. This also creates a loop in the logic.

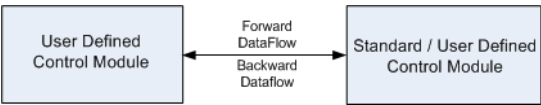


Figure 307. Forward and Backward Communication between Control Modules

It is always recommended to avoid such configurations in user defined control modules, which may create forward and backward data flow between connected control modules.

Showing Allocation Details

User can view and print allocation details and data flow order through function diagram.


Border color and short name of an allocatable group can be set through the Color and Short Name property of the allocatable group aspect.



The allocatable group details and data flow order of a function component can be displayed by any text or label component using the field codes:

- \$AllocDetails
- \$AllocGroup
- \$Order

To define allocation details

1. Open the required function diagram.
2. Choose the menu command **Allocation > Allocate Diagram** or click the  icon of the Allocation toolbar. The allocation details are shown in the function diagram.

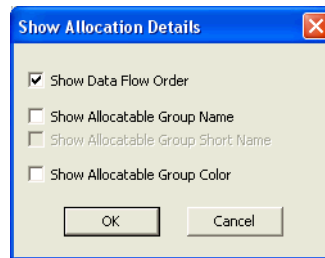


Figure 308. Showing Allocation Details

Vertical Navigation

For a better overview on Function Diagrams, it is possible to build a hierarchical Functional Structure with parent-child diagrams. Functions on a lower level can be summarized in a separate diagram and displayed in an overview diagram on a higher level as a single function component with input and output ports / parameters. Diagrams on lower level are also called child or nested diagrams. They can be opened from an overview diagram by choosing **View > Goto Child Diagram** or **Goto Child Diagram** from the context menu. This is also called vertical navigation - top down, bottom up, throughout the functional hierarchy.

Inputs and outputs of child diagrams are function components of type [Object Type Structure]Object Types/Functional Planning/Generic Function Components/Connectors/Diagram Parameters/**Input (Output, Inout) Parameter**:

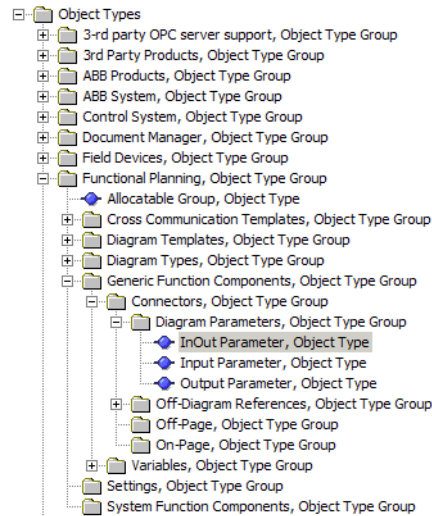


Figure 309. Diagram Parameters in Object Type Structure

User can apply diagram parameters to:

- child diagrams in Functional Structure, see [Figure 310](#) and
- diagram types in Object Type Structure.
For more information on diagram type, refer to *System 800xA Engineering, Engineering Studio Function Designer Getting Started (3BDS100968*)*.

In [Figure 310](#), an overview diagram D0 and two child diagrams D1 and D2 are available. The diagram parameters D1:In1, D1:In2, D1:Out and D2:In, D2:Out

- are accessible as aspect properties in diagram D0 and
- are visible as ports in overview diagram D0.

Thus they can be connected in the overview diagram as any other ports. Diagram Parameter connections are visible in the overview diagram, not in the child diagrams.

If there is no component view defined for a child diagram, a symbol is automatically created on base of the default component template. However you can also define your own symbol for child diagrams:

1. Create a Component view for the child diagram to be customized.
2. Select [File > New](#), and New Component.
3. Draw your customized symbol with corresponding ports.
4. Name the ports exactly as the diagram parameters.



On configuration data generation, the diagram parameter description is copied to the corresponding control module / single control module parameter.

User can apply any Control Builder parameter keyword in the diagram parameter description as IN, OUT, IN(OUT), OUT(IN), NODE, or EDIT. Control Builder expects that control module:

- input parameter starts with description 'IN' or 'IN(OUT)'
- output parameter starts with description 'OUT' or 'OUT(IN)'
- inout parameter starts with description 'NODE' or 'EDIT' or any string other than a keyword.



The diagram parameter description gets displayed as tooltip at the corresponding port of the child diagram.



User can define **initial values** at the **Value.InitVal** aspect property of the diagram parameter.

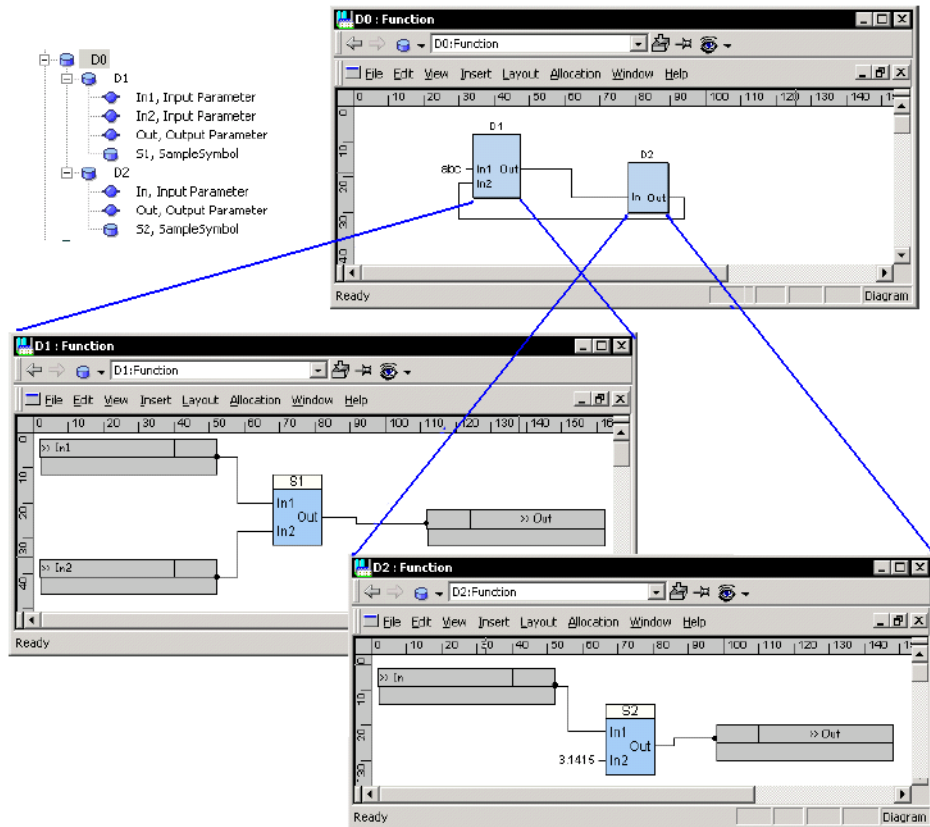


Figure 310. Child Diagrams



In Horizontal navigation, user can combine diagram references and diagram parameters in one diagram.

Variable Creation

In *Control Builder M*, variables are used for connections within child diagrams (vertical navigation). If there are connections across child diagrams / nested allocatable groups, the Diagram parameter objects are used and the following variables will be used:

Table 62. Diagram Parameter⁽¹⁾

Target ⇨ (outside AG) Source ⇩	Diagram Parameter
FB	Parent-SCM:Var SCM: Parameter
CM	Parent-SCM:Var SCM: Parameter
FUN	Parent-SCM:Var SCM: Parameter

(1) The abbreviations are described in Table 12

Table 63. Diagram Parameter⁽¹⁾

Target ⇨ (outside AG) Source ⇩	FB	CM	FUN
Diagram Parameter	Parent- SCM:Var SCM: Parameter	Parent- SCM:Var SCM: Parameter	Parent- SCM:Var SCM: Parameter

(1) The abbreviations are described in Table 12

Explicitly and Implicitly Defined Variables

Data Flow Order of variable/parameter/reference. The combo box **Data Flow Order Insert Before** lists all function components of the current allocatable group.

Select the function component before which the variable/parameter/ reference shall be located. Select **at the end of the Allocatable Group** if you want the variable/parameter/ reference to be located at the end of the current allocatable group. The combo box **Data Flow Order Insert Before** is visible at creation time of the variable/parameter/reference only. If you want to change the data flow order of the variable/parameter/reference later use the **Data Flow Order** dialog.



The Data Flow Order of variable/parameter/reference influences the order of execution of MOVE functions implicitly inserted into a Function Diagram and generated into corresponding code blocks by Function Designer in cases as:

- Mapping of structured variables of different data types.
- Connection of variables to other variables (diagram references etc).
- Connection to the "old" values in own code block, if using the state attribute.

Labelling Split Connections

If the logic of a function diagram is assigned to several allocatable groups, so that a connection link between two or more function components is split, a communication object is needed in *Control Builder M* to hold the connection. This communication object will be a variable (output of the allocatable group) corresponding to single control module parameters of *Control Builder M*. The variable is created automatically on allocating the allocatable groups to the Control Structure respectively to *Control Builder M*.

The name of the variable depends on whether the connection line in the function diagram has a label or not:

- If you have given a unique label to the connection line (see [Inserting Label Components](#) on page 379), the variable will get the name of that label.
- If you have not given a label to the connection line a unique default label will be created automatically by the *Function Designer* itself. Then, the name is composed of: “__”+“Out”+[ordered number] (for example, “__Out1”)

[Figure 311](#) shows an example for splitting a connection line between two directly connected function components grouped into two different allocatable groups.

On allocating the allocatable groups **Allocatable Group1/2**, for the split connection line **Label1**, a variable with the same name **Label1** is created connecting the single control module (SCM) parameters **__Out1** and **__In1**:

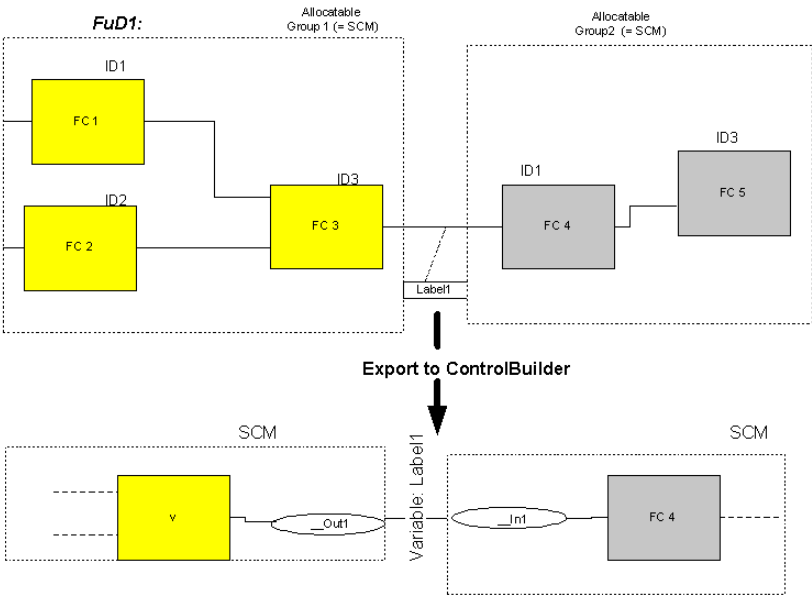


Figure 311. Splitting of Connected Function Components

Building up Control Structure

The Control Structure gets automatically updated, see [Table 64](#).

Table 64. Changes in Control Structure

Change in Function Diagram or Functional Structure	Change in Control Structure
Instantiate/copy/delete function or procedure	none (SCM gets updated on configuration data generation)
Instantiate/copy/delete function block or control module as symbol object	none (SCM gets updated on configuration data generation)
Instantiate/copy function block or control module as aspect object	Function block or control module gets inserted below diagram / allocatable group (SCM)
Delete function block or control module aspect object in diagram	Function block or control module gets deleted from diagram / allocatable group (SCM)
Delete function block or control module aspect object in Engineering Workplace Advanced Menu Delete from All Structures	Function block or control module gets deleted from diagram / allocatable group (SCM)
Delete function block or control module aspect object in Plant Explorer object browser	none. You need to manually remove the instance from the Control Structure.
Instantiate/copy/delete CBM IO signal	none (SCM/application get updated on configuration data generation)
Instantiate/copy/delete diagram parameter	none (SCM gets updated on configuration data generation)
Instantiate/copy/delete diagram reference and connect to diagram variable	none (SCM/application get updated on configuration data generation)
Instantiate/copy/delete local variable	none (SCM gets updated on configuration data generation)
Copy/paste diagram objects	Creates a copy of diagram and put it into the same CBM application. (IO signals need to be reallocated to appropriate hardware units)

Table 64. Changes in Control Structure (Continued)

Change in Function Diagram or Functional Structure	Change in Control Structure
Creates a copy of Functional Structure with same parent	Copy/paste diagram objects (IO signals need to be reallocated to appropriate hardware units)
Create copies of related diagrams with same parents	Copy/paste application (IO signals need to be reallocated to appropriate hardware units)
Delete diagram object	Diagram object gets deleted from Control Structure and CBM (and vice versa)
Allocate diagram / allocatable group	Diagram / allocatable group (SCM) gets moved from unallocated application to corresponding CB application
Generate configuration data	Nested (still unallocated) children get moved to corresponding CB application
Unallocate diagram / allocatable group	Diagram / allocatable group (SCM) gets moved from CB application to unallocated application

Allocating Allocatable Groups

After having defined the data flow order within the allocatable groups, you can allocate the function diagram logic to a specific application. The following requirements must be fulfilled:

- *Control Builder M* (CBM) has to run
- The corresponding CBM project has to be opened

The insertion of allocatable groups from the Functional Structure into the Control Structure can be done through the Allocate button in the Allocatable group's aspect page. User can also use BDM.

1. Open the Functional Structure and navigate to the allocatable group to be allocated.
2. Open its allocatable group aspect.
3. Select the **Allocate** button.

4. The **Allocate** dialog is opened.

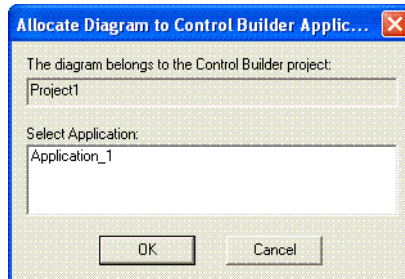


Figure 312. Allocate to Control Builder Application in Control Structure

The allocation of the allocatable groups initiates the following actions:

- Control Structure aspect and Single Control Module aspect is added to the allocatable group object.
- Control modules and function blocks are allocated, i.e. moved from the CBM unallocated application to the selected application.
- Same holds for child function diagrams.

Depending on the allocatable groups defined in the Functional Structure and depending on the intended allocation within the Control Structure, one of the following allocation variants is possible:

- [Allocation of a Whole Function Diagram](#)
- [Allocation of Parts of a Function Diagram](#)

Allocation of a Whole Function Diagram

If no extra allocatable groups have been created for a function diagram in the Functional Structure, the allocatable group contains the whole function diagram.

The allocated diagram gets a single control module aspect in the *Control Builder M*.

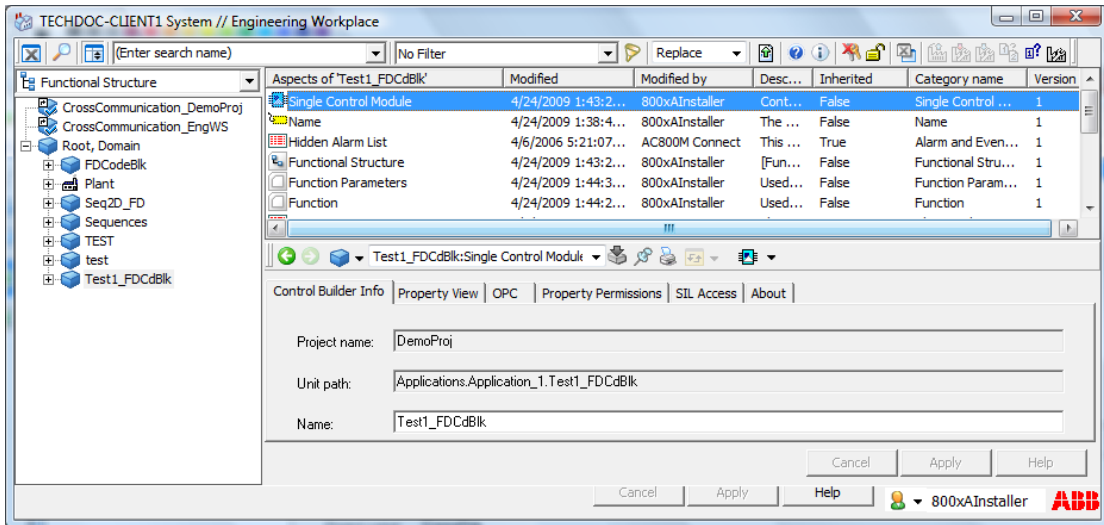


Figure 313. Function Designer Generated SCM in Functional Structure

Allocation of Parts of a Function Diagram

If extra allocatable groups have been created for a function diagram in the Functional Structure (each allocatable group contains a part of the function diagram), you have two possibilities to allocate the allocatable groups to applications

- You allocate the allocatable groups into different control applications. Then, you will get a single control module for each allocatable group object (see [Figure 314, left](#)).
- You allocate the top level object containing the function diagram to a control application. You will get a single control module for the top level object and additionally single control modules for each allocatable group object (see [Figure 314, right](#)).

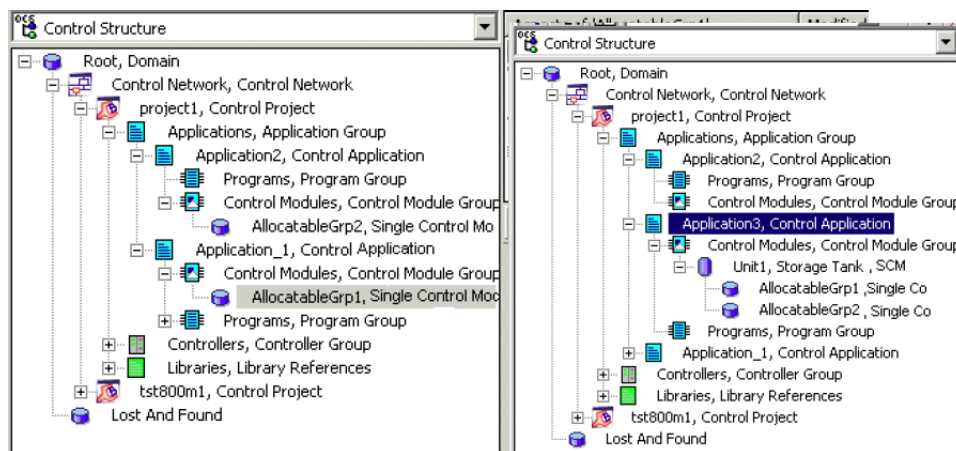


Figure 314. Several Allocatable Groups in Control Structure

Allocating Child Function Diagrams

It is possible to nest function diagrams within other function diagrams. Nesting means that a function component of a function diagram contains an own (child) function diagram (see function components FC2 and FC6 in [Figure 315](#)).

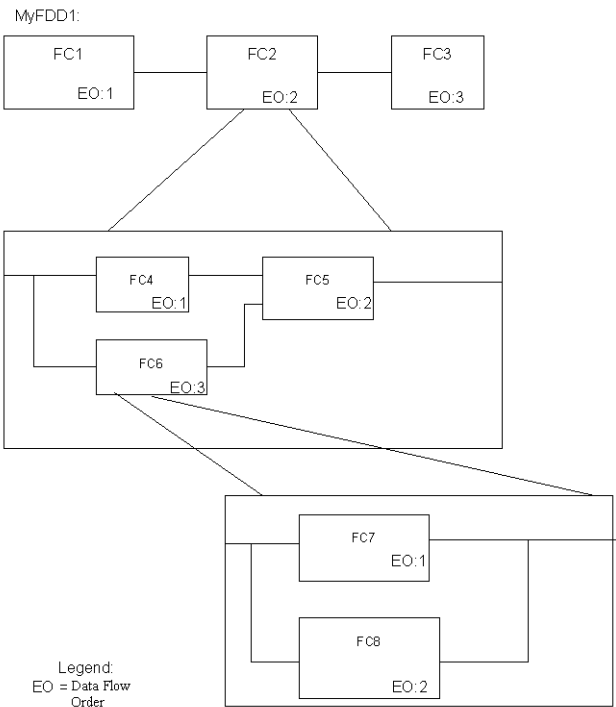


Figure 315. Child Function Diagrams

The whole logic of a function diagram (inclusive the logic of all child function diagrams) has to be allocated to the same application. Only the function diagram on the highest level has to be allocated (see MyFDD containing the function

components FC1, FC2, FC3 in [Figure 316](#)).

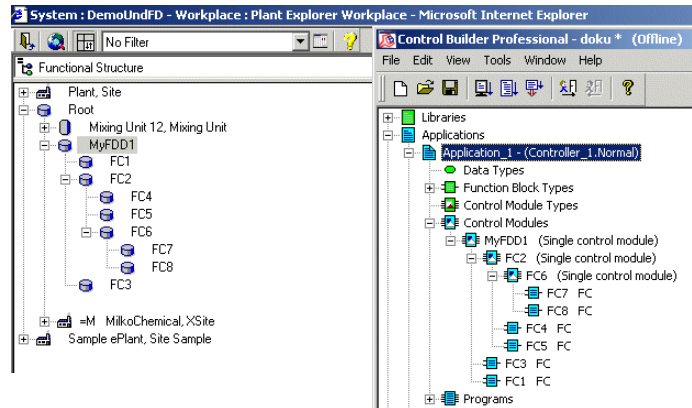


Figure 316. Child Function Diagrams Assigned to one Allocatable Group (Within Functional Structure and Control Builder M)

Every function component representing a nested function diagram becomes a nested single control module within *Control Builder M*.

Connections between Allocatable Groups inside same Application

Connections between allocatable groups always means we have to deal with two single control modules, one for each allocatable group; on both single control modules parameters have to be created: Parameter for source single control module S_SCM and target T_SCM and connect these single control module parameters with a global variable on application level.

If you have connections between allocatable groups, the following variables will be used:

Table 65. Connections Between Allocatable Groups Inside Same Application⁽¹⁾

Target ⇨ (outside AG) Source ⇩	FB	CM	FUN
FB	Appl: GlobVar S_SCM: ExtVar T_SCM: ExtVar	Appl: GlobVar S_SCM: ExtVar T_SCM: ExtVar	Appl: GlobVar S_SCM: ExtVar T_SCM: ExtVar
CM	Appl: GlobVar S_SCM: ExtVar T_SCM: ExtVar	Appl: GlobVar S_SCM: ExtVar T_SCM: ExtVar	Appl: GlobVar S_SCM: ExtVar T_SCM: ExtVar
FUN	Appl: GlobVar S_SCM: ExtVar T_SCM: ExtVar	Appl: GlobVar S_SCM: ExtVar T_SCM: ExtVar	Appl: GlobVar S_SCM: ExtVar T_SCM: ExtVar

(1) If there are more than one action listed inside a cell, all of them are created.

Referencing Devices

HART Devices

After a function diagram as shown in [Figure 111](#) has been IO allocated and configuration data generated the IO variable created for the HART device can be referenced in another (or the same) function diagram:

- 1. Create a diagram reference of the data type given by the HART device.
- 2. Connect this reference to a variable by entering exactly the name of the IO variable created for this HART device. The variable you can look up in the control application, it is named according to the Name aspect or according to the Control Builder Name aspect.

3. Automatically the yellow part will show the control designation of the hardware channel.

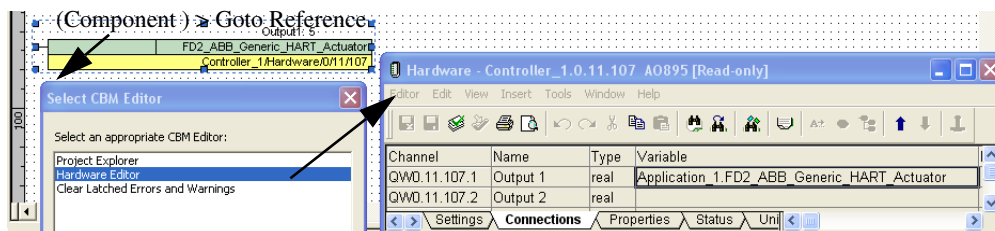


Figure 317. Diagram Reference to an Existing HART Device I/O Variable

The connect to diagram variable part of the creation of diagram references can create a not yet existing application global variable. Therefore you can create the IO variable before doing device insertion, IO allocation and corresponding configuration data generation.

1. Create a diagram reference of the data type given by the HART device.
2. Connect it to a variable (this here means creation of this variable) named exactly according to the device to be inserted later.
3. Perform configuration data generation.
4. Later on: Insert the corresponding device with the already chosen name.
5. Perform IO allocation and corresponding configuration data generation. The existing variable is then identified via the Name or Control Builder Name aspect of the HART device.

As soon as the configuration data generation is done the control designation of the hardware channel will show up in the corresponding diagram reference symbol(s).

Synchronization between Control Builder Name aspect and Name aspect

Function Designer extensions enable synchronization between **Name** aspect and **Control Builder Name** aspect of an objects in 800xA System. In case engineering is not carried out using Function Designer, synchronization can be disabled by creating a new function setting.

The procedure to create new function setting is as follow:

1. Open **Engineering Workplace** and navigate to **Object Type Structure -> Function Planning -> Settings**, see [Figure 318](#).

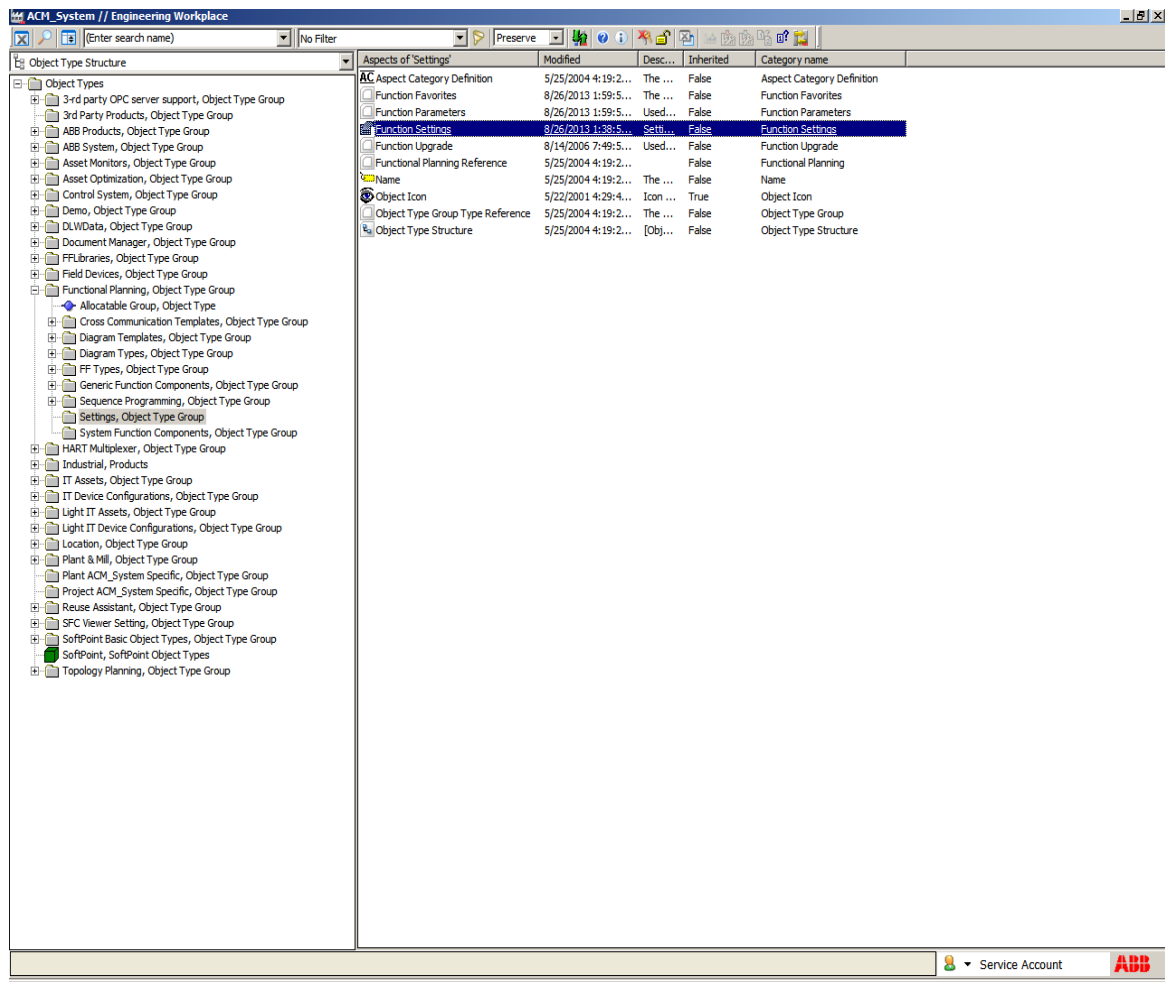


Figure 318. Settings in Object Type Structure

2. Right-click on **Function Settings** and select **Config View**.

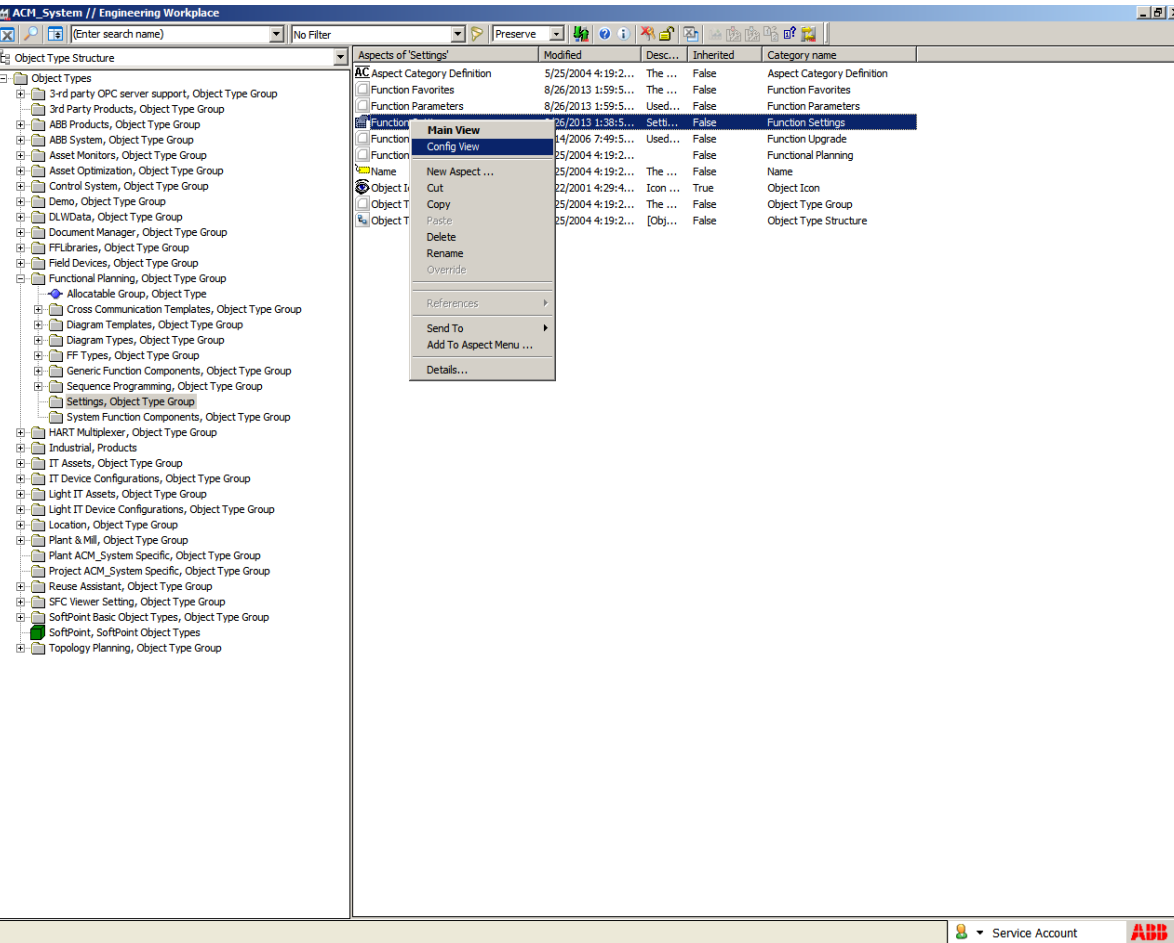


Figure 319. Function Settings

3. **Function Settings** dialog opens.

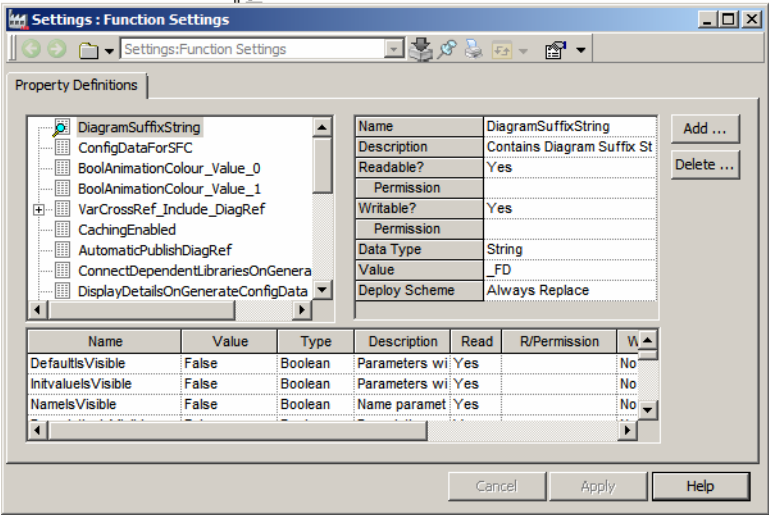


Figure 320. Function Settings Config View

4. Click **Add** and select **Property After**.

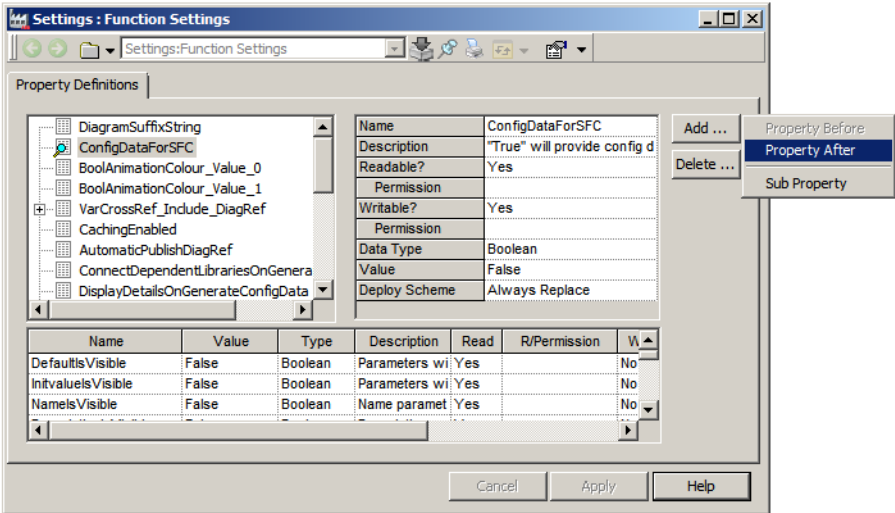


Figure 321. ADD Dialog

5. The following dialog is seen.

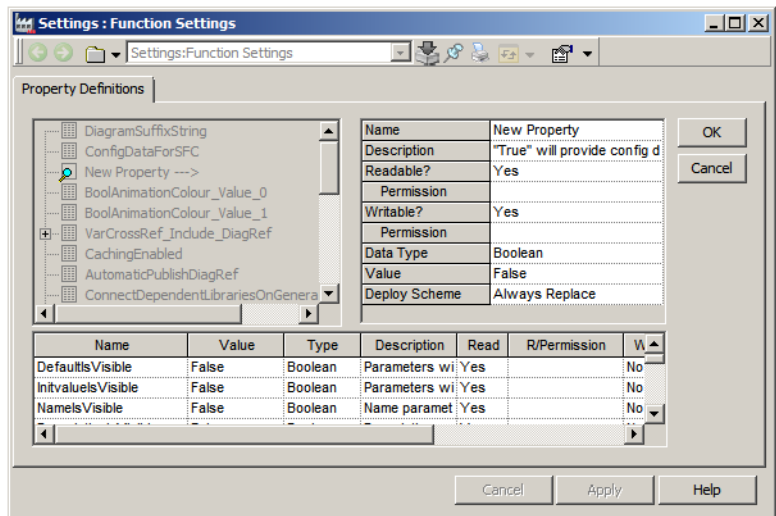


Figure 322. Add New Property Dialog

6. Enter the **Name**, **Readable permission**, **Writable permission**, **Value**, **Data Type** and **Deploy Scheme** as shown in the figure below.

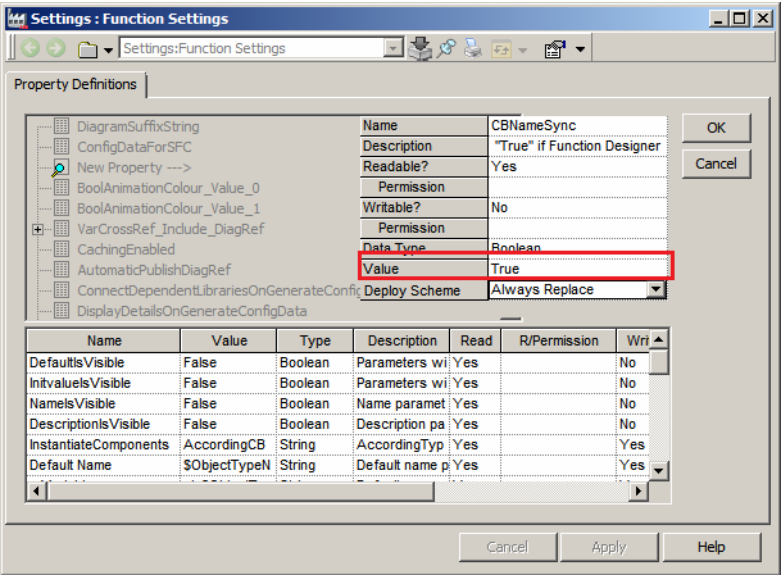


Figure 323. Details Addition Dialog

7. Click **Ok**. A new property CNameSync is added.

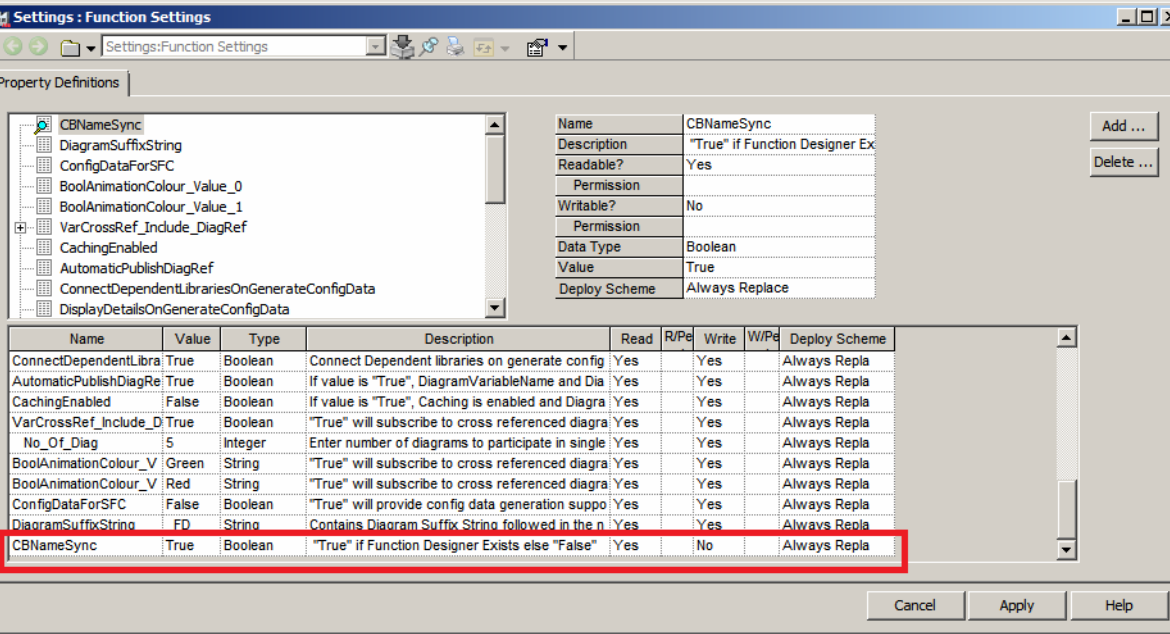


Figure 324. CNameSync Dialog

- Click **Cancel** to revert the newly created setting.
- Click **Help** to refer help at any point during the operation.
- Click **Apply** to apply the newly created setting to the system.
- Restart **Engineering Workplace** for the settings to take effect.



Synchronization can be disabled by changing the setting **Value** to **False** from the **Main View** of the Function Settings aspect.

Symbols

.DataType 46
.Direction 46
.In 46
.Initval 46
.InitvalOfObjectType 46
.Inverted 46
.Out 46
.Visible 46, 83
\$? 444
\$AllocDetails 444, 594
\$AllocGroup 444, 579, 594
\$Date 444
\$GroupName 444
\$LastPage 445
\$Name 445
\$NextPage 445
\$OPCFORCED 445
\$OPCIOStatus 445
\$OPCIOValue 445
\$OPCValue 445
\$Order 445, 594
\$Page 445
\$PGComment 445
\$RefPage 445
\$RefPort 445
\$RefSymbol 445
\$SymbolName 445
\$Time 445

A

AC 800M/C controller 112
AC800M Connect 112
accelerator keys 370
AccordingCB 68

AccordingCBAlarmOwnerFlag 68
AccordingTypeDefCreateInfo 68 to 69
ActiveX Control 434
ActiveX Run Mode 303, 435
add
 aspect property 46
Administrative Document Properties 537
AES Error Context aspect 248
AlarmOwner 68
align 313, 340
 components 411 to 412, 454, 459
Allocatable Group
 Color 579
 Exclude from Configuration Data
 Generation 579
 Short Name 579
 Show Allocation Details 594
allocatable group 28, 288, 444
allocate
 diagram to AC800M/C 126, 602
 I/O Signals 127
allocation
 child diagrams 605
 details 316, 593
 in Control Structure 204
 of diagram 604
 toolbar 340
Alt key 375, 394
 left click 371
 left drag 371
Alt+Ctrl key 375, 394
Alt+Shift key 375, 394
Alt+Shift+Ctrl key 375, 394
ambient properties 331, 429
anchor component 394

- Application Engineer User Role 111
- Arranging Windows 277
- AsAspectObject 69
- aspect
 - default 78
 - parameters 36
 - properties 36
- aspect command Generate Configuration Data 247
- aspect command Generate Configuration Data (Full Build) 247
- Aspect Control 73
- Aspect Object Mapping 28, 576
- Aspect Property 43, 46, 84, 297 to 298, 331
- Aspect Property Reference 49, 65, 443, 445, 457
- aspect ratio 375
- Aspect Verb Synchronize Names 156
- AspectObject 44
- assign
 - components to task 581
- AsSymbolObject 69
- Audit Trail 110
- Authority 111
- Authorouting page 497

B

- Backup/restore 108
- BDM option Subtree enabled 237
- BDM option Transaction ?y Object? 238
- BDM option Transaction ?ata Area? 238
- BDM options 237
- BoolIO 188 to 189, 511
- Bulk Data Manager 232
- Bulk Data Manager Allocate Diagrams 243
- Bulk Data Manager Copy Diagram(s) 234, 237
- Bulk Data Manager Diagram Status 249
- Bulk Data Manager Generate Configuration Data 247
- Bulk Data Manager Generate Diagram(s) 259
- Bulk Documentation 251
- bulk print 487 to 488

C

- Casted Data Types 510
- Casts 509
- Change Type 80
- Checks
 - Connections 138
 - Constants 139
 - Inversions 138
- Checks for Connections 138
- Checks for Inversions 138
- circle port 380
- Color
 - Background 431, 447
 - Foreground 446
 - Marked Port 432
 - Selection 432
 - Transparent Background 447
- color
 - dialog 460
 - page 431
 - user-defined 471
- command
 - group 315
 - ungroup 315
- Complete Contents 487
- Complete contents 251
- component
 - document 276 to 277, 330
 - graphic 297, 373
 - group 351
 - moving 351
 - pages 276
 - properties 297
 - types 27
 - view 274
- Component Properties 429, 449, 464
- Component Properties dialog 446
- Component View
 - Context Menu 329
- Component view 274

- ComponentHidden 44
 - Components Dialog 400, 463
 - connect
 - by keyboard 90
 - by mouse 92
 - string 84
 - to diagram variable 99, 335
 - connection
 - across diagrams 135
 - between allocatable groups 607
 - between blocks 134
 - from I/O channels 137
 - to I/O channels 136
 - Consistency 144
 - Contents Chapter 251
 - contents diagram 251, 487
 - Contents Template 488
 - contents template 255
 - context menu 325
 - dockable window 343
 - Object Aspect 274
 - output window 342
 - toolbar 339
 - Control Builder Name 154
 - control module
 - user-defined 26
 - Control Properties 144
 - ControlBuilderDescription 45
 - ControlBuilderName 45
 - ControlConnection 189, 511
 - copy
 - aspect 328
 - components 376, 420, 426, 584
 - document 330
 - range 424
 - reference 287
 - template 287, 480, 482
 - tool buttons 349
 - copy and paste components 419
 - Copy Diagram(s) via Bulk Data Manager 234, 237
 - Copy to All Instances 73
 - create
 - allocatable group 578
 - component 374
 - connections 380
 - graphic component 374
 - toolbar 366
 - variables 133, 598
 - Create Contents dialog 251
 - Create Info options 69
 - CrossCommunication 44
 - CrossCommunication.CycleTime 44
 - CrossCommunicationValid 44
 - Ctrl key 375, 394
 - cursor 371
 - left click 371
 - left drag 371
 - Customize dialog for toolbars 344
 - cut
 - components 419
 - document 330
 - cut and paste components 418
- D**
- Data Flow Order 116
 - data flow order 316
 - DataType 42
 - default aspect 78, 330, 334
 - default properties 331, 465
 - default properties dialog 450
 - DefaultIsVisible 29
 - DefaultName 29
 - define
 - component properties 429
 - function components 50
 - delete 334
 - active page 301
 - aspect 328
 - Aspect Property 48
 - components 427, 463

- connections 372
 - page 305, 483
 - ports 372
 - symbols 372
 - toolbar 347
 - user-defined toolbar 369
 - Delete from All Structures 124, 601
 - DescriptionIsVisible 29
 - Detailed contents 255
 - Device Integration 201
 - diagram
 - child 306
 - context menu 325
 - document 276, 330
 - nested 127, 301 to 302
 - pages 276
 - parent 306
 - reference 101
 - referenced 306
 - variable 101
 - view 274
 - diagram page size 256
 - Diagram Parameter 118, 595
 - Diagram Reference 96
 - Diagram States 146
 - Diagram Types 149
 - Diagram view 275
 - Diagram View Context Menu 329
 - DiagramIsBad 45
 - DiagramSize.Height 45
 - DiagramSize.Unit 45
 - DiagramSize.Width 45
 - dialog
 - color 460
 - Component Properties 439
 - components 463
 - page 465
 - DintIO 189, 511
 - disconnect
 - Diagram Variable 335
 - from diagram variable 105
 - port 93
 - Display List 190
 - Display of On-line Values 185, 187
 - dockable
 - windows 278
 - dockable windows 278
 - Dockable Windows Context Menu 343
 - Document Properties, List of 537
 - Document Reference 444, 457
 - Documentation 251
 - drag and drop 80, 274, 277, 279, 366, 368, 374 to 375, 384, 389, 409, 418, 420, 423
 - drawing
 - page size 298
 - scale 298, 476
 - Drawing Scale 476
 - DTM 203
 - DwordIO 189, 511
- ## E
- Edit > Measurements and Size 256
 - Edit page 440, 467
 - Electronic signature 108
 - Engineering Studio Workplace 273, 423
 - Error Messages 505
 - Explicit Object Type 70
 - export file 500
 - External Links
 - Input 425
 - Output 426
- ## F
- Field Code 443, 457
 - Field page 443, 455
 - Fieldbus Builder 201
 - Fieldbus Management 203
 - file
 - check 291 to 292
 - create contents 290

- export 289, 500
- import 289, 501
- open contents 290
- template 290
- File > Generate Configuration Data 284
- File > Generate Configuration Data (Full Build) 285
- File > Open Contents 252
- File > Page Setup 256
- File > Print 257
- File > Template 256
- File Menu 282
- Fill page 446, 467
- fit to sheet(s) 257
- flip
 - around the horizontal 312
 - around the vertical 312
 - components 414
 - horizontal 314
 - vertical 314
- focus component 395
- Font page 447, 461, 468
- font properties 465
- function block
 - user-defined 26
- Function Component
 - context menu 333
- Function Designer for AC800M 112
- Function Diagram Contents Template aspect 255
- Function Settings 29

G

- General Accelerator Keys 370
- General page 438, 449
- graphic component 298, 437, 465
 - context menu 336
- graphic element 383
- grid 300, 331, 350
 - properties 331, 407
- group

- box 459
- components 417, 581

H

- HART 26, 200
- HSI 26
- HWStatus 189, 511

I

- IEC 61131-3 naming conventions 154
- Import Export
 - Dependencies 263
- import file 501
- indicator ?ead? 303
- inherit
 - properties 73
- Inherit to All Instances 73
- initial value 596
- Initial values 121
- InitVal 144
- InitValIsVisible 29
- InitValOfObjectType 144
- insert
 - ActiveX 310
 - ActiveX Control 381, 383
 - circle port 309
 - closed curve 310
 - component 371, 374, 376, 384
 - ellipse 310
 - Fieldbus Builder 203
 - function component 275
 - graphic component 298, 374
 - graphic element 310, 383
 - graphic objects 350
 - graphic primitive 377
 - image 380
 - label 310
 - label components 378
 - line port 309
 - page 305, 310, 482

- picture 310, 380
- polycurve 310
- polygon 310
- polyline 310
- ports 380
- rectangle 310
- single line 310
- symbol 387
- text 309
- text component 378
- via drag and drop 384
- via menu command 386

InstantiateAsAspectObject 68

InstantiateComponents 29, 68 to 69

InstantiatedByDesignerVersion 45

integration AC 800M/C 112

Internal Links 424

Inversion 93

L

label 451

- insert 310
- orientation 450
- page 450
- properties 465

layer 350

- annotation 470
- Annotation Layer 305
- background 470
- logic 470
- master page 287, 305, 470, 480 to 481
- properties 304, 470
- set 301

Layout

- Align Bottom 313
- Align Center 313
- Align Left 312
- Align Middle 313
- Align Right 313
- Align Top 313

- Flip 314
- Group 315
- Make Same Site 313
- of components 407
- Order 314
- Rotate 314
- Space Evenly 313

Line page 453, 469

line port 380

Local variables 118

M

master page 301

- layer 480 to 481
- template 479, 481

Master Page Client Area 436

Master Page Template 479, 481

measurement unit 476

Measurements 298, 331, 475 to 476

Measurements and Size 474

Menu

- Allocation 315, 317
- bar 278
- Context 325
- Edit 293
- File 282
- Help 324
- Insert 308
- Layout 311, 350
- Online 317
- View 299
- Window 323

menu bar 281

Module 476

move

- components 409
- object 371
- symbols 371
- via drag and drop 409

multiple selection 370, 393, 395, 405, 429

N

Name Synchronization 155
 Name Uploader 155
 NameIsVisible 29
 naming rules 29
 navigate 291
 Goto page 305
 page 301, 484
 to page 305
 navigation
 horizontal 95
 vertical 95, 594
 navigation toolbar 274
 Nested Diagram 95, 594
 Network 84, 87, 512, 514
 network traversal 372
 New Component Name 153, 497
 Nudge 372
 Number of Input 298
 NumberOfInputs 44, 81

O

Object Type Definition Aspect 69
 Object Type Definition aspect 72
 Object Type Definition Extension aspect 72
 object types
 define 286
 of Fieldbus Builder 201
 ObjectDescription 44, 499
 ObjectName 44, 499
 ObjectTypeGroup 44
 ObjectTypeName 44
 OffPageConnector 436
 Online
 Add to Display List 190 to 191
 Remove from Display List 190 to 191
 Watch Window 192
 Online > Remove all Links from Display List 191
 Online > Subscribe for Live Data 191
 On-line Display 184

Online Menu 317
 Online Optimized 188
 OPC 48
 bad quality 186
 Forced value 186
 open
 aspect dialog 373
 Open Diagram/Component Documents 276
 Open New Windows 276
 Options dialog 493
 order components 414
 output window 140
 Override 75
 override components 328
 Overview
 workflow 113
 Overview Contents 487
 Overview contents 251, 255

P

page
 ActiveX Controls 434
 Autorouting 497
 colors 431
 drawing 473
 Drawing Scale 474
 edit 440
 field 443, 455
 fill 446
 font 447, 461, 468
 general 449
 label 469
 labels 450
 line 453, 469
 navigation 350, 484
 Off Page Connector 436
 paste 496
 Position and Size 454
 print 495
 setup 291, 479

- text 457
- page comments 253
- Pan
 - diagram 372
 - functionality 392
- Parent Diagram view 275
- paste
 - components 419 to 420, 584
 - dokument 330 to 331
 - links 496
 - page 496
- Paste Rename Functionality 172
- Permissions 111
- port
 - disconnect 93
 - input 25
 - invert 93
 - multiple selection 401
 - output 25
- Port Direction 41
- Position and Size page 454
- Position.Page 45
- Position.Unit 45
- Position.X 45
- Position.Y 45
- Preview
 - window 273
- Preview window 273
- print
 - allocation details 593
 - Component 291
 - Diagram 291
 - page 495
 - preview 291
- Print Components on Layers 490
- Printer and Printer Page Settings 490
- printer page size 256
- Product
 - scope 20
- PROFIBUS 26

- Project constants 84
- properties 63, 65, 297, 339
 - ambient 294, 298
 - component 440
 - components 437
 - default 298, 465
 - font 465
 - graphical 294
 - label 465
- property reference 48
- Property Reference dialog 49
- Property References 48
- Publish via Parent Diagram 233

R

- Read indicator 111, 303
- Read Only 441
- RealIO 188 to 189, 511
- redo command 293, 503
- reference
 - Goto 106
 - template 480 to 481
- remove aspect property 47
- Rename Substructure 156
- rotate components 413

S

- Scope 20
- scrollbar
 - horizontal 342
 - vertical 341
- Security Definition Aspect 111
- select
 - all connections 407
 - components 393
 - connections 394, 404
 - contiguous components 395
 - contiguous connections 405
 - multiple pages 393
 - ports 394, 400 to 401, 403

- single connections 404
- single ports 400
- single symbols 396
- symbols 394, 396, 398, 400
- text 395
- toolbar 398
- set
 - properties 464
- setup page 479
- SHIFT
 - left click 370
- Shift key 375, 394
 - cursor key 370
 - left click 370
 - left drag 370
- Shift+Ctrl key 375, 394
 - left click 371
 - left drag 371
- show / hide 455
- show allocation details 594
- Show/Hide Ports 82
- Signal Extension for AC800M Connect 112
- size
 - component 374, 390, 411, 441, 454
 - font 448, 465, 476
 - grid 304
 - page 454, 473 to 474, 477
 - paper 300, 304, 479
 - properties 440
 - symbols 370
 - text 460
- Space key
 - cursor 371
- splitt connection 599
- Splitting Windows 277
- Structured Data Types 511
- Subscribe for Life Data 306
- Subscribe for Live Data 457
- support
 - Bulk Data Manager 234

- SVG 500
- symbol object 70
- symbol traversal 372
- Synchronize Names 156
- system extension 112
- System Status Viewer aspect 250

T

- Template 479, 481
- Template dialog 479
- Text page 457
- tool buttons 349
- toolbar
 - add tool buttons 366
 - align 355
 - allocation 351, 357, 578
 - area 339
 - categories 349
 - conection 351
 - connection 84, 90, 93, 358
 - create user-defined 366
 - delete 369
 - drawing 382, 438
 - hide 364
 - layout 351, 357
 - list box 346
 - look of 346
 - name 346
 - nudge 350, 356
 - online 358
 - page 345, 350, 355
 - remove tool buttons 368
 - rename user-defined 368
 - rotate 350, 413 to 414
 - selection 351, 399, 403, 406
 - sequence 359
 - show 364
 - standard 352
 - structure 356
 - user-defined 347

- view 354, 392 to 393
- working with toolbar 344
- Toolbar Commands 349
- toolbars
 - name 346
- TypeOfInputs 44

U

- Unallocated Application 243
- undo command 293, 503
- Undo/Redo 503
- ungroup
 - command 315
 - components 417
- User Log-Over 111
- User Roles 111

V

- Value.InitVal 596
- Variable Property 298
- VariableAttributes 44
- VariableDataType 44
- VariableDataTypeGroup 44
- View
 - allocation details 593
 - Component and Diagram 274
 - Diagram 275 to 276
 - grid 303
 - grid properties 304
 - layers 304
 - pages 305
 - Printable Area 304
 - Snap to Grid 303
 - Status Bar 303
 - zoom 304

W

- Watch Window 192
- Watch window 192
- window

- Component document 275
 - overlap 344
 - preview 344
- Windows registry 33
- Workspace 33

Z

- zoom 331, 390, 392, 493
- zooming 300, 392

Revision History

This section provides information on the revision history of this User Manual.



The revision index of this User Manual is not related to the 800xA 5.1 System Revision.

The following table lists the revision history of this User Manual.

Revision Index	Description	Date
-	First version published for 800xA 5.1	June 2010
A	Published for 800xA 5.1 Rev A	May 2011
B	Published for 800xA 5.1 Rev B	June 2012
C	Published for 800xA 5.1 FP3	August 2012
D	Published for 800xA 5.1 Rev C	November 2012
E	Published for 800xA 5.1 FP4	February 2013
F	Updated for 800xA 5.1 Rev D	December 2013
G	Updated for 800xA 5.1 Rev E/FP4 Rev E	April 2015

Updates in Revision Index A

The following table shows the updates made in this User Manual for 800xA 5.1 Rev A.

Updated Section/Sub-section	Description of Update
Section 2, Configuration	Under <i>Horizontal Navigation</i> subsection, communication line indicator is added.
Section 2, Configuration	Under <i>Preference Setting</i> subsection, two new settings added in Function Settings
Section 2, Configuration	Under <i>Global Variable and Communication Variables</i> subsection, Restore Function Diagram is added.
Section 2, Configuration	Under <i>Connect to Diagram Variables</i> subsection, a new data bit value is added.
Section 2, Watch window Grid	Added a new information note on communication variables.
Appendix A, Single Control Modules	Added a new section <i>Creation of an SCM based Function Diagram</i>
Appendix A, MMS Communication Overview	Limit of the length of MMS cross communication message is added.
Appendix A, MMS Communication Overview	Added procedure to delete the MMS generated diagrams.
Appendix A, Export/Import of MMS Generated Function Diagrams	New steps added to import the additional MMS generated diagrams from Engineering systems to Production system.
Appendix A, Grouping Functional Components into Allocatable Groups	Added new information is <i>Code Block Generation</i> subsection.

Updates in Revision Index B

The following table shows the updates made in this User Manual for 800xA 5.1 Rev B.

Updated Section/Sub-section	Description of Update
Section 2, Configuration	Under <i>Horizontal Navigation</i> subsection, information regarding empty reference variable added.
Appendix A, MMS Communication Overview	Updated information regarding recommendations for optimal MMS communication.

Updates in Revision Index C

The following table shows the updates made in this User Manual for 800xA 5.1 FP3.

Updated Section/Sub-section	Description of Update
Section 2, Configuration	Under <i>Field Device Integration</i> subsection, IO support for Foundation Fieldbus is added.
Section 2, Configuration	Under <i>Field Device Integration</i> subsection, IO support for Profinet Devices is added.
Section 2, Configuration	Under <i>Field Device Integration</i> subsection, IO support for IEC61850 Devices is added.

Updates in Revision Index D

The following table shows the updates made in this User Manual for 800xA 5.1 RevC.

Updated Section/Sub-section	Description of Update
Section 2, Configuration	Under <i>System Function</i> subsection, logover information is deleted.

Updates in Revision Index E

The following table shows the updates made in this User Manual for 800xA 5.1 FP4.

Updated Section/Sub-section	Description of Update
Section 2, Configuration	New topic “Handling Cascaded Loop Back Signals” is added.
Section 4, Error Messages	New section added.

Updates in Revision Index F

The following table shows the updates made in this User Manual for 800xA 5.1 Rev D.

Updated Section/Sub-section	Description of Update
Section 3, Default Properties	Information added for the Font property.
Section 3, Print	Print dialog box is changed and there is a bullet point description for ‘Print as one job’ added. A tip added for Bulk Printing.
Section 2, Network Connection between Pages	Information about the network error added.
Appendix A	Synchronization between Control Builder Name aspect and Name aspect.
Appendix A, MMS Communication overview	Information about the MMS error added.
Section 2, User Interface	Subscription for live data functionalities information updated.
Section 2, Configuration	Conversion of Global Variables to Communication Variables.

Updated Section/Sub-section	Description of Update
Section 2, PROFIBUS Devices	Added workaround for importing .gsd file under preparation of PROFIBUS Devices.
Section 2, Configuration	Added Paste Rename functionality.
Section 2, User Interface	Information on MOVE blocks added.
Section 3, Print	A tip added for executing print operation of Function Diagrams.

Updates in Revision Index G

The following table shows the updates made in this User Manual for 800xA 5.1Rev E/FP4 Rev E.

Updated Section/Sub-section	Description of Update
Appendix A, Create Differences	Deleted the following information: <i>Whenever a new receiver application is mapped to an existing sender application which is already involved in cross communication across applications, MMS > Create Differences does not establish the cross communication to the new receiver application. Alternatively, perform MMS > Clear Generated Data and MMS > Analyze and Create</i>
Section 2 Configuration, Change Type; Subscribe to Live Data functionalities	The following information has been added: <i>Subscribe for Live Data is not supported for Extensible Parameters.</i>

Updated Section/Sub-section	Description of Update
Section 2 Configuration, Generate Configuration Data	<p>The following information has been added:</p> <p><i>Before performing the Generate Configuration Data for Function Diagrams, set ConnectLibsOnGenerateConfigData property to False, so that the library version of Function Diagram is not replaced by latest ones during the system update. After performing the Generation of Configuration Data for Function Diagrams, change ConnectLibsOnGenerateConfigData property to True.</i></p>
Section 3 Graphic Editor Reference, Changing the Location of Docked Windows/Toolbars	<p>The following information has been added:</p> <p><i>The menu bar cannot be docked back by drag and drop action. Double-click the menu bar to dock it back.</i></p>
Section 2 Configuration, Preparation of PROFIBUS Devices	<p>Deleted the following information:</p> <p>If there are modules starting with DP in the .gsd file, perform the workaround before importing the .gsd file of Hardware Library into Function Designer.</p> <ol style="list-style-type: none"> 1. Open .gsd file in a Notepad. 2. Search for module names starting with DP. 3. Replace the letters DP with some other name. 4. Save the file. 4. Import the .gsd file from Hardware Library. <p>The signal allocation using IO Allocation tool of Function Designer for the GSD Library will not have issues after this</p>

Contact us

www.abb.com/800xA
www.abb.com/controlsystems

Copyright© 2015 ABB.
All rights reserved.

3BDS011224-510 G

Power and productivity
for a better world™

