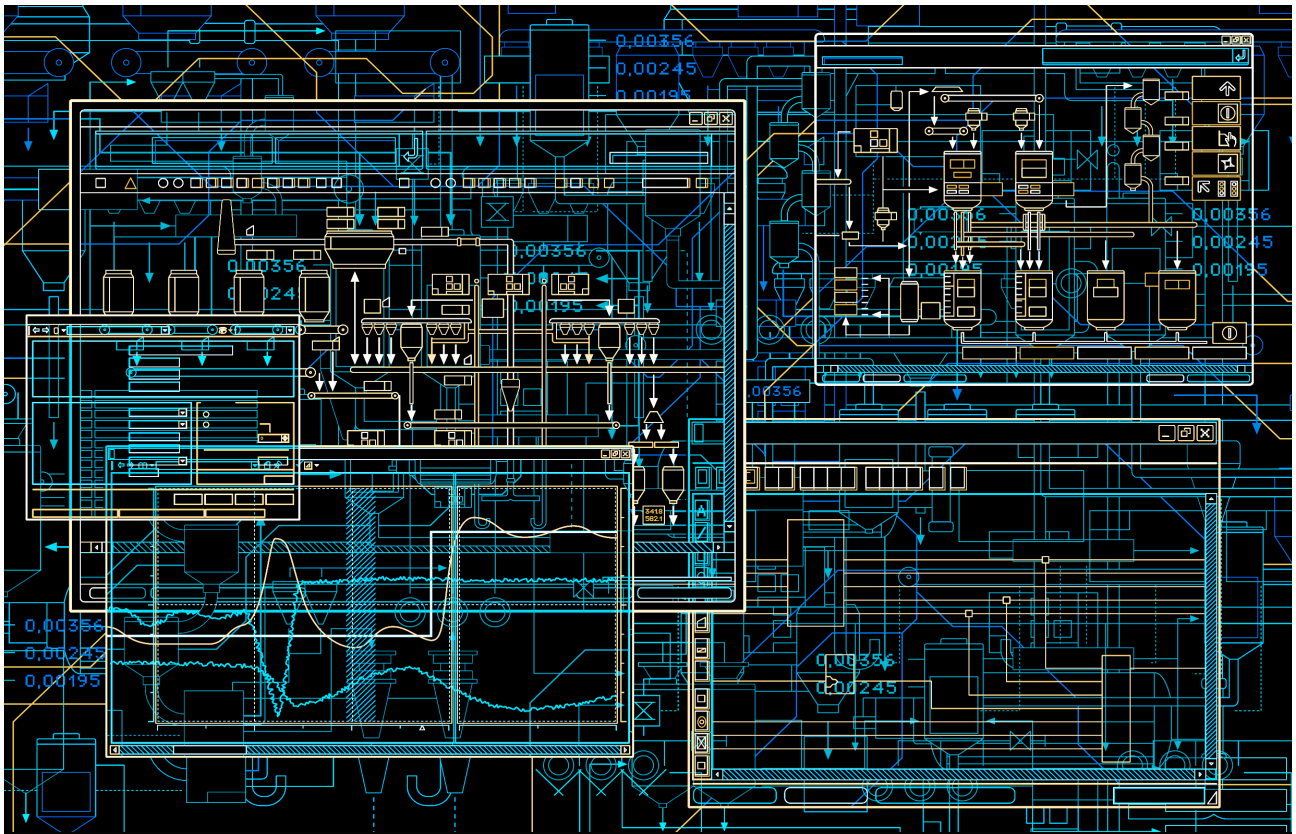


System 800xA Engineering

Engineering Studio

System Version 5.1

Power and productivity
for a better world™

System 800xA Engineering

Engineering Studio

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.

Release: July 2015
Document number: 3BDS011223-510 F

TABLE OF CONTENTS

General

Section 1 - Introduction

Section 2 - Engineering Workplace

Section 3 - Bulk Data Manager

Section 4 - Bulk SPL

Section 5 - IO Allocation

Section 6 - Parameter Manager

Section 7 - Document Manager

Section 8 - Reuse Assistant

Section 9 - Script Manager

Appendix C - Document Aspect Properties

Appendix D - Parameter Categories Examples

Appendix E - Word Report Templates

Appendix F - Reuse Assistant Architect - Reference

General

General Information

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 serves as a reference document for the different engineering functions provided with the Engineering Workplace of the System 800xA.

The Engineering Workplace covers the following major features:

- Engineering Base, a System Extension providing a special Plant Explorer configuration - the Engineering Workplace- with advanced engineering menu items. See Section 2, Engineering Workplace.
- Bulk Data Manager, a tool for configuring Aspect Objects in bulk based on Microsoft® Excel file formats. See Section 3, Bulk Data Manager.
- IO Allocation, a tool to allocate engineered signals and HART devices to channels of I/O boards and to create variable connections for PROFIBUS devices. See Section 5, IO Allocation.
- Script Manager Basic, a tool allowing to run Script Aspects of Aspect Objects running Visual Basic Scripts. See Section 9, Script Manager.
- Parameter Manager, a tool to handle Parameter Aspects of Aspect Objects. See Section 6, Parameter Manager.

- Document Manager, a tool to handle documents as Document Aspects of Aspect Objects. See Section 7, Document Manager.
- Reuse Assistant, a tool to design and instantiate Reuse Instructions. See Section 8, Reuse Assistant.

Configuration of the Control System with Function Designer and Control Builder M Professional in the Engineering Workplace is described in separate instructions.

Topology Designer / Topology Status Viewer is also described in a separate instruction.

Information pertaining to the system is also available in corresponding release notes and product bulletins.

Feature Pack Functionality

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 .

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

Section 1 Introduction

Engineering Workplace of System 800xA provides access to the user for the engineering tools offered by Engineering Studio:

- Extensions to Plant Explorer Workplace.
- IO Allocation.
- Bulk Data Manager.
- Document & Parameter Manager.
- Script Manager.
- Reuse Assistant.
- Function Designer.
- Topology Designer.

Function Designer and Topology Designer are described in individual instructions:

- *System 800xA Engineering, Engineering Studio Function Designer Getting Started (3BDS100968*)*
- *System 800xA Engineering, Engineering Studio Function Designer (3BDS011224*)*
- *System 800xA Engineering, Engineering Studio Topology Designer (3BDS011225*).*

Engineering Workplace also supports Control Builder M Professional, Graphics Builder, and other engineering tools.

Product Scope

Engineering Studio functionalities allow the user to increase the efficiency in engineering, configuration, documentation, and maintenance of System 800xA applications.

If environment support is enabled in a system Engineering Workplace and the engineering tools provided by Engineering Studio can be used in Engineering environment and in Production environment.

Engineering Studio is installed according to the instructions available in the *System 800xA, Installation (3BSE034678*).*

Product Verification

User can verify the product version through **Control Panel > Programs > Programs and Features** in the windows classic start menu. 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 1.

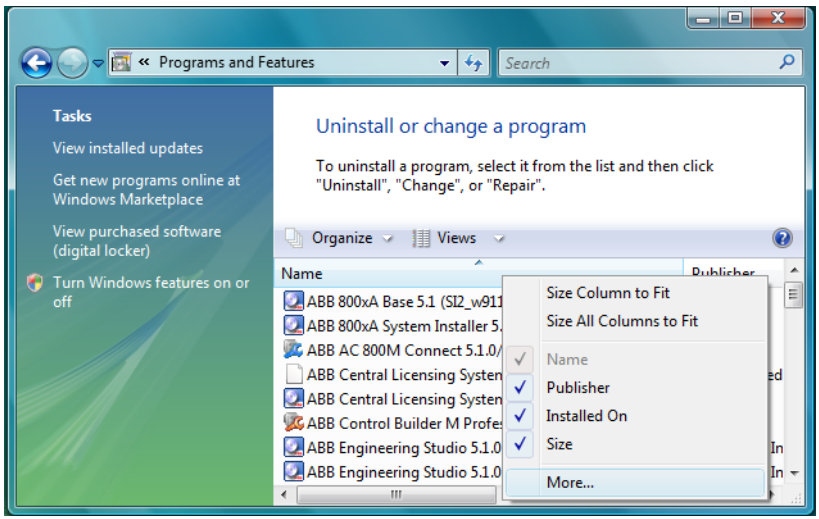


Figure 1. Context Menu to Access More Information

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 Link**, **Support Telephone**, **Help Link**, etc.

User can also verify the product version from the Plant Explorer through **Help > About Industrial**$^{\mathbf{IT}}$.

Further, version of the specific product can be verified through its **Help > About** menu command. For example, version details of Function Designer can be verified through its **Help > About** menu command from the Diagram or Component view. The version details are displayed as shown in Figure 2.

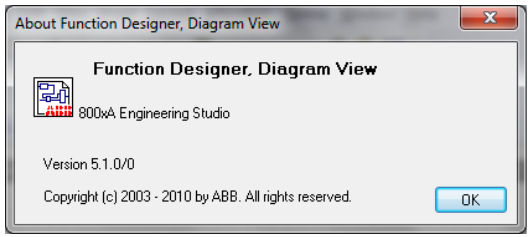


Figure 2. About Function Designer, Diagram View Window

Reporting Problems

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

Section 2 Engineering Workplace

This section describes how to use the Engineering Workplace and Configuration Wizard. Some typical scenarios such as handling of reference designations, loading system extensions, and application start-ups are described in this section.

Engineering Workplace Functions

The Engineering Workplace offers the following functionalities:

- Aspect Object context menu extensions.
- Reference designations.
- Aspect category called *Object Category.*
- Individual user identity properties.

Before You Start

Within the Engineering Studio product, few start-up settings and configurations such as those listed below are described:

- Setting User Identity Properties.
- Specifying Rules for Absolute Reference Designations.
- Specifying Separators.
- Displaying Absolute Reference Designations.
- Function Designer extensions enable synchronization between Name aspect and Control Builder Name aspect. If Function Designer is not used for engineering, synchronization can be disabled by creating a new function setting. Refer *System 800xA Engineering, Engineering Studio Function Designer (3BDS011224*), Appendix A*.

Preference Settings

To work with the Engineering Workplace a project database must be setup and operative. This database application can be started by two methods:

1. Using the Windows service - If the **Autostart System on Windows startup** check box shown in Figure 3 is selected, the system is automatically started when Windows operating system is started.
2. Using the **Start System** command - The Configuration Wizard provides the following dialog to start a system automatically. This dialog appears during the installation of the 800xA system.

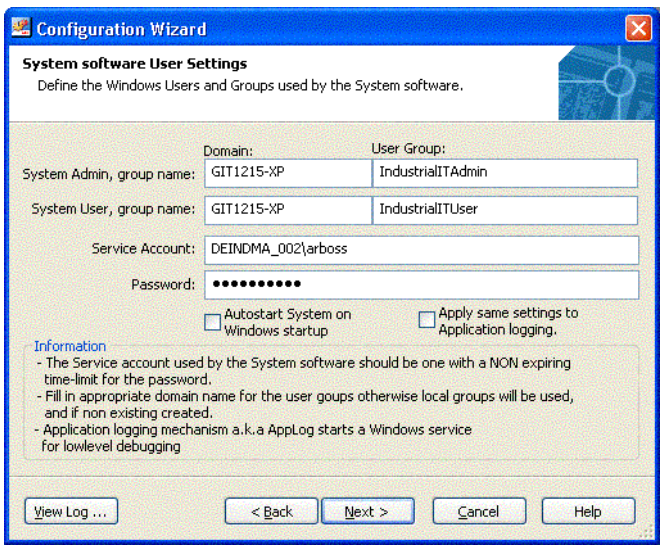


Figure 3. System Software User Settings Dialog.

Regional and Language Options

Independent of other Windows Regional and Language Options the Engineering Workplace and all Engineering Studio tools require a dot (.), as the decimal symbol setting for numbers.

Configurations within Engineering Workplace

The following configurations are only valid for the current project and the user has to perform these configurations for each project. Some of the configurations can be performed automatically during project creation by selecting the **Engineering Base** system extension entry (refer to Using Engineering Base System Extension).

Setting User Identity Properties

With the System Extension *Engineering Base* (refer to Using Engineering Base System Extension) an **User Identity** aspect is provided to each user with a set of properties as shown in Figure 4:

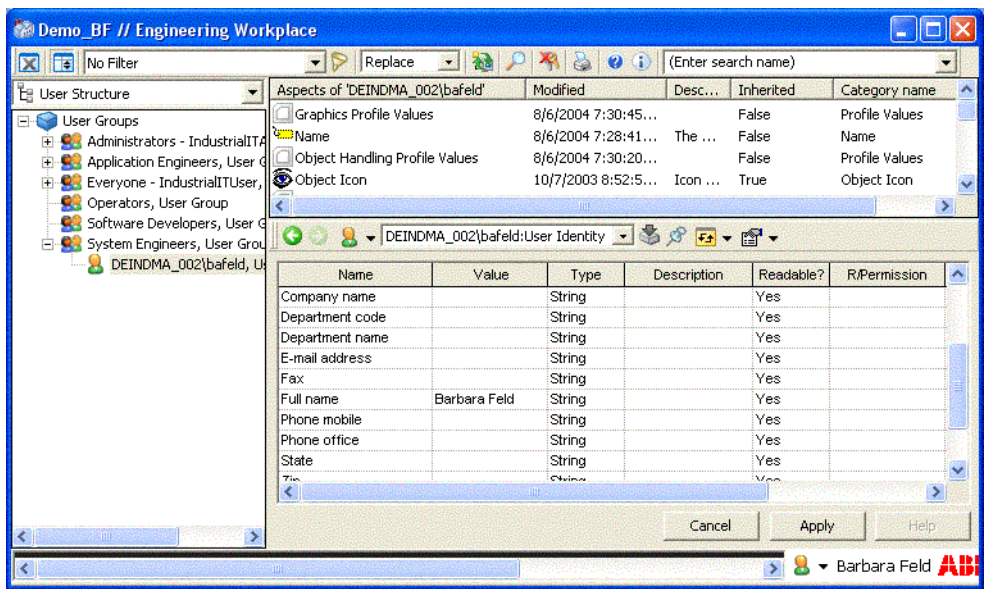


Figure 4. User Identity Properties

These properties can be set and used through the reference mechanisms, which are described in Section 7, Document Manager.

Specifying Rules for Absolute Reference Designations

In Engineering Workplace, navigate to the **Service Structure** and select the **AspectDirectory** service as shown in Figure 5.

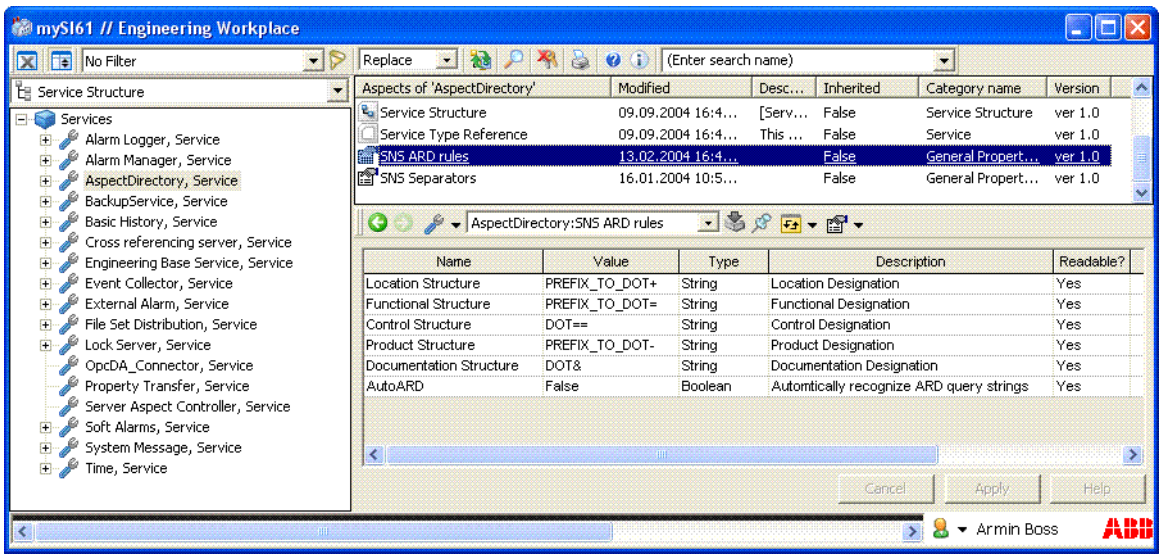


Figure 5. Example for Designation Rule Definition

Select the **SNS ARD rules** aspect and for each structure, specify one of the following rules in combination with the preferred prefix such as, '+', '&', and '==':

- NONE
- DOT
- PREFIX
- PREFIX_TO_DOT

The structures that are not defined in the **SNS ARD rules** aspect cannot display absolute reference designations. Different rules can be selected for different structures.

By default, the following rules are defined for the following structures within Engineering Workplace:

Table 1. Predefined Designation Rules for Structures

Structure Name	Designation Rule and Prefix
Control Structure	DOT==
Documentation Structure	DOT&
Functional Structure	PREFIX_TO_DOT=
Location Structure	PREFIX_TO_DOT+
Product Structure	PREFIX_TO_DOT-

For more information about the working of the designation rules, refer to Section 7, Document Manager.

If the user has changed a designation rule, the system must be restarted for the modifications to become active.

Specifying Separators

In Engineering Workplace, navigate to the **Service Structure** and select the **AspectDirectory** service as shown in Figure 6.

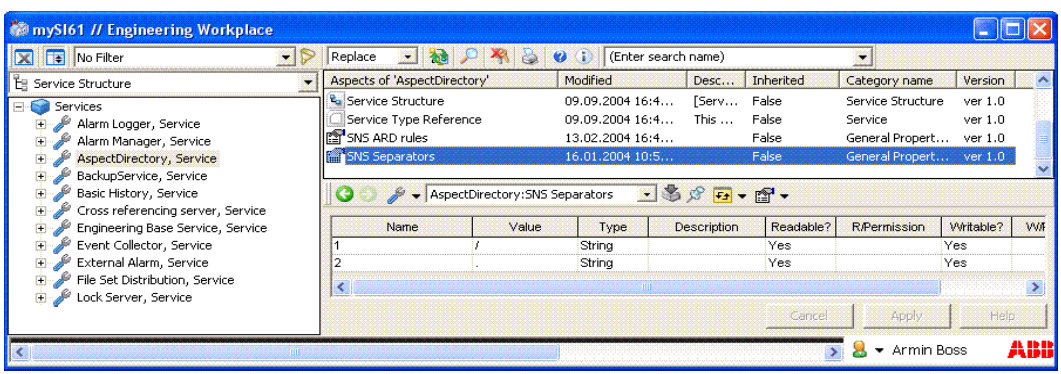


Figure 6. Example for SNS Separators

Select the SNS **Separators** aspect and specify the required separator. Table 2 provides some of the separators used within Engineering Workplace.

Table 2. Separators within Engineering Workplace.

Property Name	Separator
1	/
2	.

Separators are used by the system to identify which characters are part of the names while searching for them. Absolute reference designations also use these separators to separate between the different levels of a relative name.

If the user has changed a separator, the system must be restarted for the modifications to become active.

Displaying Absolute Reference Designations

In the Engineering Workplace, navigate to the **User Structure** and select the required Aspect Object (for example, **Systems Engineers)** in the **User Groups** category as shown in Figure 7.

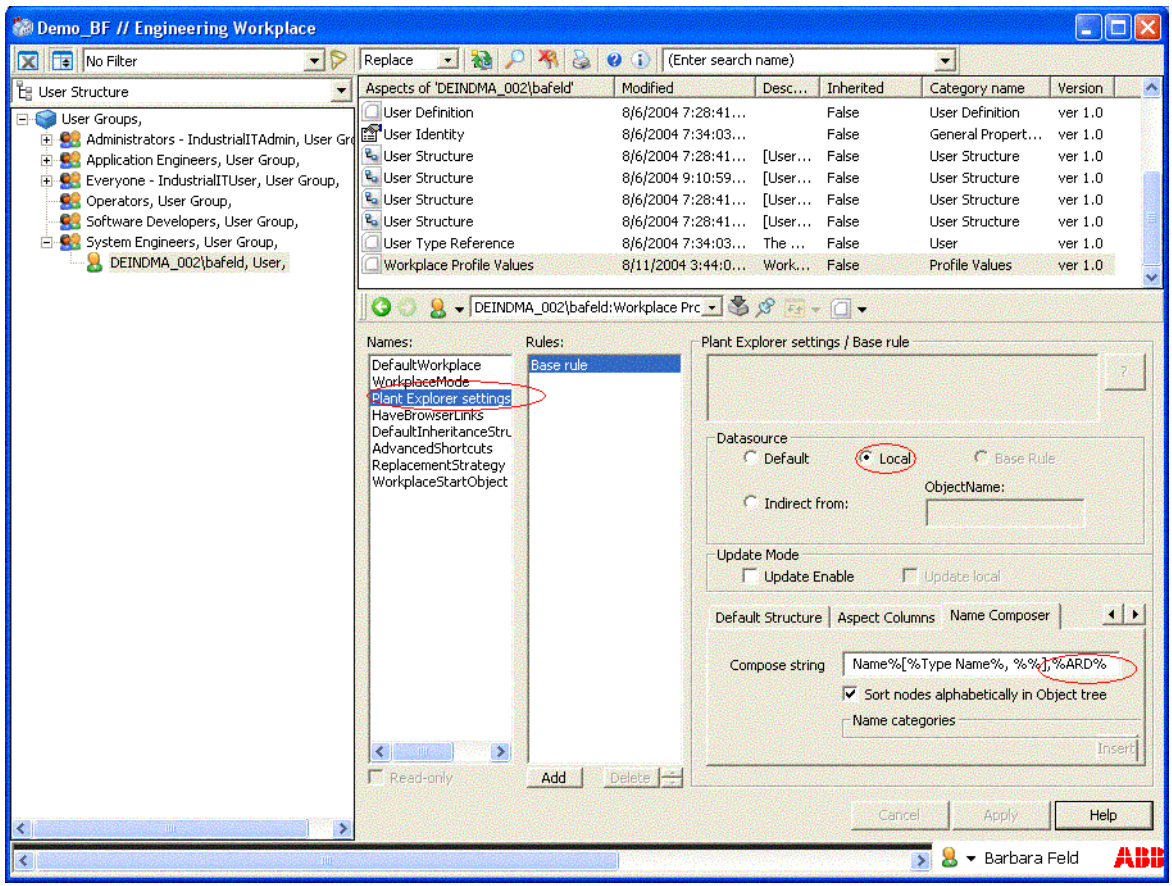


Figure 7. Example of Name Composer Definition

Select the **Workplace Profile Values** aspect, click **Local** in the **Datasource** panel for the **Plant Explorer Settings** property name. In the **Name Composer** tab of the lowest panel, position the cursor at the beginning or the end of the displayed string, as shown in Figure 7. Choose a separator such as a comma with a blank space in the beginning or in the end, or with any string and enter the key word "**%ARD%".**

The definition to display absolute reference designations is valid for all structures available in the current project. If all the definitions are deleted except the key word "**%ARD%**", an empty structure is displayed for those Aspect Objects which has no designation aspects.

Getting Started

Engineering Workplace is a special configuration of the Plant Explorer. With the basic functionality of *Engineering Workplace* and the 800xA system, user can

- handle and administrate engineering projects.
- load system extensions into projects.
- work with relative and absolute reference designations as names for Aspect Objects.
- use an advanced Aspect Object context menu.
- use general project properties.

Those items are describes in the following sections.

User Interface

Application Start-up

The *Engineering Workplace* can only be started on the running projects. A project is started by the *Configuration Wizard* of the 800xA system.

How to start a project (system) see *System 800xA, Administration and Security (3BSE037410*).*

Starting Engineering Workplaces

To start the Engineering Workplace through the *ABB Workplace Login* application select the menu command

Start > All Programs > ABB Industrial IT 800xA > System > Workplace

The following application appears:

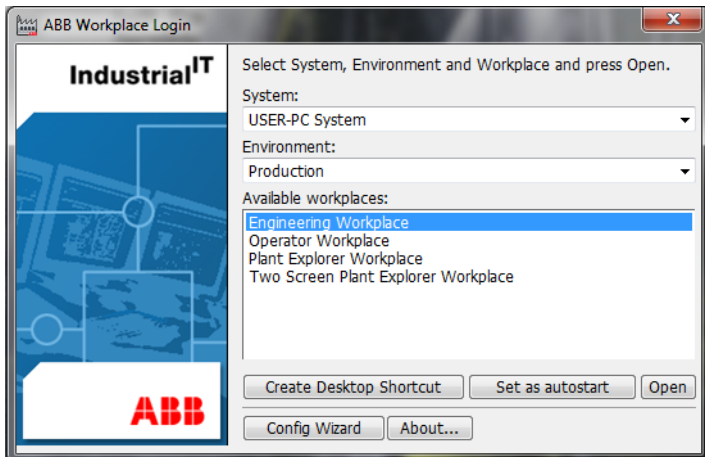


Figure 8. ABB Workplace Login Application.

Select the required System, Environment, and Workplace. Click **Open**.

To create the desktop shortcut icon, click **Create Desktop Shortcut**. Double-click the **Engineering Workplace** icon on the screen to open the application.

The following *Engineering Workplace* application is started:

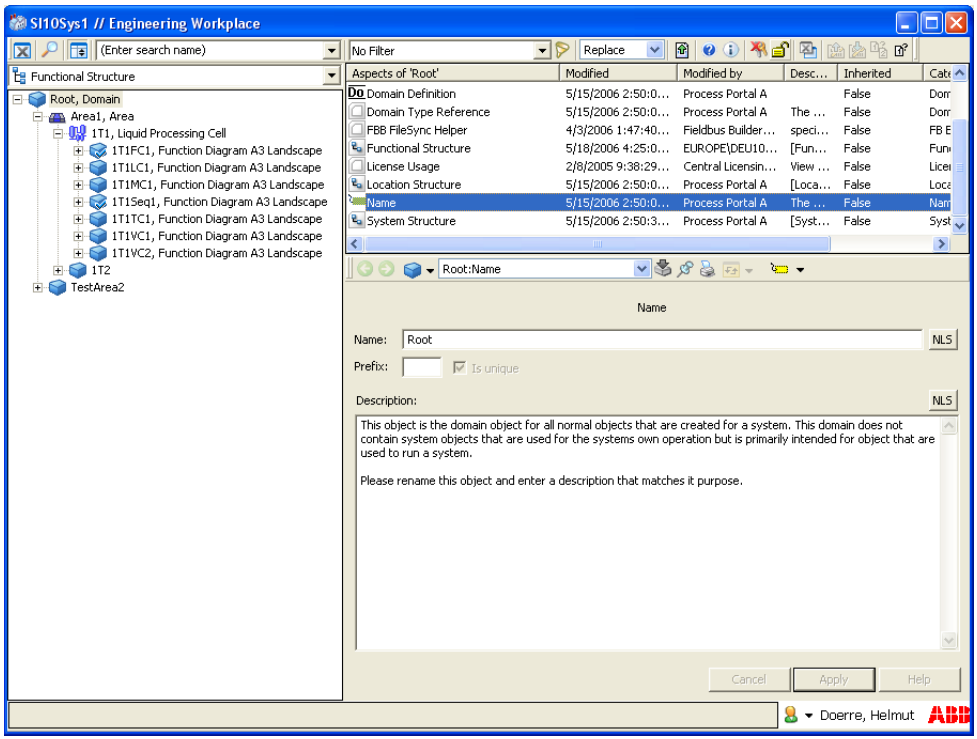


Figure 9. Engineering Workplace

Working with Engineering Workplace

Using Documentation

The Engineering Workplace documents such as user/reference manuals can be viewed using the Windows Start menu items:

All Programs > ABB Industrial IT 800xA > System > User Documentation

All Programs > ABB Industrial IT 800xA > System > Reference Documentation

Using Engineering Base System Extension

Engineering Workplace provides a set of System Extensions like Engineering Base, DM & PM Applications, etc.

The *Engineering Base* System Extension provides the following functionality:

- Engineering Workplace
- Application bar shortcuts for Engineering Workplace
- Aspect Category filter for Engineering Workplace
- Aspect Object context menu extensions
- New aspect category called *Object Category* (additional name for Aspect Objects to group them)
- User Identity properties for each user

To get this functionality available in each project user has to add the *Engineering Base* System Extension by checking the check button **Engineering Base.**

Using General Project Properties

A set of defined general properties exist for engineering project information. These properties are split into 3 aspects with its predefined properties shown in the following table:

Table 3. General Project Properties.

Aspect Name	Aspect Properties
Basic Project Properties	ProjectDataDir
Customer Settings	Name, Project Name 1, Project Name 2, Project Name 3, Factory, District
Responsible Company Settings	Name, Department, Project Leader, Revision, Start Date

User can find these definitions in Aspect Object *Project* located in structure *System Structure*.

Creating Reference Designations

There is a possibility to create relative reference designations automatically by using Aspect Object types and the “Name Hook” mechanism of the 800xA system.

Define your own Aspect Object type or use an existing one. Select this Aspect Object in the *Object Type Structure* and add a new aspect of category *Basic Object Name Hook*. Open the pre-view area of that aspect and enter the requested information.

The following figure shows an example of the requested information:

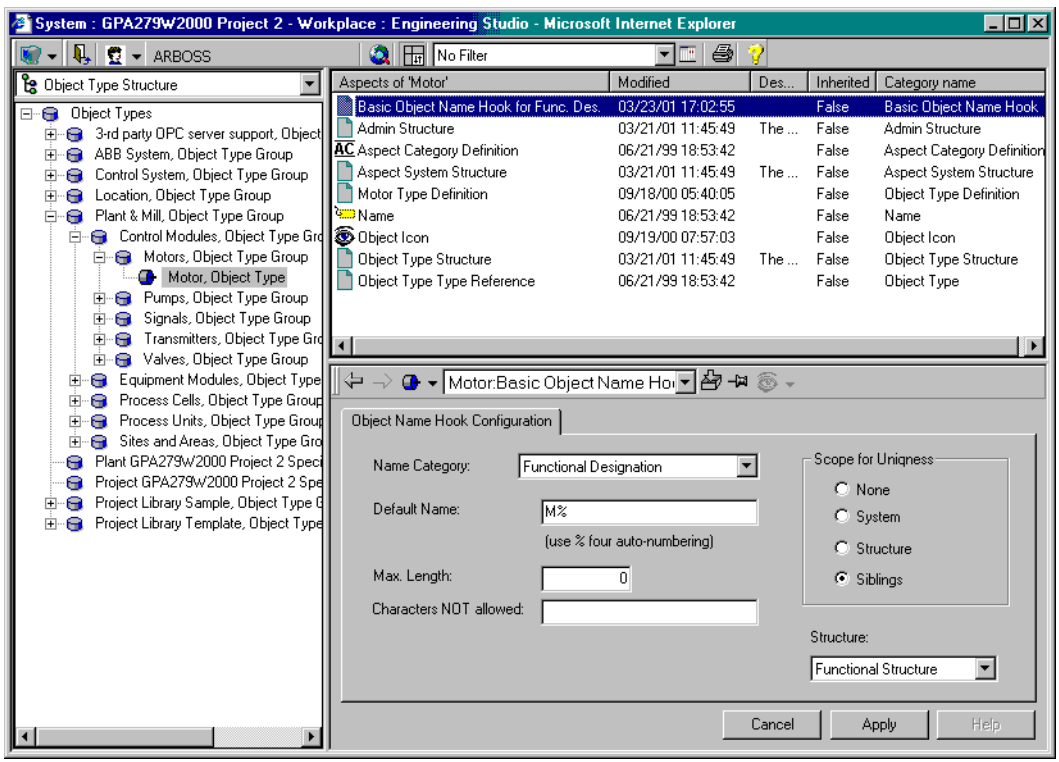


Figure 10. Basic Object Name Hook Example for Designations.

Click **Apply** and create a new aspect. This new aspect have to be the same aspect category user specified in the *Name Category* of the *Basic Object Name Hook* aspect. If user specified e.g. *Functional Designation* then user has to create an aspect of that category. Then open the preview area of the Aspect Object type definition aspect e.g. *Motor Type Definition* and select the **Aspect Control** tab. Select the related aspect e.g. *Functional Designation*, do not check the button

Inherit when object is created and check the button **Copy when object is created** as shown in the following figure:

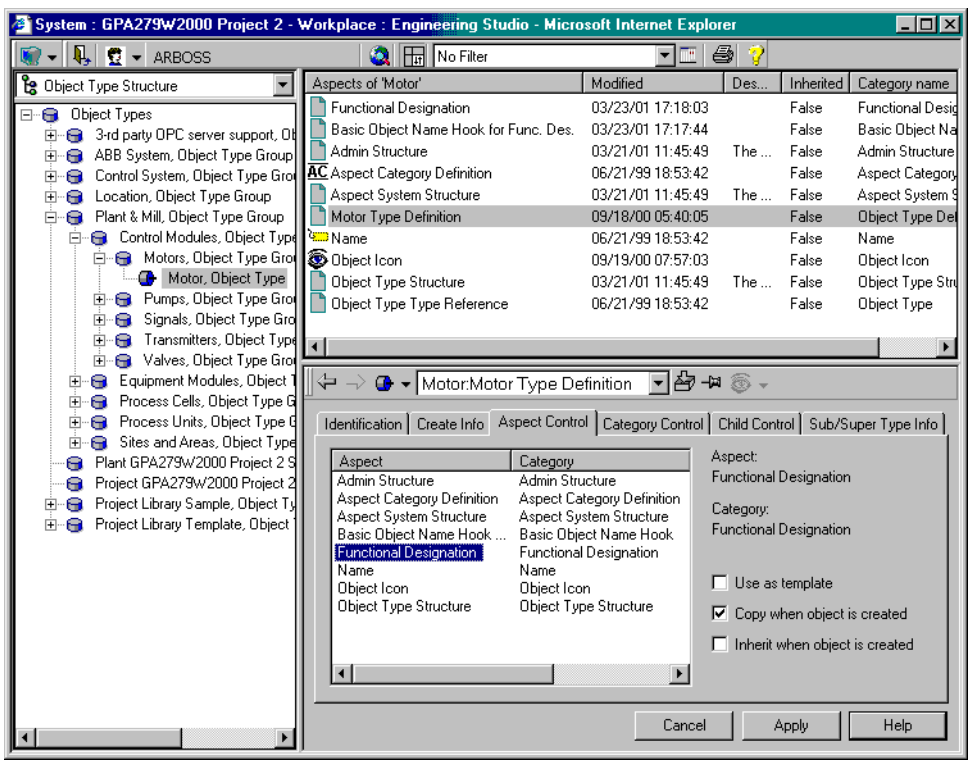


Figure 11. Functional Designation Aspect in Object Type Definition.

Click **Apply.** This Aspect Object type can be used to make new Aspect Objects. The result of this example is

- an aspect of category *Functional Designation* is automatically created if the Aspect Object is created in the *Functional Structure*.
- The relative reference designation is created automatically - here in this example the value *M1* will be generated if the children Aspect Objects of the selected parent are having no other *Functional Designation* aspects.

Advanced Operations for an Object

Right-click the required object and select **Advanced** from the context menu to access functionalities such as:

Renaming Aspect Objects of Whole Substructures

Select **Rename Substructure** and the following dialog appears:

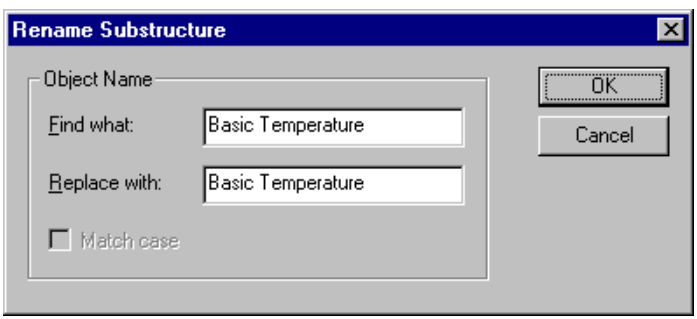


Figure 12. Rename Substructure Dialog

Enter your characters or search string (case sensitive) into the **Find what:** edit field which is used to match Aspect Objects in the whole substructure (if exists). Enter the new characters or the new string into the **Replace with:** field which will be used to substitute the matched Aspect Objects. Click **OK** to rename all matched Aspect Objects.

This feature is a generic mechanism to rename Aspect Objects. It cannot check the automatically renamed items in other aspect systems or tools such as AC 800M Connect / Control Builder M Professional or Function Designer. Refer to their naming rules and functions in the corresponding manuals.

Deleting Aspect Objects from all Structures

Select **Delete from all Structures** and the following dialog appears:

Figure 13. Confirm Delete from all Structures Dialog.

Click **Yes** to delete the Aspect Object from all structures. Aspect Objects as children from the selected parent which are placed **below different parents** in other structures or in the same structure **will not be deleted**.

Adding Document Aspects Faster

Select **New Document** to create a new document aspect. This is a faster method for adding document aspects without navigation to the document aspect category.

Starting Bulk Data Manager

Select **Bulk Data Manager** and the following application appears:

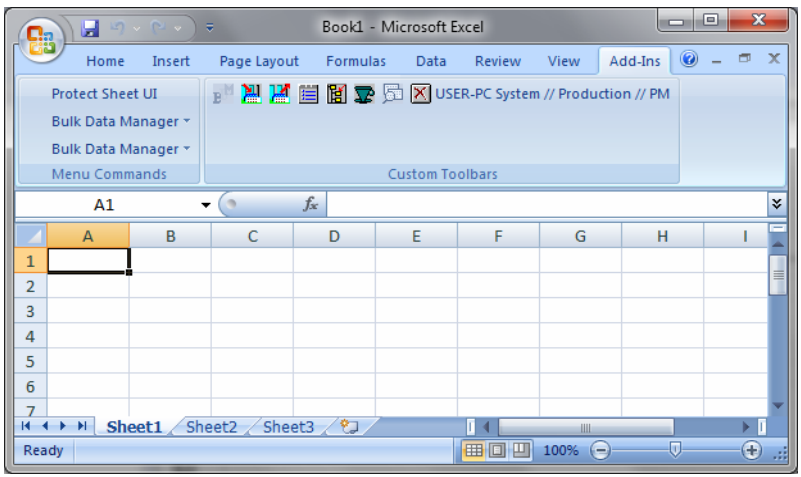


Figure 14. Bulk Data Manager Excel Add-in Application.

After the **Bulk Data Manager** workbook appeared the sheet is already activated and the related Aspect Object is already attached. For further information refer to Section 3, Bulk Data Manager.

Using Engineering Templates

Select **Engineering Templates** and the Engineering Templates folder is opened:

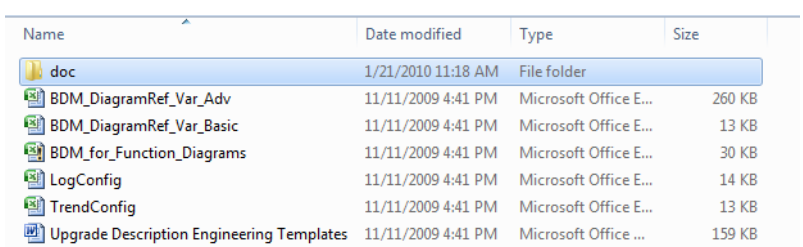


Figure 15. Engineering Templates Folder

All Engineering Templates are available in this folder. Double-click the required template to access it. Double-click the doc folder where the descriptions are provided for the available Engineering Templates.

Section 3 Bulk Data Manager

This section describes Bulk Data Manager and its functions, commands, and utilities. To get the most recent hints, recommendations, and settings read the Release Notes which are delivered along with the product.

Bulk Data Manager is a component of Engineering Workplace. It supports list oriented bulk data operations like handling of signal lists, tag lists, etc. It also integrates Excel sheets into Aspect Objects and enables the user to create formatted templates.

Bulk Data Manager is composed of:

- Microsoft® Excel (not included with Bulk Data Manager). See the *System 800xA, Installation (3BSE034678*)* manual regarding the supported product versions.
- A set of Excel Add-Ins that integrate Microsoft Excel into Engineering Workplace.
- A set of predefined templates.

When Environment support is enabled Bulk Data Manager can be used in Engineering Environment and in Production environment. It relies on default reservation strategies of 800xA platform.

Bulk Data Manager displays the environment besides the attached system in the **Attach /Detach System** button in its toolbar.

800xA 5.1 onwards, Engineering Studio no longer supports Microsoft Office 2003 / XP / 2000.

800xA 5.1 Rev A onwards, Engineering Studio supports Microsoft Office 2007 / 2010.

Bulk Data Manager Functions

Bulk Data Manager offers the following main functions:

- Bulk data management such as:
  - fast and easy list oriented processing of engineering data in Microsoft Excel.
  - retrieving, filtering and saving data from/to other System 800xA applications.
  - automatic creation of objects, aspects, and structures based on reusable solutions.
  - import and export of customer data or data from other applications.
  - Macro environment
    VBA macros can be created in Microsoft Excel handling objects and aspects.
  - work off-line, i.e. without a connection to the Aspect Server, with application data once they are loaded from the platform.
- Formatted templates have the following features:
  - enter and retrieve (reporting) engineering data using pre-defined or user-defined formatted templates based on Microsoft Excel. Data can be retrieved from other System 800xA applications, manipulated and written back to these applications.
  - link data from other System 800xA applications into a template. Linked data can be updated automatically or on the user's demand. Data linked into the template can also be changed and is automatically written back to the "owning" application.
  - build user defined templates using the rich set of Microsoft Excel formatting and graphic features. This allows the user to create spread sheet applications like budget planning, resource calculations, price calculations, or even advanced applications like TagSheets, SignalLists, LoopDiagrams, dimensioning of motors or valves. Calculations can be enriched with texts and graphics. Hyperlinks allow to embed external text or graphic files into the spread sheet.

A set of predefined templates are provided as a starting base (see Appendix A, Engineering Templates).

- add templates as document aspects to objects using the Document Manager. In this case administrative attributes like author, creation date, status, etc. can be maintained for the aspect.

- Monitoring of live (process) data enables the user to:
  - monitor e.g. the process value or status information of a set of selected objects like process signals or field devices (provided the properties of interest are accessible for other applications)
  - present important process information in a dynamic updating charts like radar diagrams or pie charts
- Performing calculations using a variety of strong Microsoft Excel functions including the possibility to:
  - retrieve and include data from other System 800xA applications into calculations
  - include live (process) data into calculations
  - write results of calculations back to other System 800xA applications

If user uses Bulk Data Manager to save data to System 800xA applications in an operating environment ensure that this does not conflict with secure plant operation.

Do not embed a Bulk Data Manager worksheet into another document (for example a Word document). If user embeds the Bulk Data Manager worksheet, the Bulk Data Manager menus will not be available and will not work properly.

Store user-defined Bulk Data Manager sheets in protected folders if they are not managed through Document Manager aspects.

Considering Project and Directory Structures

Bulk Data Manager stores files in the following directories:

- Software:
  <drive>:\Program Files\ABB Industrial IT\Engineer IT\Engineering Studio\Engineering Platform\Bulk Data Manager\bin
- Predefined templates can be found in two locations:

1. In the **Engineering Templates** folder, on the Desktop
2. <drive>:\Program Files\ABB Industrial IT\Engineer IT\Engineering Studio\Engineering Templates

Capacity & Performance

The capacity of the system is defined by the limitations of Microsoft Excel, the 800xA system, and the physical limitations of the computer.

The Bulk Data Manager can handle in one Excel worksheet

- up to 250 properties (columns) in a horizontal view
- and 65536 (rows) not filtered
- and 60000 (rows) filtered (the total amount of dropped Aspect Objects and the expected filter result).

When using Bulk Data Manager to read/write **sizable** amounts of system data the response time is also depended on the involved aspect systems. Be aware that reading or writing 10 thousands of Excel rows (one row equals an Aspect Object in horizontal configuration) can take a considerable period of time.

Bulk Data Manager must not be run:

- in parallel on several engineering clients as this will impact performance.
- in a system under operation as this will impact Operating performance.

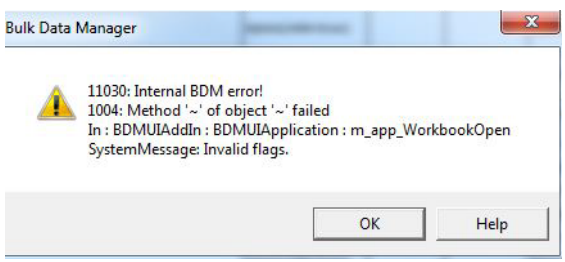
Settings

Keep the macro security setting as **Enable all Macros for Bulk Data Manager** (Microsoft Excel 2007 / 2010).

1. Launch Microsoft Excel 2007 / 2010.
2. Click **Office** button placed at the top left corner of the Excel application.
3. Click **Excel Options**.
4. Click **Trust Center** in the opened **Excel Options** dialog window.
5. Click **Trust Center Settings**.
6. Click **Macro Settings**.
7. Select **Enable all macros**.
8. Click **OK**.

In Microsoft Word, user can work with the default macro security settings.

If an Excel sheet is opened from the network drive, following error messages are displayed due to protected view of Microsoft Excel:



This message can be ignored, click **OK** to acknowledge this message.

Getting Started

User Interface

Bulk Data Manager is a Microsoft Excel based application. This chapter describes the additional menu entries and toolbars.

Menus

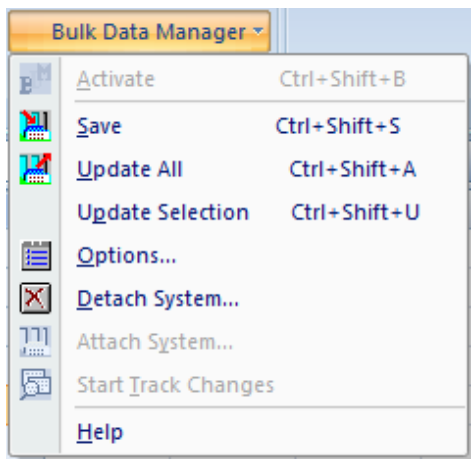


Figure 16. Menu Bulk Data Manager

- **Activate**
  Activate the Bulk Data Manager for the active Workbook.
- **Save**
  Save data from Microsoft Excel to System 800xA applications.
- **Update Selection**
  Retrieve data from System 800xA applications and update the selected dynamic data areas in Microsoft Excel. This command applies to Auto-update Data Areas (see Configuring an Auto-update Data Area) and to Property References (see Inserting Property References) but not to the Default Data Area (see Configuring the Default Data Area).

- **Update All**
  Same as Update Selection, however all dynamic data areas are updated.
- **Options**
  Diverse options can be set to control the behavior of the Bulk Data Manager (see Loading Data into the Default Data Area and Saving Data in an Auto-update Data Area).
- **Detach System**
  The Bulk Data Manager will be detached from the System used for saving data areas and Start Object used for updating Auto-update Data Areas and Property References.
- **Attach System**
  A dialog allows to attach the Bulk Data Manager to a System used for saving data areas and a Start Object used for updating Auto-update Data Areas and Property References. When Environment Support is selected in Configuration Wizard, this dialog also allows to select between Engineering and Production environment.
- **Help**
  Opens help contents. Same as submenu Contents in Help > Bulk Data Manager.

Help > Bulk Data Manager

- **Contents**
  Opens help contents.
- **About Bulk Data Manager**
  Displays the actual version of the Bulk Data Manager.

Toolbar

A subset of menu entries as provided under menu **Bulk Data Manager** and the context menu are also available in the toolbar **Bulk Data Manager**. Those entries showing the same images have the same function.

Figure 17. Bulk Data Manager Toolbar

Context Menu

To open the context menu right-click the required cell.

- **Property Reference > Insert**
  Opens Create Property Reference dialog, which allows to create a Property Reference with or without subscription for live data for all selected cells.
- **Property Reference > AutoInsert**
  See Create a Subscription for Live Data.
- **Property Reference > Delete**
  Property Reference will be deleted from the selected cells.
- **Property Reference > Update All**
  Update values in all cells, which have Property References.
- **Property Reference > Update Selection**
  Update values in selected cells, which have Property References.
- **Property Reference > Hide Reference**
  This toggle command is used to Show / Hide all cell comments.
- **Property Reference > Hide Indicator**
  This toggle command is used to Show / Hide all cell comments and cell comment indicators.
- **Make Auto-update Data Area**
  Changes a Default Data Area to an Auto-update Data Area provided that a Default Data Area exists. The menu entry is only shown if a Default Data Area exists on the sheet.
- **Insert Object Path**
  Opens the Insert Object Path dialog, which allows to navigate to various

objects in different structures and insert the object path in the Bulk Data Manager Data Sheet.

- BDM Filter
  See Bulk Data Manager Filter.
- **Aspect Commands**
  How to perform commands upon aspect see Aspect Commands).
- **Aspects**
  This command pops-up the list of aspects of the object whose name is specified by the selected cell (see Cross-Navigation to System 800xA Applications).

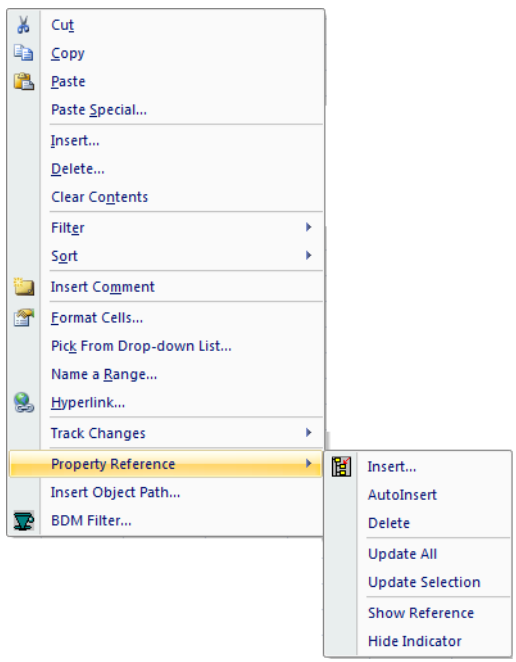


Figure 18. Bulk Data Manager Context Menu

Application Start-up

The Bulk Data Manager distinguishes between Workbooks that are managed by the Bulk Data Manager and Workbooks that are not managed (i.e. standard Excel workbooks). By default, Excel workbooks are not managed until they are activated by **Bulk Data Manager > Activate**. Once the workbook is activated, the menu items **Save**, **Update Selection**, **Update All**, **Options,** etc. become available in the **Bulk Data Manager** menu and user can start using the Bulk Data Manager as described in the following sections.

The fact, that a workbook is managed by the Bulk Data Manager is saved together with the workbook.

There are different ways to start-up the Bulk Data Manager:

- Start Microsoft Excel with a new Workbook or open a Workbook that has not yet been activated, then activate the Bulk Data Manager by **Bulk Data Manager > Activate**.
- Open an existing activated Excel file.
- Open a Document Aspect which refers to an Excel file in the Plant Explorer. From the Aspect list window of Plant Explorer, right-click and open the Document aspect. The Bulk Data Manager is activated automatically.
- Open the Bulk Data Manager through the context menu **Advanced > Bulk Data Manager** on an object in the Object Browser of the Plant Explorer. The Bulk Data Manager is activated automatically.
- Use the Start Menu entry **Start > All Programs > ABB Industrial IT 800xA > Engineering > Bulk Data Manager**. The Bulk Data Manager is activated automatically.

Use **CTRL+SHIFT+B** to activate the Bulk Data Manager by keyboard.

For a new user (a user that was not the installer of Bulk Data Manager) the Excel Add-In “LBEMacrcos.xla” is not registered.

If this user starts Excel or Bulk Data Manager for the very first time, an internal BDM error message appears. Click the Bulk Data Manager toolbar button Activate or menu item Bulk Data Manager > Activate. Then the missing Excel Add-in “LBEMacros.xla” will be registered automatically. This step needs to be done once a time for each new user account on the machine.

Excel application with BDM sheet may not terminate itself automatically when opened through Fileviewer in Process Portal or opened through Internet Explorer. Follow the workaround explained below to resolve this issue by setting some folder options in Windows Explorer.

1. Open Windows Explorer.
2. Click Tools > Folder Options form the menu bar.
3. Click File Types tab in the opened Folder Options dialog window.
4. Select XLS from the registered file types and click Advanced.
5. In the opened Edit File Type dialog box, select Open from the Actions and check Browse in same window.
6. Click OK.

Working with Bulk Data Manager

The following sections describe:

- how to work with a Default Data Area (Working with the Default Data Area)
- how to work with an Auto-update Data Area (Configuring an Auto-update Data Area to Saving Data in an Auto-update Data Area)
- how to work with different systems/projects (Setting System and Start Object)
- how to exchange data with other applications (Data Exchange with Other Applications)

- how to build formatted templates (Creating Formatted Templates)
- how to monitor live (process) data (Monitoring of Live (Process) Data)
- advanced techniques (Using Functions to Read/Write Data to Creating Macros)

Working with the Default Data Area

The Default Data Area allows the user to load and save data from/to other System 800xA applications. The main steps to perform are:

- configure the Default Data Area to specify where to read data from or where to save data to (Configuring the Default Data Area)
- load data from other System 800xA applications (Loading Data into the Default Data Area)
- save data to other System 800xA applications (Saving Data to System 800xA Applications)

Configuring the Default Data Area

A data area is a contiguous range of cells in a Worksheet with a certain number of columns and a limited or unlimited (max 65536 rows) number of rows. Data can be loaded automatically in a data area (**Auto-update Data Area**) or by drag and drop of objects (**Default Data Area**). There can be one Default Data Area and many Auto-update Data Areas per Worksheet. Each data area has a headline that defines which properties of which aspect category are to be handled in the data area.
The following **steps** describe how to configure the Default Data Area.

1. Drag an aspect from the Plant Explorer and drop it into a cell of the Worksheet. During the drop action the dialog Configure Properties appears, see Figure 19. showing all accessible properties of the selected aspect category. Mark all properties in the check box which has to be written into the headline of the Worksheet and click **OK**. By this all selected properties of the selected aspect category are filled into the headline. The first drop operation automatically includes three required properties (“Command”, “Object Identification”, and “Source Object”), see Figure 20.

2. If certain selected properties are not needed, they can be deleted (e.g. delete a complete column or just the property from the headline).

Do not delete the properties “Command”, “Object Identification”, and “Source Object”.
Do not delete the comments of the property “Command” and the last property in the headline.

3. Repeat the steps above for all aspect categories for which properties should be included in the data area.

See also Understanding the Configuration Headline.

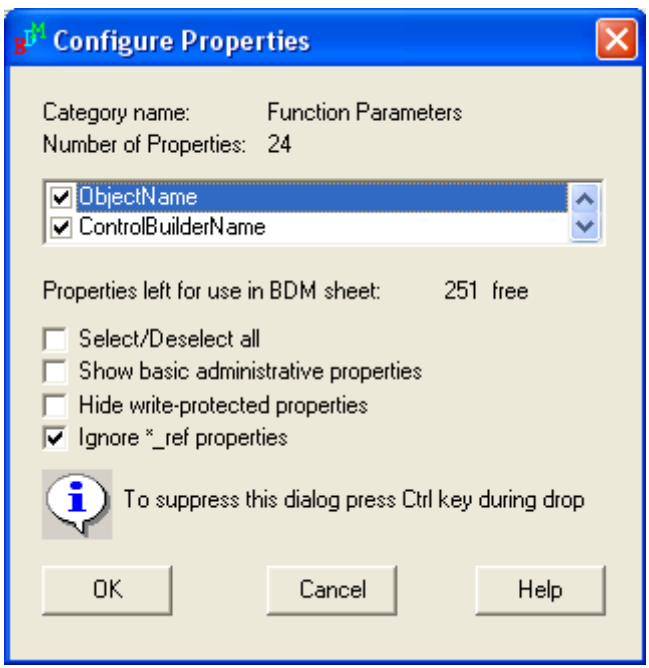


Figure 19. Configuring the Default Data Area, Configure Properties Dialog

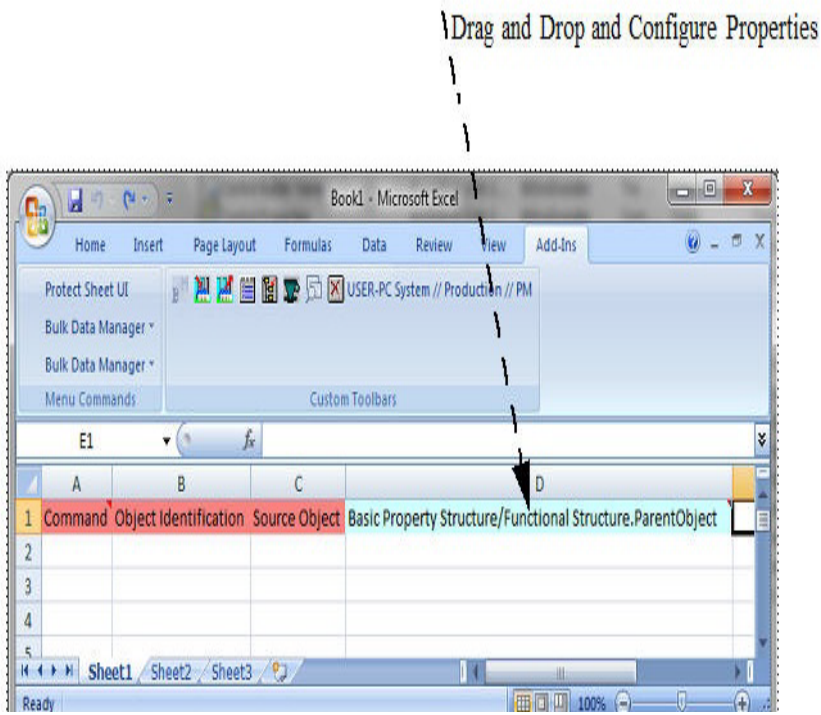


Figure 20. Configuring the Default Data Area

Do not click Cancel Button of Excel Save Changes Dialog.

If user quits Microsoft Excel after activating a Bulk Data Manager workbook, a query dialog appears with the message: Do you want to save the changes you made to <name of the workbook>.xls? Bulk Data Manager workbook will not work properly, if user clicks **Cancel**.

Loading Data into the Default Data Area

Data from different System 800xA applications can be loaded into the Default Data Area. Which properties from which aspects are to be loaded is determined by the headline of the data area. Please follow the **steps** described below:

1. Drag and drop an object from the Plant Explorer into the Worksheet in a row below the headline. By doing so, the data of all configured properties of all aspects of the selected object and its children in the current structure (optional) are loaded into the Worksheet.
2. Repeat this step for all other objects to be loaded.

Remarks:

- In order to include structure information, for example the location of an object in the Functional Structure, drag and drop the Functional Structure aspect into the Worksheet.
- To limit the number of rows to be filled into the data area perform the following steps:
    a. Edit the comment in the cell “Command” (context menu **Edit Comment**).
    b. Replace the keyword “$$ROW” by “$$ROWn” (n is the number of rows to be filled with data). Close the comment by clicking outside the comment.
    c. If more data is retrieved during drag and drop as defined by the keyword “$$ROW”, the remaining data is cut.

Automatic Excel cell format creates sometimes wrong cell format while reading aspect property values. Excel cells do not have a specific cell format, the default cell format “General” is used for a worksheet. Aspect property values are containing commas, points, etc. between numbers or True or False.

After reading this kind of property values Excel does an auto format on the current cell. For example the value “1031,1033” is converted to number 1031033. During the next save the wrong value is written or if the property is not writable (by its definition) an error message appears because of an old/new value comparison in Bulk Data Manager.

Format these kind of cells to the cell format “Text”. This can also be done for entire columns or rows.

Read and write boolean values of an “AC 800M Connect” aspect system:

The Aspect Object has an aspect from the AC 800M Connect aspect system and this aspect has properties of data type Boolean. The cells in Excel are formatted as category “General”.

The AC 800M Connect aspect system handles aspect properties of data type Boolean as character strings. This means reading aspect property values of this data type into the Bulk Data Manager converts the value into the numbers 0 and -1 where writing the properties requires the value to be True or False. Reading and writing would write the values to illegal values for the AC 800M Connect aspect system.

Change the category/format of the Excel cells/columns before reading or writing data to “Text”.

After reading data into Excel cells user has to perform a replace on cells with Boolean aspect property values and replace the values True by ‘True’ and False by ‘False’.

Options

For a detailed description of all Bulk Data Manager Options see Bulk Data Manager Options.

Filter Data

Data retrieved from the Plant Explorer and included in the Worksheet can be filtered according to user defined criteria. To enter a filter, follow the steps below. When using filters, the option “**Subtree enabled”** should be selected, otherwise, only a single object will be read.

a. Enter the keyword “Filter:” in the column “Command” immediately below the headline of the data area.

b. Add additional filter criteria related to the properties of the headline in the same row (see example below (refer to Figure 21)).

It is possible to use wildcards and other operations (even user defined functions) according to Microsoft Excel “Advanced Filter” possibilities (see Microsoft Excel documentation or on-line help). However only one row with filter criteria is allowed.

c. To read child objects from a different structure than the structure that was used for drag and drop in the Plant Explorer, append the full structure name to the “Filter:” keyword. E.g.:
“Filter:Control Structure”

d. To read all root objects of a structure use the **“#ROOT:”** filter command. When the option “**Subtree enabled”** is selected all children of the root objects will be read. This means, that the whole hierarchy of objects is read.
For example:
“Filter:#ROOT:Functional Structure” configures Bulk Data Manager to read all root objects of the “Functional Structure”.

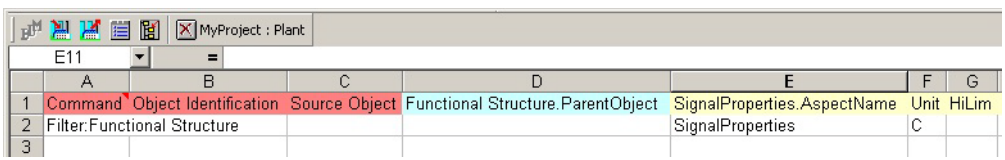
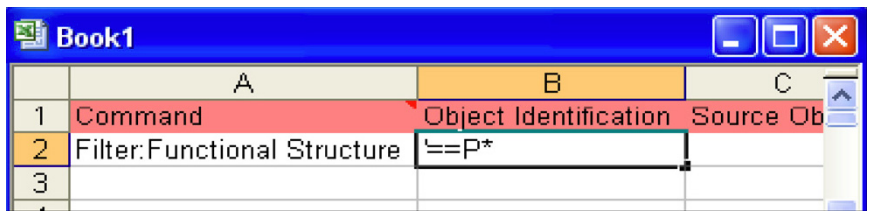


Figure 21. Filter Data

The filter depicted in Figure 21 above retrieves only the data of objects located in the Functional Structure having an aspect called “SignalProperties” with the property Unit set to “C”.

Additionally to the filter function described here, the Autofilter function of Microsoft Excel can be applied. The difference is that the “Filter:” command limits the objects loaded into Excel while the Autofilter function only filters the display of the objects (user can switch between filter criteria easily).

Usage of Filter and Absolute Reference Designations starting with Prefix “=”: The prefix character called equal (=) has a special meaning in Excel. It must be duplicated to match the specified filter conditions. Also it has to be started with a single quote.The filter cell look like in the following example:



Saving Data to System 800xA Applications

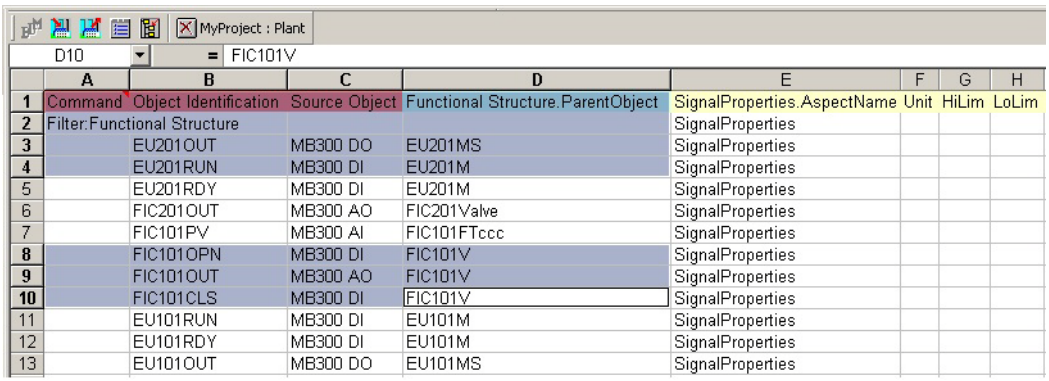
If user uses this function to save data to System 800xA applications in an operating environment ensure that this does not conflict with secure plant operation.

Once data has been loaded into a data area and has been manipulated, it can be saved back to System 800xA applications. To which application the data is saved, is determined by the headline defining which aspect category the different properties belong to. In order to save the data, follow the steps described below.

- To save all data, perform menu **Bulk Data Manager > Save** (or keys **CTRL** + **Shift** + **S**). This saves all data in all data areas.
  A Data Area is by default terminated by the first empty row. For user defined or pre-defined templates (see Creating Formatted Templates) there are however cases where the Data Area have to be terminated when the first empty object identification (column “Object Identification”) is encountered. To specify this behavior set the keyword **$$COI** (i.e. **Check O**bject **I**dentification) as comment to cell “Command” of the headline.
- To save only a selection of cells:
  - Select one or several areas of cells including at a minimum the columns “Command”; “Object Identification” and “Source Object”. All areas of the selection must belong to one data area and the configuration headline

must be included.
It is possible to select different areas using the **CTRL** key (see example below).

Perform menu **Bulk Data Manager > Save** (or keys **CTRL + Shift + S**).



	A	B	C	D	E	F	G	H
1	Command	Object Identification	Source Object	Functional Structure.ParentObject	SignalProperties.AspectName	Unit	HiLim	LoLim
2	Filter:Functional Structure				SignalProperties			
3		EU201OUT	MB300 DO	EU201MS	SignalProperties			
4		EU201RUN	MB300 DI	EU201M	SignalProperties			
5		EU201RDY	MB300 DI	EU201M	SignalProperties			
6		FIC201OUT	MB300 AO	FIC201Valve	SignalProperties			
7		FIC101PV	MB300 AI	FIC101FTccc	SignalProperties			
8		FIC101OPN	MB300 DI	FIC101V	SignalProperties			
9		FIC101OUT	MB300 AO	FIC101V	SignalProperties			
10		FIC101CLS	MB300 DI	FIC101V	SignalProperties			
11		EU101RUN	MB300 DI	EU101M	SignalProperties			
12		EU101RDY	MB300 DI	EU101M	SignalProperties			
13		EU101OUT	MB300 DO	EU101MS	SignalProperties			

Figure 22. Save Selected Data Ranges

The following rules apply to the save operation:

- The data is saved into the attached system. The Start Object is not used for a Save operation. See also Setting System and Start Object.
- If an object already exists, it will not be created again (except the command “New” is used - see below). Objects are identified by the column Object Identification using their name, path, reference designation, or object id.
- If an aspect exists, it will not be recreated, but updated regarding its properties. Aspects are identified by the Aspect Category and the aspect name. If no aspect name is given, the category name is used as default name (e.g. in Figure 22 it is checked if a Functional Structure aspect with name “Functional Structure” exists).
- If a source object is entered, the object to be created is either:
  - instantiated if the Source Object is an object type
  - duplicated if the Source Object is an object instance

- The object created will be named according to the column “Object Identification”. If a complete substructure is duplicated or a composite object type (object type with a substructure) is instantiated, the objects are renamed based on the following naming convention:

Figure 23. Naming Convention

  - The source object is called “object”, for example. This string is also part of the object name of the child objects.
  - The destination object is called “FIC”. The name of the destination object will replace the string “object” in all child objects of the source object.
  - If the string “object” is not part of the name of a child object, no renaming takes place.
  - The string “object” can be inserted anywhere in the name of the child objects (it doesn’t need not to be at the beginning).

If errors occur in the save operation, they are listed in a separate Error-Worksheet. Typical problems are:

Error writing aspect property.
The aspect system returned an error on writing this property. Depending on the aspect system, there is additional information available in the Details and Error Code column.

Ambiguous object identification.
Use path names or ARDs in column “Object Identification” that identify the object unambiguously, e.g. Parent/Obj. Objects matching the ambiguous object identification are listed in the Details column.

Ambiguous aspect identification.
Usually, this error occurs if there are multiple aspects with the same name (e.g. two Functional Structure aspects). Either rename the aspects to make them unique or use aspect ids as identification.

The object type definition option is activated (see below):
In this case the save operation might have manipulated object types. This option is intended only for manipulating object types.

Do not use pre-defined prefix characters like “+”, “- ”, “=”, “&” as part of a designation name. This can result in creation of two Aspect Objects with the same designation.

Do not create Aspect Objects without any Structure aspect. If a Bulk Data Manager workbook is configured without any structure aspects or empty values for aspect property “ParentObject” then Aspect Objects are created – but without any structure aspect. These Aspect Objects do not appear in Plant Explorer.

Use the workplace’s Find tool to delete these Aspect Objects. Or configure a structure aspect to the headline of the existing Excel workbook and fill these cells with a valid parent Aspect Object name and press save. Then these Aspect Objects will appear in the related structure.

If Bulk Data Manager workbook contains aspects of aspect system “AC800 M Connect” data are written to the currently opened Control Builder M project independent of the attached 800xA system.

To write to a different Control Builder M project, open the destination project with aspect context menu item “Open Project”.
To write to a different 800xA system (in an environment with multiple Aspect Servers) and a different Control Builder M project, detach the current system and attach the destination system in Bulk Data Manager and open the destination project of Control Builder M.

If the user creates a new object of an object type with a name hook where the name is not unique, the object is renamed by the system. This causes unexpected modifications such as placing child objects on root level if the created object is referenced later in the Excel worksheet (for example as ParentObject). Always use object names that do not violate the name hook rules.

Options

For a detailed description of all Bulk Data Manager Options see Bulk Data Manager Options.

Do not activate the option “**Object Types / Allow Modification**”.
Use this option only when creating or changing object types (see Appendix A, Engineering Templates)

Error Logging

Errors occurring during a save operation will be logged in a separate Worksheet called “Errors”. Just click on the “Errors” Worksheet to see what error occurred. Click on the Error to open the data cell that caused the error. The error can be corrected and the data can be saved again.

Using Commands

Commands can be performed upon objects, aspects or properties

- see Object Commandssee Aspect Commandssee Property Commands

Object Commands

Entering commands into the column “Command” offers the user the following possibilities when data is saved:

- By default **no command** is entered.
  This will create or update objects/aspects.
  - objects are either created from scratch (if no Source Object is entered) or
  - objects are copied based on an existing object instance with an optional subtree (Source Object identifies an existing object instance) or
  - objects are instantiated based on an existing object type with an optional subtree (Source Object identifies an existing object type)

  Remark: If the object to be created does already exist it will not be created again but only updated (see also Saving Data to System 800xA Applications).
- **Delete**
  This command will delete the object identified (property “Object Identification”) and its optional substructure.
- **Remove**
  This command removes the object from the structure used to identify it (property “Object Identification”) if the object is placed in more than one structure. If an object exists in one structure only the command deletes the object.

If the object with its child objects is placed in one structure only: Child objects will be re-created on root level.

If the object with its child objects is placed in the structure used to identify the object and in an additional structure: Child objects will not be re-created.

If the child objects have been dropped into the worksheet too (BDM option Subtree enabled): Use the ignore command on the child objects in the work sheet. Otherwise error messages will show in the Errors work sheet (Parent object not found, Create object failed).

- **Place**
  This command places an object that is already located in a structure again into the same structure (without the place command the object would be moved). Use this command together with the “Parent” property of structure aspects.

- **New**
  This command creates a new object even if an object with the same identification already exists.
- A **number**
  If a number is entered in the column “Command”, a number of objects will be created by one row. For example, if user enters the value “10” in the command column and “Object1” in the column “Object Identification”, ten objects called “Object1”, “Object2”,... will be created.
- **Ignore**
  This row will be ignored during save operations (comment line).
- **Filter**
  This special command is used during data retrieval (see Loading Data into the Default Data Area).

Aspect Commands

Aspect commands can only be performed interactively as described below.

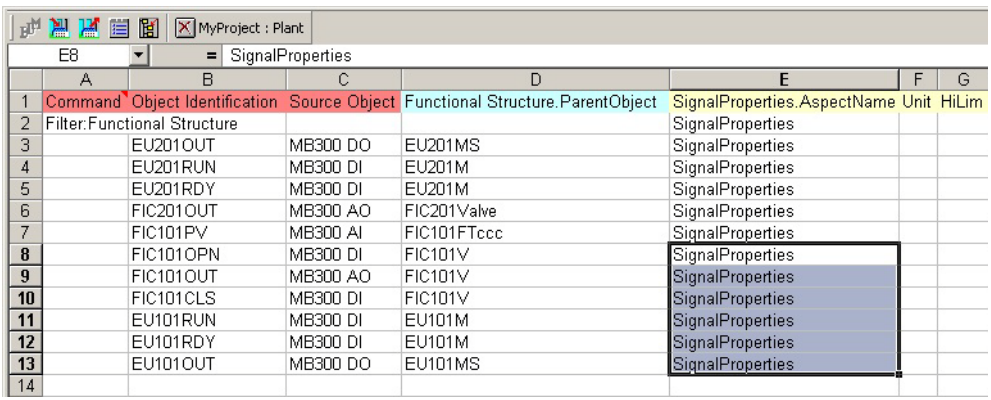
- Identify one or multiple aspects by selecting cells related to properties of the aspects.
- Open the context menu and perform menu entry “**Aspect Commands**”.
- The commands offered are:
  - **Delete -** Deletes immediately one or several aspects.

With the aspect command “Delete”, the aspect is deleted in System 800xA platform. However, the cells in Excel keep their information.

  - Aspect type specific commands like “**Open**” or “**Print**”.

– If several aspects are selected, the aspect commands offered represent the commands common to all aspects.

Aspect commands “Open Properties” and “Print Properties” for aspect system “DM & PM Application” do not display always the entire data in the aspect system’s user interface when executed within Bulk Data Manager workbooks. An error message can appear and aspect data are displayed incompletely.
Use the Engineering Workplace’s aspect context menu to execute these aspect commands. Or use the filter feature of the aspect system’s user interface to set the structure e.g. “Functional Structure” where the aspects are located.



	A	B	C	D	E	F	G
1	Command	Object Identification	Source Object	Functional Structure.ParentObject	SignalProperties.AspectName	Unit	HiLim
2	Filter:Functional Structure				SignalProperties		
3		EU201OUT	MB300 DO	EU201MS	SignalProperties		
4		EU201RUN	MB300 DI	EU201M	SignalProperties		
5		EU201RDY	MB300 DI	EU201M	SignalProperties		
6		FIC201OUT	MB300 AO	FIC201Valve	SignalProperties		
7		FIC101PV	MB300 AI	FIC101FTccc	SignalProperties		
8		FIC101OPN	MB300 DI	FIC101V	SignalProperties		
9		FIC101OUT	MB300 AO	FIC101V	SignalProperties		
10		FIC101CLS	MB300 DI	FIC101V	SignalProperties		
11		EU101RUN	MB300 DI	EU101M	SignalProperties		
12		EU101RDY	MB300 DI	EU101M	SignalProperties		
13		EU101OUT	MB300 DO	EU101MS	SignalProperties		
14							

Figure 24. Performing Aspect Commands

Property Commands

- **#NULL**
  Writing #NULL to a property value instruct the Bulk Data Manager to write an empty value (null-value) to System 800xA applications.

Auto-Update Data Area

Besides the Default Data Area there can be one to many Auto-update Data Areas per Worksheet. An Auto-update Data Area updates itself automatically according to user defined criteria whenever the Workbook is opened, printed or on the user’s demand. As a prerequisite, the workbook must be attached to a System and a Start Object (see Setting System and Start Object).

Auto-update Data Areas are especially useful for Excel files attached to Aspect Objects as Document Aspects.

Configuring an Auto-update Data Area

To create an Auto-update Data Area, follow the steps below.

1. Create a Default Data Area as described in Configuring the Default Data Area.
2. Change a Default Data Area to an Auto-update Data Area using context menu item “Make Auto-update Data Area”, see User Interface.
3. Change the filter to define which data should be loaded into the data area (see Configuring the Default Data Area). Enter a structure after the keyword “Filter:” as for example “Filter:Location Structure”. “Filter:Functional Structure” will be set automatically during change performing “Make Auto-update Data Area”. By accepting that only data of the related object and all child objects below in the Functional Structure are considered.
4. Change the comment in the “Command” Cell (context menu “Edit Comment”) from:
   - $$ROW$$CFG1
     to
   - $$ROW**m**$$CFG1
     where “**m**” is an optional limitation of the number of rows to be retrieved. If more data rows are retrieved than specified the remaining rows will be cut. Drop the number after “$$ROW” in order to define an unlimited data area.

Changing the comment from “$$STD_CFG” to “$$CFGn” by performing context menu item “Make Auto-update Data Area” changes the data area from a Default Data Area (which is updated by drag and drop) to an Auto-update Data Area that updates itself automatically.

- Repeat the steps above for each needed Auto-update Data Area.

Figure 25 shows two Auto-update Data Areas.

The first Auto-update Data Area “CFG1” is limited to 5 rows and it shows objects from the Functional Structure having aspects called “SignalProperties”.

The second Auto-update Data Area “CFG2” is not limited in the number of rows displayed. It shows objects from the Functional Structure having aspects called “ArticleData”.

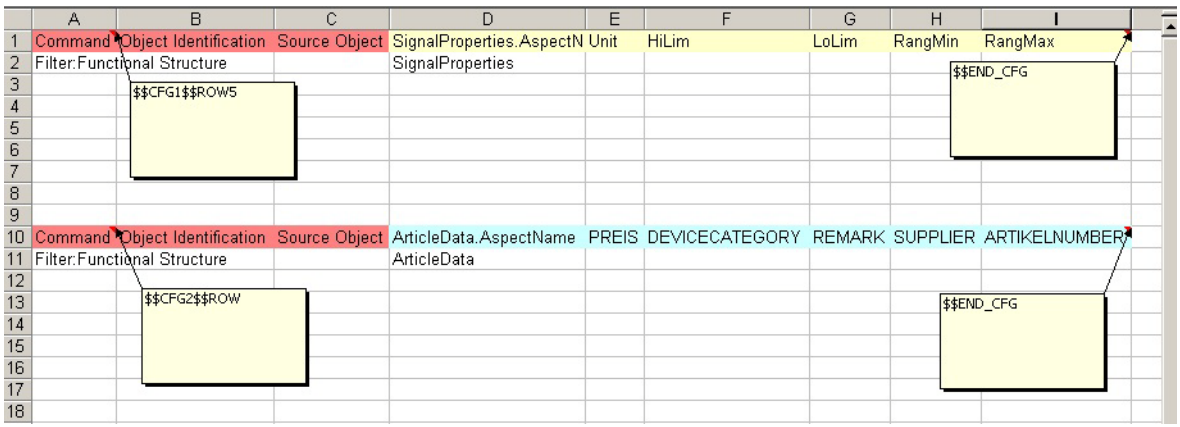


Figure 25. Auto-update Data Areas

Updating an Auto-update Data Area

An Auto-update Data Area updates itself when a workbook is opened or printed. To prevent updating during opening a workbook write the keyword **$$NoUpdate** as comment to cell “Command” of the headline. This comment has to be written to each headline of Data Areas belonging to the active sheet during opening the workbook which should not be updated.

Moreover the data area can be updated on the users’s demand by performing the menu:

- **Bulk Data Manager > Update Selection** (or keys **CTRL + Shift + U**)
  Only the selected data areas are updated.
- **Bulk Data Manager > Update All** (or keys **CTRL + Shift + A**)
  All Auto-update Data Areas are updated.

After performing these menus first the current data of the Auto-update Data Area will be cleared before updating.

If user writes the keyword **$$VisColOnly** as comment to cell “Command” of the headline only those columns will be cleared before updating which are not hidden (Excel column context menu “Hide”).

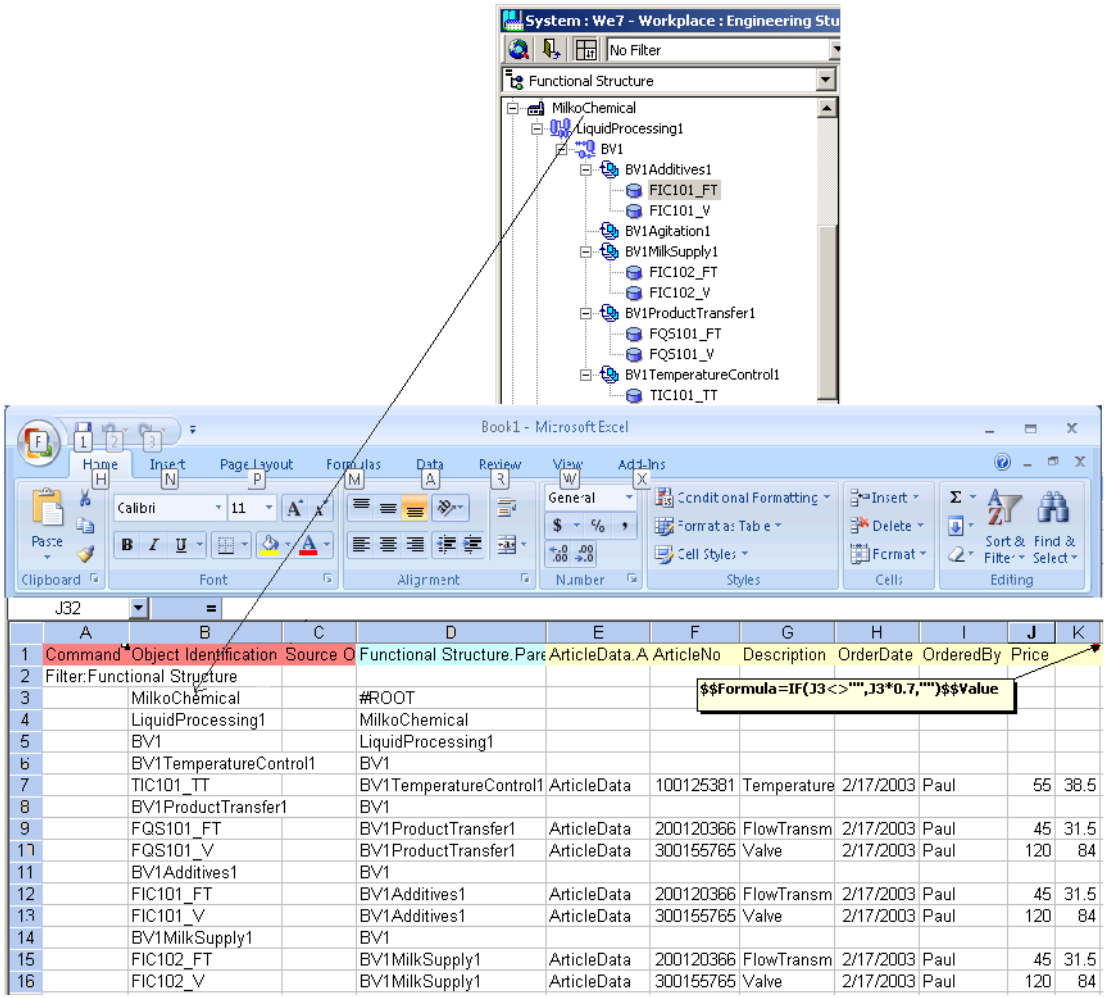
Auto-update Data Areas are updated relative to the Start Object and System this workbook is attached to.

Saving Data in an Auto-update Data Area

To save the data stored in Microsoft Excel into System 800xA application perform the menu **Bulk Data Manager > Save** (or keys **CTRL + Shift + S**).

Using Formulas Within a Data Area

Within the Bulk Data Manager formulas can be applied to the data dropped into a sheet. The calculations are performed in additional columns containing MS Excel formulas in comments as depicted in Figure 26. Notice that calculations can be applied to Default Data Areas and Auto-update Data Areas. For more information on how to load data from other System 800xA applications see Loading Data into the Default Data Area and Updating an Auto-update Data Area.



	A	B	C	D	E	F	G	H	I	J	K
1	Command	Object Identification	Source O	Functional Structure.Pare	ArticleData.A	ArticleNo	Description	OrderDate	OrderedBy	Price	
2	Filter:Functional Structure										
3		MilkoChemical		#ROOT							
4		LiquidProcessing1		MilkoChemical							
5		BV1		LiquidProcessing1							
6		BV1TemperatureControl1		BV1							
7		TIC101_TT		BV1TemperatureControl1	ArticleData	100125381	Temperature	2/17/2003	Paul	55	38.5
8		BV1ProductTransfer1		BV1							
9		FQS101_FT		BV1ProductTransfer1	ArticleData	200120366	FlowTransm	2/17/2003	Paul	45	31.5
11		FQS101_V		BV1ProductTransfer1	ArticleData	300155765	Valve	2/17/2003	Paul	120	84
11		BV1Additives1		BV1							
12		FIC101_FT		BV1Additives1	ArticleData	200120366	FlowTransm	2/17/2003	Paul	45	31.5
13		FIC101_V		BV1Additives1	ArticleData	300155765	Valve	2/17/2003	Paul	120	84
14		BV1MilkSupply1		BV1							
15		FIC102_FT		BV1MilkSupply1	ArticleData	200120366	FlowTransm	2/17/2003	Paul	45	31.5
16		FIC102_V		BV1MilkSupply1	ArticleData	300155765	Valve	2/17/2003	Paul	120	84

Figure 26. Using Formulas within a Data Area

The example above shows how the user can calculate an internal price based on a 30% discount of a list price.

- Insert a new column within the Data Area
- Write the formula as a comment to the configuration headline of that column. Use the key word **$$Formula** followed by a MS Excel formula e.g. **=IF(J3<>"",J3*0.7,"")**. Any MS Excel formula can be used.
  Notice that cell addresses (here "J3") have to be set to the first row number where data are written to (here row number 3).
- If the value should be saved in the calculated column instead of the formula add the keyword **$$Value** additionally to the comment.

After reading data from the different System 800xA applications to the Data Area all formulas of the configuration headline (more than one is possible) will be copied to each data row (in this case row 3 up to 16). MS Excel then calculates automatically the values. If the keyword **$$Value** is found all formulas of that column will be replaced by their calculated values.

The Excel formula "OFFSET" is used for referencing the cells dynamically. An example is shown below with reference to Figure 27. The BDM formula (applied on column H) is used to subtract content of "Min" column from "Max" column for every row. In the formula, every cell of the particular row is referenced using the "OFFSET" function. An example of the formula used in the excel sheet is as follows:

$$Formula=OFFSET(F1,ROW()-1,0)-OFFSET(G1,ROW()-1,0)

Syntax:

OFFSET(Reference cell, Row offset, Column Offset)

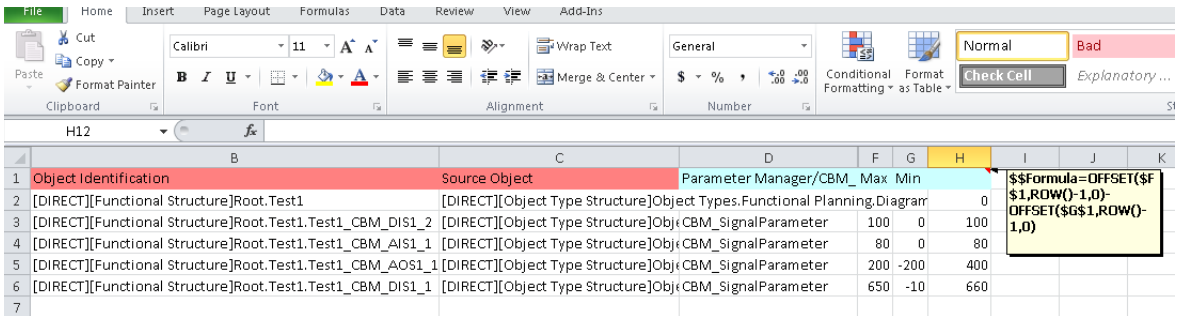


Figure 27. Referencing the cells dynamically

Track Changes Between Data Areas

Function Track Changes allows the user to compare two Data Areas in order to identify changes. Those changes can be modified property values, created and deleted objects. The Data Areas needed to track changes can be located in different Workbooks or in the same Workbook but on different sheets.

Track Changes doesn't support worksheets containing **Structured Properties** or **vertical** headline configurations.

To decide which Data Area is the New Version (i.e. changes has to be stored to System 800xA applications) and which one is the Old Version (i.e. current version of System 800xA applications) use the following context menu:

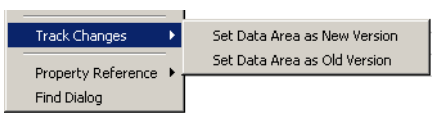


Figure 28. Bulk Data Manager Context Menu Track Changes

As a result of this setting the context menu will be changed:

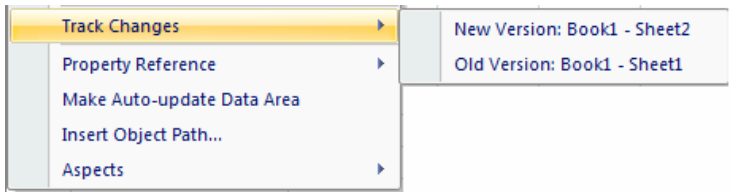


Figure 29. Bulk Data Manager Context Menu Track Changes

To start the comparison of both Data Areas activate Start Track Changes in the Bulk Data Manager toolbar or menu:

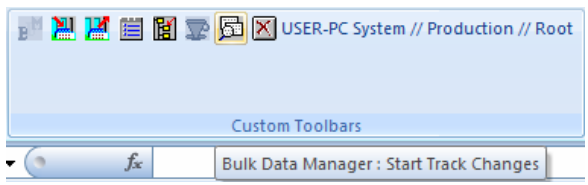


Figure 30. Bulk Data Manager Toolbar Start Track Changes

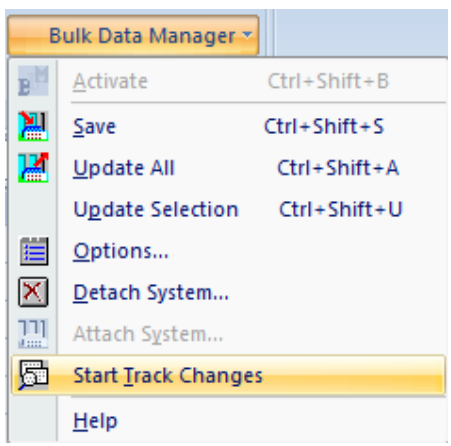
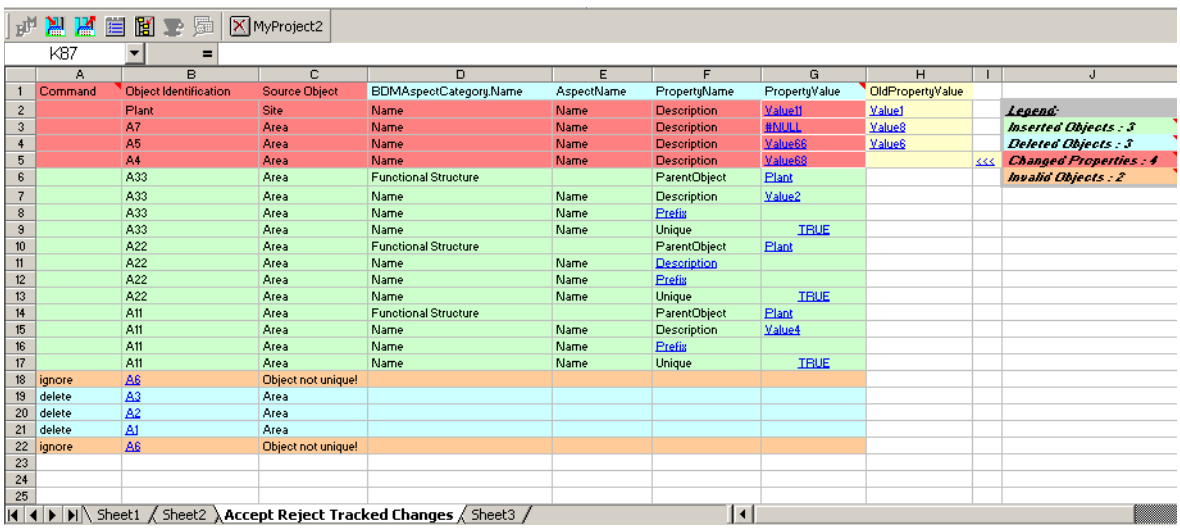


Figure 31. Bulk Data Manager Menu Start Track Changes

The result of comparison is shown in Figure 32. All changes found will be generated in a new Worksheet called Accept Reject Tracked Changes. The contents of this sheet is a Data Area and can be saved through Bulk Data Manager Save function to System 800xA applications. The Data Area based on the Display rules for Aspect Properties (Vertical using 4 attributes) refer to Bulk Data Manager Options.



	A	B	C	D	E	F	G	H	I	J
1	Command	Object Identification	Source Object	BDMAspectCategory.Name	AspectName	PropertyName	PropertyValue	OldPropertyValue		
2		Plant	Site	Name	Name	Description	Value11	Value1		Legend:
3		A7	Area	Name	Name	Description	#NULL	Value8		Inserted Objects : 3
4		A5	Area	Name	Name	Description	Value66	Value6		Deleted Objects : 3
5		A4	Area	Name	Name	Description	Value68		<<<	Changed Properties : 4
6		A33	Area	Functional Structure		ParentObject	Plant			Invalid Objects : 2
7		A33	Area	Name	Name	Description	Value2			
8		A33	Area	Name	Name	Prefix				
9		A33	Area	Name	Name	Unique	TRUE			
10		A22	Area	Functional Structure		ParentObject	Plant			
11		A22	Area	Name	Name	Description				
12		A22	Area	Name	Name	Prefix				
13		A22	Area	Name	Name	Unique	TRUE			
14		A11	Area	Functional Structure		ParentObject	Plant			
15		A11	Area	Name	Name	Description	Value4			
16		A11	Area	Name	Name	Prefix				
17		A11	Area	Name	Name	Unique	TRUE			
18	ignore	A6	Object not unique!							
19	delete	A3	Area							
20	delete	A2	Area							
21	delete	A1	Area							
22	ignore	A6	Object not unique!							
23										
24										
25										

Figure 32. Bulk Data Manager Sheet Accept Reject Tracked Changes

Identified changes in detail

- **Inserted Objects**
  All objects of the New Version of Data Area which are not found in the Old Version will be inserted.
- **Deleted Objects**
  All objects of the Old Version of Data Area which are not found in the New Version will be deleted.
- **Changed Properties**
  All objects found in both Versions will be checked for differences in their Aspect Properties. If differences were found the objects with their new

property values (New Version) will be written to the Data Area. If the property value is changed to an empty value #NULL will be written to the Data Area (see also Property Commands in Section Saving Data to System 800xA Applications). Additionally the old property values (column OldPropertyValue) will be shown outside the Data Area (i.e. they will not be stored if BDM-Save is executed).

- **Invalid Objects**
  If multiple objects were found in the Old Version which are not unique an error will be generated and no check will be done for property differences.

In addition to all changes written to the Data Area, hyperlinks will be created to the Data Areas of the Old and the New Version. That allows the user to see the differences in its context (e.g. parent - child relationships). In the end a legend gives a view about all changes.

Setting System and Start Object

For Save operations, the Bulk Data Manager needs to know into which System the data shall be saved. For updating Auto-update Data Areas and for resolving relative Property References, the Bulk Data Manager needs a Start Object.

The currently selected System, Environment, and Start Object are shown in the Bulk Data Manager toolbar.

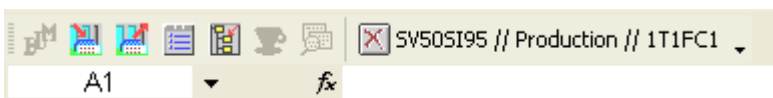


Figure 33. System and Start Object in Toolbar

There are several ways how to attach a Workbook to a System and how to set the start object:

- The first object that is dragged into the Default Data Area sets the System and Start Object (if not already set). This method was used in Working with the Default Data Area.

- If the Bulk Data Manager is started from a Document Aspect or via the context menu “Advanced > Bulk Data Manager” of the object browser in the Plant Explorer, System and Start Object are set automatically to the selected object.
- Finally, System, Environment and Start Object can be set within the Bulk Data Manager using the “Attach System” menu entry or the toolbar. The Dialog shown in Figure 34 pops up.

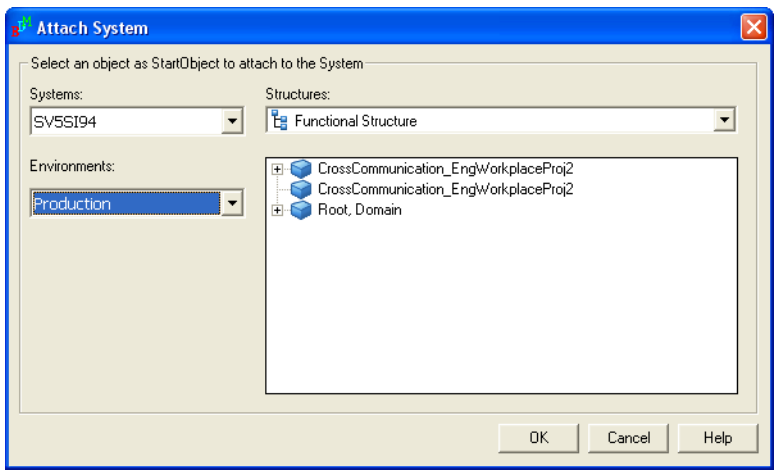


Figure 34. Select System and Start Object Dialog

Object from any system can be dragged (and loaded) into the Default Data Area. However, saving the Default Data Area saves the data always into the attached system. In addition it is possible to load and save data of several Data Areas from and to different systems at the same time. How to define an individual system and start object for a Data Area is described in Bulk Data Manager Filter.

The attached System and Start Object is stored as system and object ids (GUIDs) in the workbook. When the workbook is transferred to a different Aspect Server, these ids are not valid anymore. In this case, a warning is shown and the workbook is detached.

If attached Aspect Object has been deleted while opening a workbook an error message box “11030 Internal BDM error! ...”appears after opening an existing BDM workbook. After closing the error message box the Bulk Data Manager worksheet needs to be Detached and Attached with an existing Aspect Object.

Bulk Data Manager Options

Select Options in Bulk Data Manager Menu to open the Options dialog.

Bulk Data Manager Options (General Tab)

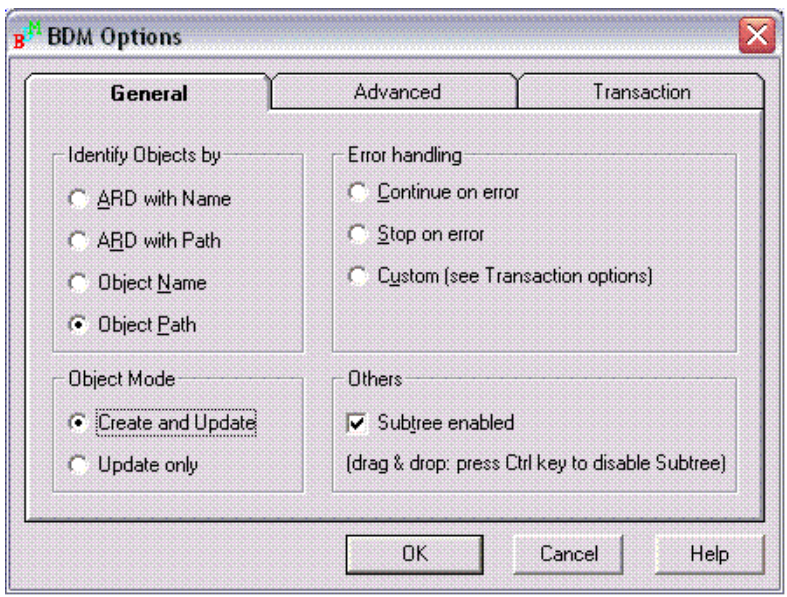


Figure 35. Bulk Data Manager Options Dialog (General tab)

- Section **“Identify Objects by”** specifies what type of object identifications shall be generated for the columns “Object Identification” and “ParentObject”. The default is “ARD with Name”, but user can also use plain names or object paths. Table 4 shows examples for the different object identification types. See also Object Identification, for more information.

- The options **“Error handling”** determine what to do in case of an error:
  - If “Continue on error” is selected, the Bulk Data Manager will ignore errors and proceed with saving data.
  - If “Stop on error” is selected, the Bulk Data Manager will stop processing on the first error and roll-back the last transaction.
- The option **“Subtree enabled”** defines if only the data of the selected object will be loaded into the Worksheet or if the data of all its child objects will also be included. By default, **“Subtree enabled”** is active.

Table 4. Options of Section “Identify Objects by”

Option	Example Id	Description
ARD with Name	=A.1	Plain ARD
	=A.1{}Obj1	Combined ARD and name if object has no designation
	Obj1	Plain name if object and all parents have no designations
ARD with Path	=A.1	Plain ARD
	=A.1{}Obj1/Obj2	Combined ARD and relative path if object has no designation.
	Obj1/Obj2	Object path if object and all parents have no designations.
Object Name	Obj1	Plain object name.
	Obj1.2	Plain object name. No escaping for special characters. The dot is part of the name, not a separator.
Object Path	Obj1.Obj2	Object path (min. path)
	[Direct][Functional Structure]Obj1.Obj2	Full path
	Obj1.Obj1\.2	Escaping is needed for special characters.

Remark: The separator "{}" is used to switch from ARD to name or path.

ARD + Name : =A.1{}Obj2

ARD + Relative Path: =A.1{}Obj1/Obj2

Section "Object Mode" allows to choose between Option "Create and Update" and "Update". These options determine the behaviour of BDM when the data is saved through the menu item **Bulk Data Manager > Save** or BDM toolbar button **Save All Objects**.

The option "Create and Update" (default) creates non-existing objects and updates them. The option "Update" assumes all objects are already existing and updates them. If in the latter case an object does not exist an error message "Cannot find Object for update" is given.

Section "Error handling" provides pre-configured sets of the Transaction Error flags in the Transaction tab.

Option "Continue on error" ignores all errors (the transaction is not aborted) and writes as much data into the platform as possible. Errors are listed in the error sheet. However, in case of errors, objects and aspects may become incomplete and inconsistent. Option "Continue on error" is the same as options "Object/Aspect/Property Error" disabled and option "Continue with next transaction" enabled in section "Abort Transaction on".

Option "Stop on error" lets the Bulk Data Manager stop on any error immediately and the last transaction is aborted (rolled back). Option "Stop on error" is the same as options "Object/Aspect/Property Error" enabled and option "Continue with next transaction" disabled in section "Abort Transaction on".

Option "Custom" indicates that the user has changed the detailed transaction options in the Transaction tab.

Option "Subtree enabled" in section "Others" indicates whether on Drag and Drop only the dropped object or the dropped object with all its child objects shall be exported.

Use CTRL+Drop to override the set option "Subtree enable".

When using object identification with ARDs:
Do not mix up Prefix of designation aspect for Functional Structure (==) with Prefix of designation aspect of Control Structure (=). Otherwise this can result in misplaced objects.

Do not use changing prefixes in a structure else use object identification without ARD.

If reading Aspect Objects with “ARD with Path” mode and with “Object Path” mode for the option “Identify Objects by”, different path separators are used. Mode “ARD with Path” uses ‘/’ as separator (for example =A1.2{ }obj1/obj2). The option “Object Path” uses ‘.’ as separator (e.g. Obj1.Obj2). Both separators are recognized correctly when writing objects into the Platform.

While this is mainly a cosmetic issue, there is a problem when instantiating composite object types within a Data Area transaction: Bulk Data Manager maintains a name cache because the Name server of 800xA system doesn’t know objects created in a transaction until the transaction is committed. Bulk Data Manager adds all names to the name cache as they appear in the Excel workbook.

For composite objects types and when copying objects with children, Bulk Data Manager also adds all child objects of the created or copied object to the name cache. In this case, Bulk Data Manager creates paths always using the ‘.’ separator.This will conflict with paths that were generated by the Bulk Data Manager in “ARD and Path” mode resulting in “Object not found” errors.

For the “instantiating composite object types within a Data Area transaction” problem change the separator from ‘/’ to ‘.’ in the paths.

Use “By Object” transactions because the name cache is not needed here.

- In case an aspect such as a DM-PM aspect and the general properties aspect has a property with:
  - Data type field set to String.
  - Value field containing a number.

  Select the Treat all incoming data as Text check box in the General tab of the BDM Options dialog, prior to dropping the aspect on to the BDM sheet. This is to ensure that the string does not get converted to a number.

Bulk Data Manager Options (Advanced Tab)

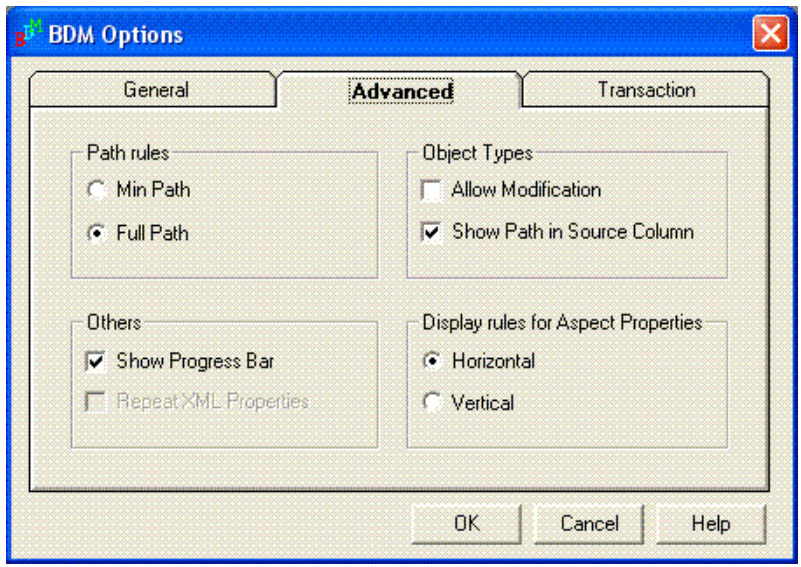


Figure 36. Bulk Data Manager Options Dialog (Advanced tab)

Section “Path rules” determines whether Bulk Data Manager shall generate minimum paths (shortest but unique path) or full paths. Generating minimum paths takes longer than generating full paths.

The option “Allow Modification” in section “Object Types” must be set in order to be able to modify the Object Type Structure (create/modify/delete Object Types).

When the option “Show Path in Source Column” is set then Bulk Data Manager generates a path (according to the Object Identification rules) for the Object Type in the column “Source Object”. If it is not enabled, the plain Object Type name is generated regardless the Object Identification settings.

By switching off option “Show Progress Bar” in section “Others” the performance of Bulk Data Manager can be improved (~10%). However, it is then not possible to stop an import or export of data.
For option “Repeat XML Properties” see Applying Filters.

Section “Display rules for Aspect Properties” controls the configuration of the Default Data Area. If the option is set to “Horizontal” the dragged aspects from the Plant Explorer and all their accessible properties will be dropped into the headline of the Worksheet in a horizontal order. In this case all read data later on will be organized in a record structure where one record represents an object. See also Configuring the Default Data Area.

If the option is set to “Vertical” and an aspect is dropped into the headline of the Worksheet no aspect properties will be written to the headline. Instead common attributes like “PropertyName” and "PropertyValue" will be set. By this an object read from the system will be shown in more than one record and the aspect properties and their values are organized vertical. Figure 37 shows a Data Area with two aspects Functional Structure, and Name dropped to the Worksheet. Each aspect is organized in 3 attributes. The first attribute contains the aspect category and is part of the headline.

Data retrieved from the Plant Explorer and included in the Worksheet can be filtered according to column “PropertyName” and “PropertyValue”. That is the main benefit to data organized in a horizontal manner where the property names are fixed as part of the headline and cannot participate in filtering data.

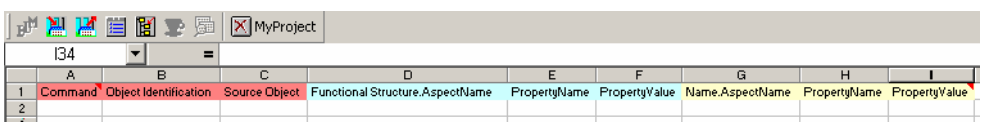


Figure 37. Configure Default Data Area (Display Rules Vertical - 3 Attributes)

A second possibility to show aspect properties and their values in a vertical manner is shown in Figure 38. This Data Area will be generated by dragging any aspect from the Plant Explorer and dropping to the Worksheet provided that the CTRL key is pressed in the meantime. The difference to Figure 37 is that an additional common attribute called "BDMAspectCategory.Name" exist. The aspect categories are not part of the headline. They will be shown in column "BDMAspectCategory.Name" if objects are loaded into the Worksheet. Therefore it is possible to use column "BDMAspectCategory.Name" as a filter criteria as well as column "PropertyName". For further information about filtering data see Bulk Data

Manager Filter.

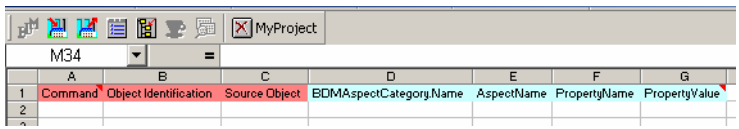


Figure 38. Configure Default Data Area (Display Rules Vertical - 4 Attributes)

Place commands generated by Bulk Data Manager in vertical headline configured worksheets for objects that have multiple structure aspects of the same category cannot be saved. An error message “Ambiguous aspect name for this category.” is shown. The structure aspect cannot be found exactly by using its not unique aspect

Type reference cannot be saved aspect from worksheets with vertical headline configuration. An error message “Can not find aspect category” appears while saving an Aspect Object type instance is saved.

Ignore this error message or enter the “ignore” command into the related cell (type reference aspect) in column “Command” and save again.

Bulk Data Manager Options (Transaction Tab)

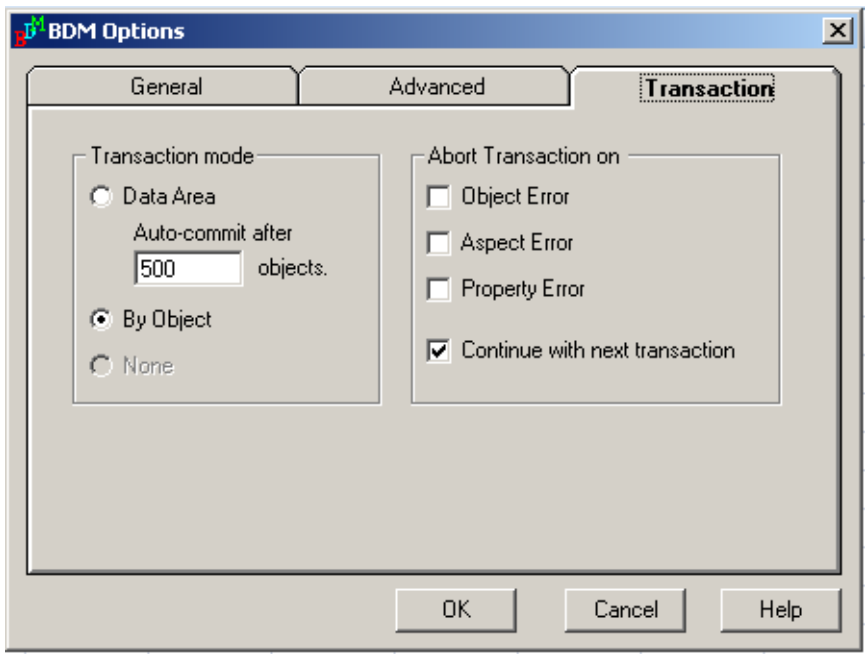


Figure 39. Bulk Data Manager Options Dialog (Transaction tab)

Section “Transaction mode” sets the scope of transactions.

Option “**Data Area**” stands for one transaction per data area. Because large transactions can require excessive memory, automatic commits can be inserted after a certain amount of objects. How many objects have be saved in one transaction can be controlled by the entered number of objects in the edit field. If the number is set to “0” no additional auto-commits will be made.

Do not use Bulk Data Manager transaction mode “Data Area” to copy “AC 800M/C Connect” Aspect Objects containing “Function” aspects. This can lead to unexpected copy results.

Use Bulk Data Manager’s transaction mode “By Object” which is the default setting.

Option “**By Object**” stands for one transaction per Excel row (this means usually one transaction per object). Option “**None**” means no transactions at all.

Section “Abort Transaction on” and options “Object/Aspect/Property Error” determine at which errors a transaction shall be aborted and rolled back. Option “Continue with next transaction” controls if Bulk Data Manager shall continue processing with the next transaction after a transaction was aborted.

The normal user would not change the transaction handling options here but use the “Error handling” options in the General tab that provide pre-configured settings for the “Abort Transaction on...” options. An experienced user, however, can control the complete transaction handling with the provided options. For example an interesting setting could be:

- Transaction mode = By Object
- Abort Transaction on Object/Aspect/Property Error = ON
- Continue with next transaction = ON

With these settings, Bulk Data Manager would import as many objects as possible with reasonable performance, but all objects with errors would be skipped (rolled back) and listed in the error sheet.

For a detailed description of Bulk Data Manager Transaction Handling see Bulk Data Manager Transaction Handling.

Bulk Data Manager Filter

Set the mouse cursor into a cell as part of a data area and select context menu item BDM filter to open the filter dialog. Alternative the filter dialog can be started by Bulk Data Manager toolbar. The filter dialog works on the selected data area and writes back the filter criteria to the data area performing the OK button.

All data areas (Auto-update Data Areas and the Default Data Area) can be set by the filter dialog provided that the dialog is associated with the data area by user selection. That's why the menu items are only visible or active if the cursor is located in a range of a valid data area.

In the first part of the filter dialog the structure type can be selected form a list. Column Category and column PropertyName list all categories and properties of the data area the filter dialog is associated with (is started from). Enter a filter criteria in

column PropertyValue for each combination of category and property. This is possible using the two list boxes and button Add or direct in column PropertyValue.

The example in figure above generates the following result in the data area:

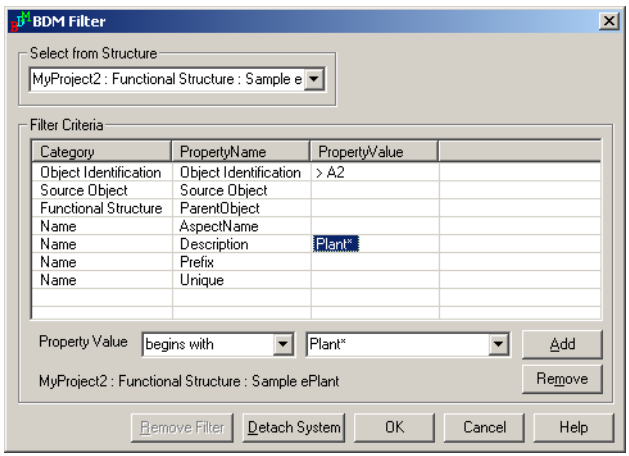


Figure 40. Bulk Data Manager Filter Dialog

	A	B	C	D	E	F	G
1	*Command*	Object Identifi	Source Ob	Functional Structure.Par	Name.AspectName	NAME	DESCRIPTION
2	Filter:MyProject2 : Functional Structure : Sample ePlant	> A2					Plant*

Figure 41. Filter Data

The filter shown in Figure 41 above retrieves only the data of objects located in the Functional Structure having as object names (column Object Identification) strings > “A2” and in property DESCRIPTION strings beginning with “Plant”. If the criteria is specified for more than one property they are in a logical AND operation.

See also Loading Data into the Default Data Area and Configuring an Auto-update Data Area for more information.

In addition to Setting System and Start Object where a central system and start object can be defined for the Bulk Data Manager the user can set an individual

system and start object for a Data Area. The difference between an individual and central setting of a system and start object is that Data Areas having their own systems and start objects load the data from that system and write back and Data Areas without own settings use the central definition of the system and start object. Press the Attach System button to start up the Attach System dialog and define an individual system, environment and start object. If an individual system and start object is already defined it can be deleted by performing the Detach System button.

The attached System and Start Object is stored as system and object ids (GUIDs) in the Data Area (comment of column Command). When the workbook is transferred to a different Aspect Server, these ids are not valid anymore. In this case, a warning is shown and the system and start object has to be reattached.

The example in Figure 41 Filter Data has set an individual system to “MyProject2” and the start object to “Sample plant” in structure “Functional Structure”.

Bulk Data Manager Configure Properties

Drag an aspect from the Plant Explorer and drop it into a cell of the worksheet. During the drop action the dialog Configure Properties appears, showing the names of all normally accessible properties of the selected aspect category in a scroll list with a corresponding check box. Mark the check boxes of all properties that has to be written into the headline of the worksheet and click OK.

Setting / unsetting the check mark in the check box *Select / Deselect all* marks / unmarks all properties in the scroll list.

Some aspect categories have properties which are accessible but not visible. To make those properties visible in the scroll list press the **ALT** key during drag and drop of an aspect category. An example for those aspect properties is aspect Name.

A check mark in the check box *Show basic administrative properties* shows further properties for each Aspect Category like e-signature properties in the scroll list.

A check mark in the check box *Hide write-protected properties* hides properties in the scroll list declared by the aspect system as not writable. If the check box is not marked, these properties are displayed in the headline with a greyed font and have a cell comment containing the text “This aspect property is not writable”.

A check mark in the check box *Ignore *_ref properties* (default) hides properties named according to this scheme in the scroll list, typically these are properties of the Control Properties aspect.

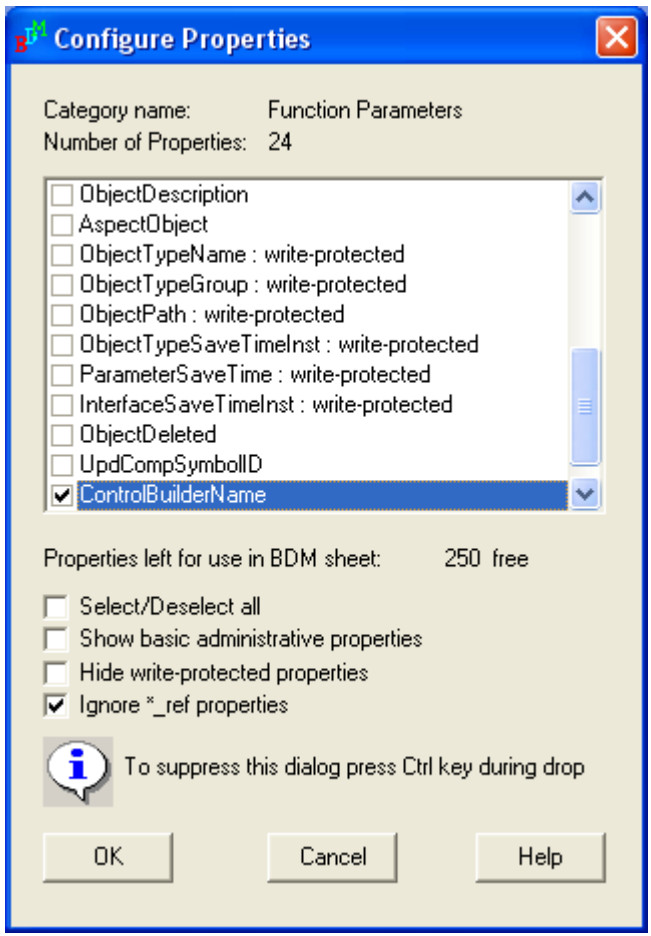


Figure 42. Bulk Data Manager Configure Properties Dialog

For multiple data areas on one worksheet the column font needs to be set to Automatic if there are overlaps for different aspect property columns.

The greyed font indication is not valid for XML based templates.

To avoid this dialog next time press CTRL key during drag and drop of an aspect category. In this case all properties of the selected aspect category will be written directly to the headline of the Worksheet. For more information about configuring the Default Data Area see Configuring the Default Data Area.

Creating new Objects or Paths

Dialog *Insert Object Path* is needed to enter known objects of a project into a Data Area. It is a fast way to write objects in different syntax to an Excel worksheet by selecting the right object from the tree control.

To start the *Insert Object Path* dialog use the Bulk Data Manager context menu or press CTRL key + F2 key alternatively.

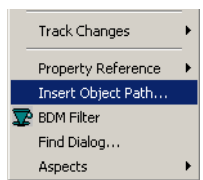


Figure 43. Bulk Data Manager Context Menu Insert Object Path

Note, that the dialog starts up context sensitive concerning the selected Excel cell. If the mouse cursor is located in a Data Area and selects a cell in column *Source Object* the dialog pops up with the Object Type Structure selected. If the column represents a structure (e.g. Functional Structure or Location Structure) the dialog selects the structure concerned.

In addition the project name is shown in the title of the dialog. Note, that each Data Area can have its own system and start object. That's why the dialog *Insert Object Path* synchronizes to the corresponding project of the Data Area if necessary (i.e. the Data Area don't uses the central system and start object of the Bulk Data Manager). For more information about setting an individual system and start object for a Data Area see Bulk Data Manager Filter.

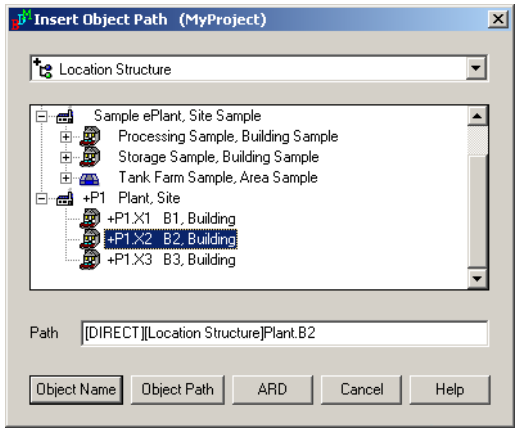


Figure 44. Insert Object Path Dialog

Buttons

- **Object Name**
  Use button *Object Name* to write the name of an object.
- **Object Path**
  Use button *Object Path* to write the object in the *full path* syntax shown in the *Path* field of the dialog.
- **ARD**
  Use button *ARD* to write the ***A****bsolute* ***R****eference* ***D****esignation* of an object.

For a detailed description of the object syntax see Bulk Data Manager Options.

Bulk Data Manager Transaction Handling

Bulk Data Manager usually saves its data in transactions. A transaction means that the Bulk Data Manager builds up and checks all data to be saved in memory and then saves (commits) this data at once to the Aspect Server. In case of an error, the transaction is rolled-back thus all objects involved in the transaction will remain as they were before the transaction started.

Transactions improve the processing speed dramatically and ensure that the system keeps a consistent state in case of errors.

Bulk Data Manager supports three transaction modes:

- Data Area: The whole data area is written within one transaction. Auto-commits will be made in between by the Bulk Data Manager as specified in order to reduce the memory consumption.
- By Object: All aspects/properties of one object (= one row of a data area) is written within one transaction. The transaction is committed at the end of the row and a new transaction is created for the next row (next object).
- None: No transaction is created at all. Each single property is written into the system.

Do not use Bulk Data Manager transaction mode “Data Area” to copy “AC 800M/C Connect” Aspect Objects containing “Function” aspects. This can lead to unexpected copy results.

There are three options that control when a transaction is aborted:

- Abort on object error: If active, a transaction is aborted when an object could not be created (or deleted). If not active, the transaction is not aborted but all other data for this object (aspects and properties) are ignored.
- Abort on aspect error: If active, a transaction is aborted when an aspect could not be created. If not active, the transaction is not aborted but all properties for this aspect are ignored.
- Abort on property error: If active, a transaction is aborted when a property could not be written. If not active, the transaction is not aborted. A read-only property does not generate an error when writing (otherwise it would not be possible to export and re-import data with read-only properties easily).

The option 'Continue with next transaction' specifies how to proceed when a transaction was aborted, i.e. an error occurred and the option 'Abort on Object/Aspect/Property error' matching the error is active. If the 'Continue with next transaction' is not enabled, Bulk Data Manager stops all further processing. If the option is selected, Bulk Data Manager continues with the next transaction.

The following table demonstrates the Bulk Data Manager processing behavior when a property error occurred depending on the transaction mode and the error options:

Table 5. Bulk Data Manager Processing Behavior

Transaction mode	Abort Transaction on Property Error	Continue with next transaction	Behavior
Data Area	0	1	Error is ignored. Proceed with next property.
By Object	0	1	Error is ignored. Proceed with next property.
None	0	1	Error is ignored. Proceed with next property.
Data Area	1	1	Abort data area transaction. Proceed with next data area.
By Object	1	1	Abort object transaction. Proceed with next row.
None	1	1	No transaction to abort. Proceed with next property.
Data Area	0	0	Error is ignored. Proceed with next property.
By Object	0	0	Error is ignored. Proceed with next property.
None	0	0	Error is ignored. Proceed with next property.
Data Area	1	0	Abort data area transaction and stop BDM processing.
By Object	1	0	Abort object transaction and stop BDM processing.
None	1	0	No transaction to abort. However, BDM processing is stopped.

The transaction mode together with the error options allow the user to control if Bulk Data Manager should write as much information into the system as possible (i.e. ignoring any errors) or to write only objects or data areas that can be written without errors altogether.

Since the error processing and its options are quite complex, these options are grouped together for the end user. See Bulk Data Manager Options for details.

Understanding the Configuration Headline

This chapter describes the details of the configuration headline and is intended for advanced users only that want to configure a data area manually.

The configuration headline of a data area is structured as shown in Figure 45.

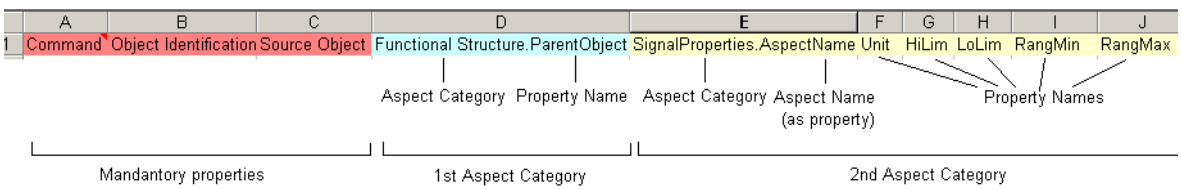


Figure 45. Structure of Configuration Headline

The first three properties are mandatory. These properties are related to the object:

- Command: Special commands used when saving objects (e.g. delete, place, ...). See Saving Data to System 800xA Applications.
- Object Identification: Name, path, or ARD to the object.
- Source Object: Object Type or source object for copying.

Do not delete the mandatory object properties. They are described in detail in the following chapters.

The following columns describe the aspect categories to be loaded or saved. The first column of each category starts with the Aspect Category name followed by a first property name. The following columns of this aspect category list the properties. It depends on the aspect system, which properties are available for a

certain aspect category (the aspect system must publish properties by implementing a certain interface).

In addition to the properties published by the aspect system, the Bulk Data Manager provides special properties for aspect categories in order to make information available that would be hidden otherwise. These special properties are listed in Table 6. Some properties are available under more than one name.

Table 6. Special Properties for all Aspects

Property name	Access	Description
AspectName	r/w	Name of aspect. The name is also used to identify the aspect, e.g. if more than one aspect of an aspect category exists for one object. Property is generated by Drag and Drop.
AspectDescription	r/w	Description of aspect.
AspectGUID AspectID	r	Aspect ID.
AspectModificationTime #AMT	r	Modification time of aspect. The time is formatted in Excel as “Time”. User can change the formatting, e.g. to display the date only.
AspectCreationTime #ACT	r	Creation time of aspect. The time is formatted in Excel as “Time”. User can change the formatting, e.g. to display the date only.
AspectCreationUser #ACU	r	Name of user who created the aspect.
AspectModificationUser #AMU	r	Name of user who last modified the aspect.

For structure aspect, the following properties are available in addition:

Table 7. Special Properties for Structure Aspects

Property name	Access	Description
ParentObject	r/w	Parent object of this object in a given structure. Depending on the options, name, path or ARDs are generated. Used to represent the object structure in Excel. Property is generated by Drag and Drop.
ParentARD	r/w	Absolute Reference Designation of the parent object of this object in a given structure.
ParentName	r/w	Name of parent object.
ParentPath	r/w	Path of parent object.
ParentID	r/w	Object ID of parent object (more exactly: ID of parent node).
Object Identification	r/w	Location of this object in the given structure. Depending on the options, name, path or ARDs are generated
ObjectPath	r/w	The path of the object specified by this structure aspect.
ObjectARD	r/w	Absolute Reference Designation of this object.
ObjectName	r/w	Name of this object.

There are also special properties for accessing data related to the object itself. These properties are made available to special “aspect category” named “#Object”. E.g.,

"#Object.ObjectCreationTime" can be used to access the object creation time. The following properties are available for an object:

Table 8. Object Properties

Property name	Access	Description
ObjectName	r/w	Name of the object.
ObjectDescription	r/w	Description of object.
ObjectGUID ObjectID	r	Object ID.
ObjectModificationTime #OMT	r	Modification time of object. The time is formatted in Excel as "Time". User can change the formatting, e.g. to display the date only.
ObjectCreationTime #OCT	r	Creation time of object. The time is formatted in Excel as "Time". User can change the formatting, e.g. to display the date only.
ObjectCreationUser #OCU	r	Name of user who created the object.
ObjectModificationUser #OMU	r	Name of user who last modified the object.

The second special aspect category "#Node" represents the structure aspect that is found by resolving the object identification in the column "Object Identification" (2nd column in the header). #Node is useful for identifying the exact location of an object in a structure especially if the object is placed multiple times in the same structure. The properties shown in Table 6 and Table 7 are available.

The configuration headline is marked by special Excel comments:

- The first column ("Command") is marked by the comment: $$STD_CFG$$ROW
- The last column ("RangeMax" in this example) is marked by: $$END_CFG

Do not delete these comments. The details are described in the following chapters.

Object Identification

Bulk Data Manager identifies objects and their locations in structures by names, paths, Absolute Reference Designations or object ids (GUIDs).

Table 9. Object Identification Modes

Mode	Example	Description
Name	Obj1 Obj1.1	Plain object name. Special characters must not be escaped. Plain names are not recommended because they lead very likely to ambiguous object identifications.
Min. Path	Obj1.Obj2 Obj1.Obj1\.1	The shortest but unique path for the object. The full Structure and Name Server syntax can be used. Special characters must be escaped. Use Min. Path as default if designations are not used in the system. Min. paths may become ambiguous if new objects are added to the system.
Full Path	[Direct] [Functional Structure]Root.Obj1.Obj2	The full path for the object including structure information. The full Structure and Name Server syntax can be used. Special characters must be escaped. Full path are almost always unique.
ARD with Name	==N1.C1 ==N1.C1{}Hardware Obj1	ARD with optional name if the object has no designation. Results in an object name if the object and all parents have no designation. Special characters must not be escaped in the name part. This is the default setting in the option dialog.

Table 9. Object Identification Modes (Continued)

Mode	Example	Description
ARD with Path	==N1.C1 ==N1.C1{}Hardware/LocalIO Obj1/Obj2	ARD with optional relative path if the object has no designation. Results in an object path if the object and all parents have no designation. Special characters must be escaped in the name part. Use ARD with Path if ARD with Name leads to ambiguous object identifications.
Id	{AED5C833-A0B8-11D3-BA34-0008C7A34A08}{AED5C835-A0B8-11D3-BA34-0008C7A34A08}	Object and Structure Aspect Id. IDs are always unique. Use Ids if existing structures shall be modified and there are problems with ambiguous objects or aspects.

Which object identification mode is actually used depends on the properties in the configuration headline of a data area and on the Bulk Data Manager option settings.

Table 10. Effective Object Identification Mode

Property	Mode
Object Identification ParentObject	Depends on Option settings. See Bulk Data Manager Options.
ObjectARD ParentARD	**ARD with Name** or **ARD with Path** depending whether an object identification mode with Name or Path is selected in the Options.
ObjectName ParentName	Name
ObjectPath ParentPath	**Min. Path** or **Full Path** depending on the “Path rules” Options.

Table 10. Effective Object Identification Mode (Continued)

Property	Mode
ObjectId ObjectGUID ParentId AspectId AspectGUID	Id
Source Object	Depends on Option settings. See Bulk Data Manager Options.

You can change the object identification mode by modifying the property in the configuration headline of a data area manually. This applies for the special, second column “Object Identification” as well (e.g. you can use ObjectARD or ObjectId here). Using Drag and Drop, the Bulk Data Manager always creates “Object Identification” and “ParentObject”.

Object Identification for Object Creation

When an object is created it is named and, optionally, placed in a structure directly at creation time depending on the object identification mode.

Table 11. Naming and Placing of Objects at Creation Time

Mode	Example	Object Name	Structure
Name One-element Path	Obj1	Set to name, e.g. Obj1.	Not placed in structure.
Min. Path Full Path	Obj1.Obj2	Set to last path component, e.g. Obj2.	Placed as child of Obj1.
ARD	==N1.C1	Designation is set (created) but no name, e.g. Control Designation set to C1, Prefix ==.	Placed as child of ==N1 in Control Structure.

Table 11. Naming and Placing of Objects at Creation Time (Continued)

Mode	Example	Object Name	Structure
ARD with Name	==N1.C1{}Hardware	Set to name part, e.g. Hardware. No designation is set.	Placed as child of ==N1.C1 in Control Structure.
ARD with Path	==N1.C1{}Hardware/LocalIO	Set to last path component, e.g. LocalIO. No designation is set.	Placed as child of ==N1.C1{}Hardware in Control Structure.
Id	{AED5C833-A0B8-11D3-BA34-0008C7A34A08}{AED5C835-A0B8-11D3-BA34-0008C7A34A08}	No name is set.	Not placed in structure.

Some object types require placing the object directly at creation time into a structure; e.g. Network objects must be placed into the Control Structure otherwise object type constraints will prevent the object creation.

Bulk Data Manager cannot create objects on root level that must be placed into a certain structure.
If you try to create an object on root level whose type definition forces that the object must be placed in a certain structure (for example ‘Network’ objects must be in the ‘Control Structure’), creation of the object fails with error message ‘Can’t create an instance of the object type’
(error code E_AFW_OT_NOT_OUTSIDE_STRUCTUR ES).

As described, a path or ARD must be used in the “Object Identification” column in order to place the object into a structure directly at creation time.

Placing an object at root level of a structure is not possible because the Bulk Data Manager can’t retrieve the parent object of a path such as “[Functional Structure]Obj1” or “=A1”.

As a work around, you can create the object and place it under a helper object and then move it to root level.

Data Exchange with Other Applications

The Bulk Data Manager can be used in order to export or import data from System 800xA applications to other, non System 800xA applications.

- **Export Data** to other applications
  Data stored in System 800xA applications can be retrieved into a Workbook and saved from there to different formats like “character separated values (csv)”, “html”, etc. (see Microsoft Excel save formats).
- **Import Data** from other applications
  - Data can be imported into Microsoft Excel from a variety of data sources (Menus **Office Button > Open** or **Data > Get External Data**).
  - The imported data must be mapped to the headline of a data area. Ensure that the columns “Command”, “Object Identification”, “Source Object” are in this order.
  - After the properties of the external data source have been mapped to the properties of aspects save the data (menu **Bulk Data Manager > Save**).

Creating Formatted Templates

Formatted templates are Microsoft Excel based spreadsheet applications that allow the user to enter, manipulate, and document data in a user definable format including calculations, graphics and business charts. Data included in the template can be retrieved from or saved to different System 800xA applications. Examples of templates are budget planning, resource calculations, price calculations, or even advanced applications like TagSheets, SignalLists, LoopDiagrams, dimensioning of motors or valves. Some templates are described in more details in Appendix A, Engineering Templates.

Example Template

The following example shows a TagSheet. It will be used to explain how to build formatted templates and to demonstrate the various features available. Note, that the blue parts (columns C, P, W and rows 31, 32) of the TagSheet are normally hidden.

See also Appendix A, Engineering Templates for more information about the TagSheet template

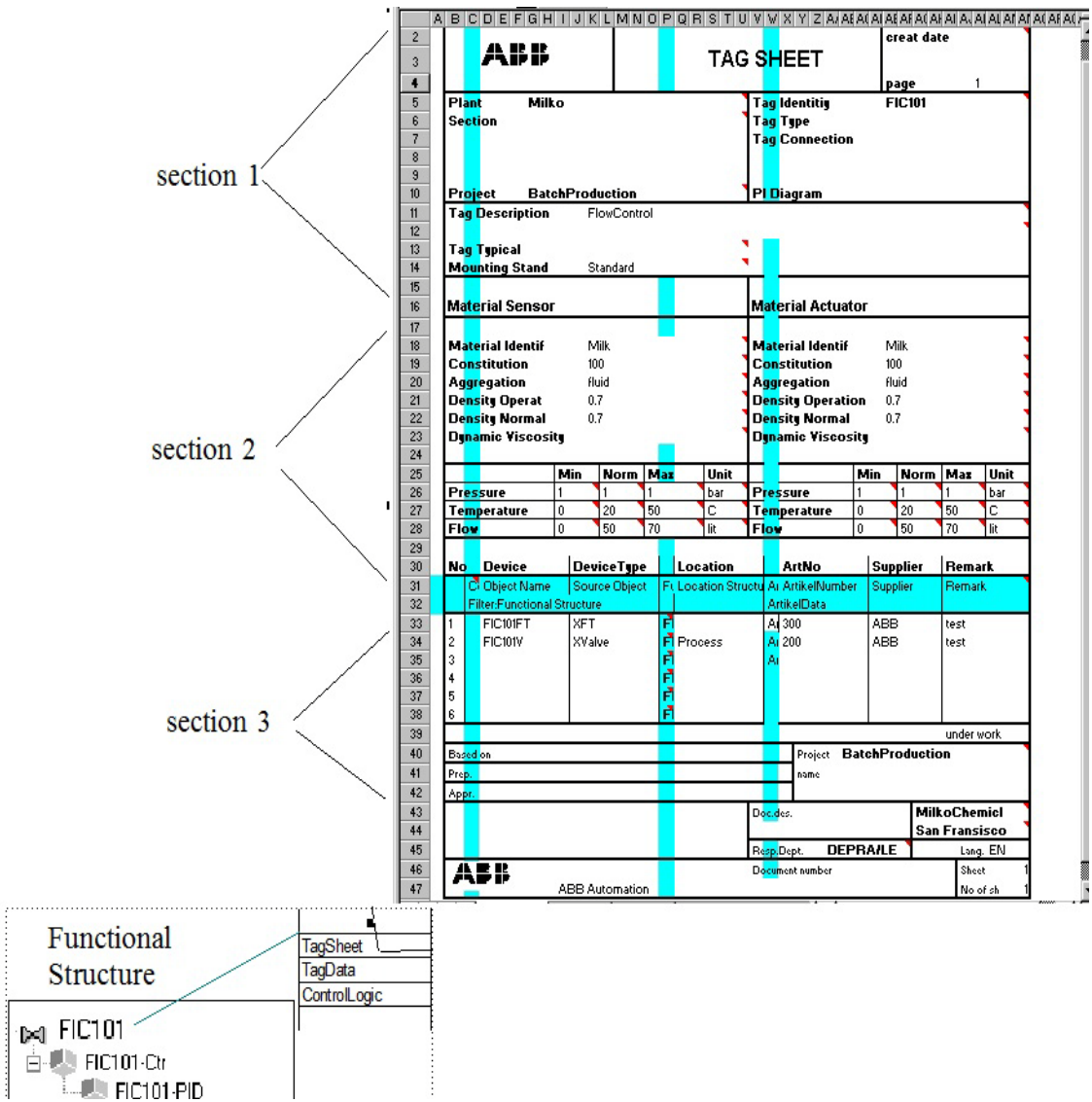


Figure 46. TagSheet

Explanations:

- The TagSheet is a Document Aspect of the object “FIC101” which represents a Flow Indication and Control loop. The object “FIC101” has among others two instruments as child objects:
  - FIC101FT: this object represents the flow transmitter
  - FIC101V: this object represents the valve
- The TagSheet is divided into different areas as described below:

  Section 1
  This area shows general data of the loop “FIC101”. For example, the TagIdentity (FIC101) is linked to the name of the object by using a so called Property Reference. Property References allow to insert references to single properties of aspects of the same or other objects. Property References are described in more detail in Inserting Property References. Whenever the name of the object is changed (e.g. in the Plant Explorer) the related cell in Microsoft Excel will be automatically updated when the Workbook is opened or printed. It is also possible to change the name of the object directly in Microsoft Excel by modifying the related cell content. When the cell was edited, the value will be written back to the referred property of an aspect (in this case the property “Name” of the aspect “Name”). Other Property References included (see comment indicators in the TagSheet) retrieve for example the type of the loop, the project name, etc.

  Section 2
  The section “Material Sensor / Actuator” provides information about the material characteristics where, for example, the sensor (flow transmitter) is inserted. Again the data items depicted, like Material Identification “Milk”, are retrieved via Property References (see also comment indicators in Figure 46 above). The material characteristics is stored in the TagData aspect of the object “FIC101”.

  Section 3
  The instrument list is a dynamic list of instruments used to implement the loop. The listed instruments represent the objects “FIC101FT” and “FIC101V”. For each instrument various data properties are presented like Device type, supplier, etc. The instrument list is implemented as an Auto-update Data Area which means the instruments and their data will automatically update whenever the TagSheet is opened or printed. For example, if a new instrument is added

below the loop object “FIC101”, it will show up automatically in the instrument section of the TagSheet. On the other side, it is also possible to add a new instrument in the instrument section of the TagSheet by simply entering a new line. When the instrument data area is saved a new instrument object will be created below the loop object “FIC101”.

The complete TagSheet is implemented in Microsoft Excel using the powerful formatting capabilities of Microsoft Excel like fonts, size, colors, merging of cells, cell framing, etc. The grid lines are switched off for the complete worksheet.

The following chapters explain in detail

- how to insert Property References and what type of references exist, and
- how to insert the instrument list (Auto-update Data Area).

Inserting Property References

This section gives an overview about Property References and how they are used in the Bulk Data Manager. For a detailed description of Property References refer to Appendix B, Property References.

A Property Reference is a dynamic link to a property of an aspect of any object. A Property Reference is automatically updated when the workbook is opened. On the other side, when a cell containing a Property Reference is changed in Microsoft Excel, the modification will be written back to the source where the Property Reference points to.

If you use this function to save data to System 800xA applications in an operating environment make sure that this does not conflict with secure plant operation.

A Property References can show strange results if Excel interprets the retrieved value as formula.

A Property Reference like “.:Functional Structure:ABS” shall result in an absolute reference designation, for example “=P1.A1”. Excel interprets such a value as a formula when the cell format is set to General, resulting in a strange value, ‘#NAME?, or even in an error message.

Format the cell with the Property Reference as Text and update the worksheet (Bulk Data Manager > Update All).

If you insert a Property Reference to an existing object and aspect but to an non-existing property, for example ".:Name.Name1", then the message #ERRROR is written. (Instead of expected #UNRESOLVED.) The normal reason for #ERROR is that the property could be found, but there was an error reading the value (e.g. because the OPC server is down).

If you insert a Property Reference to an object using an ambiguous object path, for example if there are two objects with the same name on the same hierarchy level, then the property is read of an arbitrary object matching the ambiguous object path instead of writing the message #ERROR into the cell.

Types of Property References

There are different types of Property References:

- Absolute References
  Absolute References are independent of the context or the object to which the workbook was attached as Document Aspect. They will point to the same object when the object the Workbook is related to is copied.
- Relative References
  Relative References point to properties of aspects of objects relative to the Start Object, i.e. they access the Start Object or child/parent objects of the Start Object. The latter is based on hierarchical structures. For example, within the TagSheet presented above it is possible to refer to data of objects that are either children or parents of the actual object (FIC101) in the Functional Structure. Child or parent objects can be identified based on a Relative Name which must be unique in a certain scope (substructure).
  Using Relative References has the advantage of making a template independent of an object. This allows to add a TagSheet to the FIC object type.

Relative references need a context in form of a Start Object. See Setting System and Start Object for more information about how to set the Start Object.

Property References with Subscription

Property References can be tagged to subscribe for live data. This means the value retrieved by the Property Reference updates itself automatically whenever it changes at the source. Subscription allows, for example, to create animated bar graphs.

How to Insert a Property Reference

In order to insert a Property Reference, follow the steps described below.

- Select a cell to insert the Property Reference.
- Open the context menu (right muse button) and perform menu entry **Property Reference > Insert**.
  This opens the dialog shown in Figure 47.

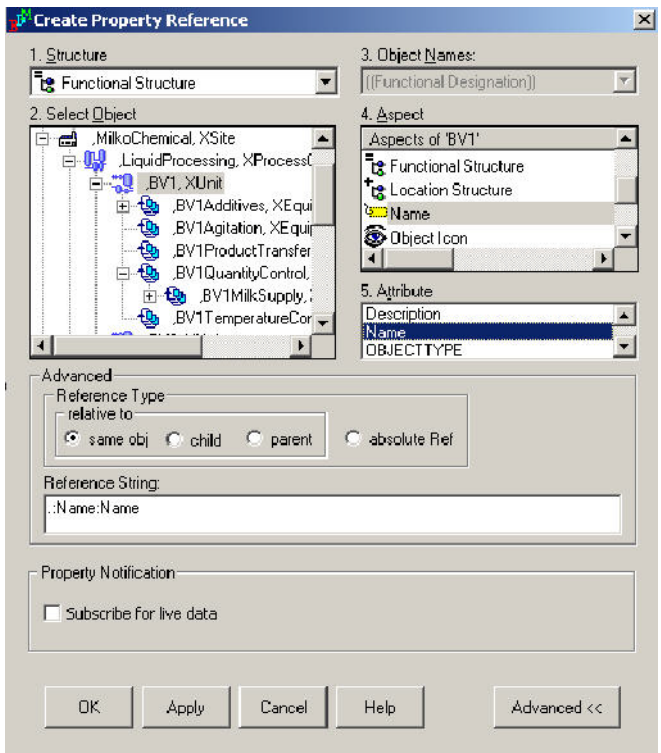


Figure 47. Insert Property Reference

Property References are stored by attaching special comments to Excel cells. Do not delete or modify these comments.

Explanations:

- When the dialog is opened the Start Object is highlighted. By default the functional structure is displayed.
- Insert **Property Reference** to the **same object**:
  - Select the aspect to refer to (Aspect area).
  - Select the attribute to refer to (Attribute area).
  - Press **OK** or **Apply** to insert the Property Reference.
- Insert **Property Reference** to a **child object**:
  - Select the structure where the child object to refer to is located.
  - Select the object to refer to by browsing through the substructure (Select Object area).
  - Select a method to identify the object. In many cases a relative name is used but it is also possible to identify the child object by other means like a relative reference designation.
  - Select the aspect to refer to (Aspect area).
  - Select the attribute to refer to (Attribute area).
  - Press **OK** or **Apply** to insert the Property Reference.
- Subscribe for live data
  If the value retrieved by the Property Reference should update itself automatically whenever the source changes select the option “Subscribe for live data”.
- It is also possible to edit the Property Reference string directly in the field “Reference String”.

Other Menu Entries Related to Property References

The following menu entries are accessible via the context menu (right mouse button).

- **Property Reference > Delete**
  This command deletes a Property Reference from a single cell or a set of selected cells.
- **Update All**
  Updates all Property References of a sheet. The difference to menu item “Bulk Data Manager -> Update All” is that only Property References will be updated and no Auto-update Data Areas.
- **Update Selection**
  Updates those Property References selected by the user. The difference to menu item “Bulk Data Manager -> Update Selection” is that only Property References will be updated and no Auto-update Data Areas.
- **Property Reference > Show / Hide Reference**
  Property References are stored in a comment related to a cell. These comments can be displayed in the sheet or hidden.
- **Property Reference > Show / Hide Indicator**
  There is a comment indicator shown in a cell if there is a comment added to the cell. These indicators can be shown or hidden.
- **Property Reference > Auto Insert**
  This menu entry is described in Monitoring of Live (Process) Data.

Inserting an Auto-Update Data Area

In the TagSheet above the instrument list is implemented as an Auto-update Data Area. How to create an Auto-update Data Area is described in Configuring an Auto-update Data Area. The following part of the TagSheet shows the Auto-update Data Area in more detail.

	No	Device	DeviceType	Location	ArtNo	Supplier	Remark
29							
30							
31		Object Name	Source Object	Location Structu	ArtikelNumber	Supplier	Remark
32		Filter:Functional Structure			ArtikelData		
33	1	FIC101FT	XFT		300	ABB	test
34	2	FIC101V	XValve	Process	200	ABB	test
35	3						
36	4						
37	5						
38	6						

Figure 48. Cutting of the TagSheet: Instrument List

As highlighted in Figure 48 above, there are hidden rows and columns included.

- The headline (row 31 in Figure 48 above) is build as described for data areas. However the complete headline is hidden in the final TagSheet. Instead an additional row with different column identifiers is added (row 30 in Figure 48 above). This row does not belong to the data area.
- The filter row is also hidden in the final TagSheet.
  The filter criteria select:
  - child objects of the actual object (FIC101) in the Functional Structure (Column “Command” = Filter: Functional Structure)
  - objects having an aspect called “ArticleData” (Column “ArticleData.AspectName” = ArticleData)
    Only objects having an aspect of category ArticleData called ArticleData are considered. The column “ArticleData.AspectName” is also hidden as it is not needed in the final TagSheet.
- The column “Functional Structure.ParentObject” is also hidden. The values in this column are retrieved via Property References. So the actual object “FIC101” is always inserted as the parent object. This controls that new instruments added in the instrument section of the TagSheet are automatically placed below the actual object “FIC101” in the Functional Structure when the data is saved (menu **Bulk Data Manager > Save**).

Remark: As the TagSheet uses relative references, the TagSheet can be attached as Document Aspect to any object and its contents will automatically fit to the context of the new object. The TagSheet could also be attached as Document Aspect to an Object Type.

Document Aspects managed by the Document Manager can be opened simultaneously by multiple users. However, multi user restrictions of Excel apply. This means, if a TagSheet is opened more than once at the same time, access is restricted to read-only. While it is not possible to save the TagSheet as Excel document, it is possible to save data to System 800xA applications using **Bulk Data Manager > Save**.

E.g., multiple simultaneous access to a TagSheet can occur easily if the Tagsheet is attached to an Object Type and marked as to be inherited, because the TagSheet will exist only once for all objects of that type. Alternatively, the TagSheet could be marked as to be copied when attached to an Object Type

Protect User Interface

Once a formatted template has been created and tested it should be protected against modifications like reshaping, changing frame borders, etc. Protecting the template also ensures that all build-in functionality will not be accidentally destroyed.

To protect a template perform the menu **Tools > Protection > Protect Sheet UI**. Optionally, a password can be provided in order to disable other users to un-protect the sheet.
By default, all options are marked but this can be changed (except “User Interface Only” and Contents).

Keep the option “User Interface only”.

Monitoring of Live (Process) Data

Subscribing for live data means that the values referred to will automatically be updated in an opened workbook whenever it changes. The subscription mechanism

is based on Property References that can be tagged to subscribe for live data. The following example shows a process monitoring scenario.

Live data property reference with very high update rate slows down Excel (as well as Word) applications.

The property references are still working. Reduce the update rate of the referenced aspect property if possible or reduce the amount of such live data property references. Or terminate the application using the Windows Task Manager.

Monitoring of Process Values

Figure 49 depicts an example of a radar diagram. It shows in an ordinary Microsoft Excel chart four important process values of the Batch Vessel 1 and 2. Different chart presentations out of the rich set offered by Microsoft Excel can be selected.

The process values are also displayed below the related chart. The values are subject to conditional formatting rules configured in Microsoft Excel (see menu “**Format >**

Conditional Formatting" in Microsoft Excel). In this case the color of the values changes when certain limits are passed.

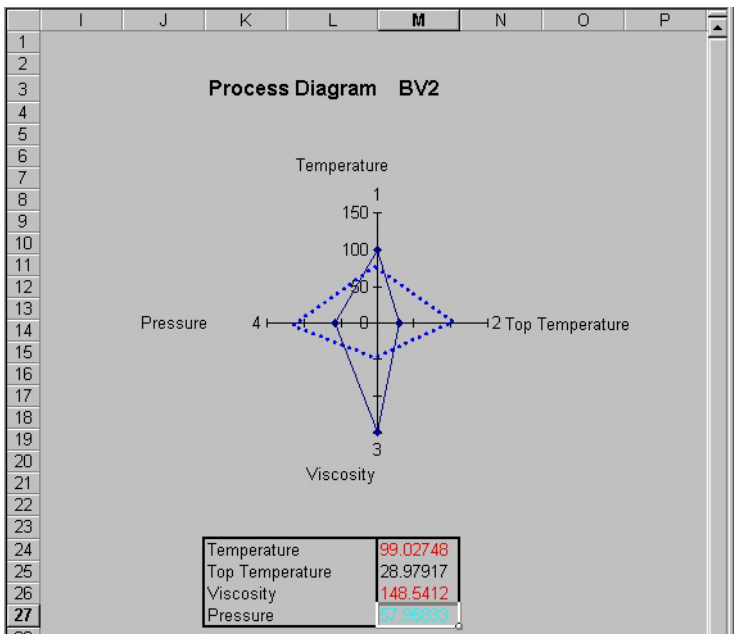


Figure 49. Monitoring of Live Data: Radar Diagram

Another example could be the supervision of the status of certain devices. In this case a subscription would be established to the status property of the Control Connection aspect (in case of AC400 series of controllers).

The monitoring function can also be used in combination with the powerful arithmetic, logical, textual and other functions offered by Microsoft Excel to perform calculations based on process data.

Not all properties of all aspect categories can be subscribed for. This depends on the aspect category.

Create a Subscription for Live Data

There are basically two ways:

- Insert an individual subscription for live data:
  To insert a single property with subscription for live data follow the steps below:
  - Select a cell.
  - Insert a Property Reference (e.g. to the property “VALUE” of the Control Connection aspect (in case of AC 400 series of controllers).
    Ensure that the check box “Subscribe for live data” is activated.
- Insert a set of subscriptions for live data:
  To insert a set of subscriptions for live data follow the steps below:
  - Create a Default Data Area by dropping e.g. the Control Connection aspect into an empty Worksheet (see also Configuring the Default Data Area).
  - Delete the properties that are of no interest.
  - Drop the objects to be monitored (e.g. Analog Input Signals) into the Worksheet.
  - Select e.g. the cells in the column “VALUE” for all objects to be monitored.
  - Open the context menu (right mouse) and perform the menu entry “**Property Reference > AutoInsert**”.

This will automatically create a Property Reference with subscription for live data for all selected cells.

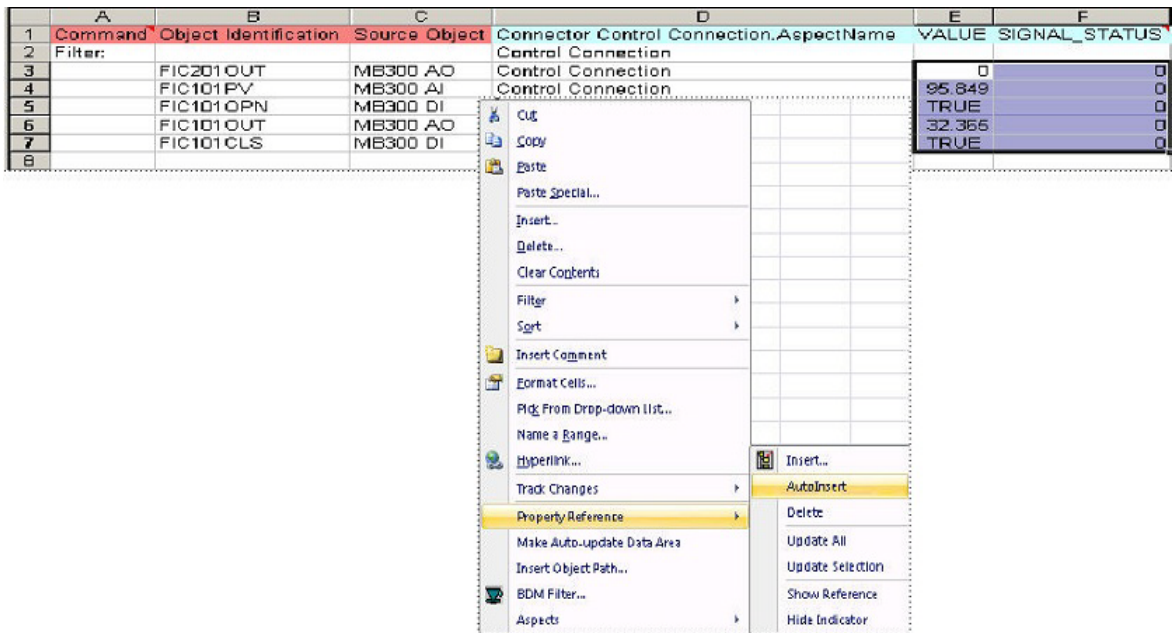


Figure 50. Monitoring of several Process Values and Status Information

If objects are deleted be sure that the Property References of these objects are also deleted (manual action).

If new objects are dropped into the worksheet the Property References need to be added to these objects as described above (manual action). See Inserting Property References.

Using Functions to Read/Write Data

Microsoft Excel offers a rich set of functions to perform arithmetic, statistical, logical and other calculations. With the Bulk Data Manager it is possible to include data from other System 800xA applications into the Microsoft Excel calculations

and to write calculation results back to other System 800xA applications. There are different ways to include data from other applications (and even process data):

- Using Property References to retrieve data.
  For more information on how to create Property References please refer to Inserting Property References).
- Using data areas.
  For more information on how to create data areas please refer to the Configuring the Default Data Area and Configuring an Auto-update Data Area).
- Using Read and Write functions as described below.

There are two functions offered:

- ***ReadRef (PropertyReference as String) as String***
  This function takes a Property Reference as input and returns the related value. If an error occurs, the function returns "#Error".

  Example: Cell "A1" contains "=ReadRef(".:Name:Name")

  This will include the name of the Start Object into cell "A1". For more information about the Start Object see Setting System and Start Object.

  Remarks:
  The Property Reference string ".:Name:Name" can be built using the dialog described in Inserting Property References. For example place the cursor in cell "B1", insert a Property Reference using the dialog. The resulting Property Reference string is stored in a comment of the cell "B1"and can be copied from there. Delete the Property Reference from cell "B1" afterwards.

  The Property Reference can also be a cell containing the Property Reference string.

  Example: Cell "F1" contains the Property Reference string ".:Name:Name"
  Cell "G1" contains "=ReadRef(F1)

- ***WriteRef (PropertyReference as string; Value as Variant) as Boolean***
  This function writes the value, provided as an input parameter, to the property identified by the Property Reference which is the 2nd input parameter.

Example: Cell “A1” contains “=WriteRef(”.:Name:Name”, “MyObject”)

This will write the string “MyObject” into the name property of the Start Object and hence rename the object. For more information about the Start Object see Setting System and Start Object.

Remark: The input parameters can also be references to cells that contain the value or the Property Reference respectively.

Example:Cell “A1” contains “=WriteRef(B1, C1)
Cell “B1” contains the Property Reference string: “.:Name:Name”
Cell “C1” contains the value to write: “MyObject”

Microsoft Excel recalculates a cell whenever the cell or a dependent cell changes.

This means that a cell containing the “ReadRef” function is only updated when the Property Reference changes but not if the referenced property changes, i.e. ReadRef does not support subscription. Use plain Property References with subscription instead.

In the “WriteRef” example above, the value stored in cell B1 will be written to the property pointed to by the Property Reference stored in C1 whenever the cell B1 or C1 changes its contents. You can’t use plain Property References to achieve the same behavior since Property References are written to the Aspect Server only when the value is changed manually.

So, if you want to use Excel for reading controller values, performing some calculation in Excel, and writing the calculated value back to the controller, you should use a Property Reference with subscription for reading, and the “WriteRef” function for writing.

The Read / Write Data Functions can be used in VBA macros as well

If you use this function to save data to System 800xA applications in an operating environment make sure that this does not conflict with secure plant operation

Cross-Navigation to System 800xA Applications

This feature is only available if Excel is started from a Document Aspect or from the “Advanced” menu of the Aspect Object context menu.

Once an object is identified it is possible to directly navigate to other aspects of the same object or even to structures the object is included in. An object can be identified in a Worksheet in different ways:

- within a data area
  a. Select a cell containing the identification of an object (column “Object Identification”).
  b. Perform menu entry “Aspects” in the context menu (right mouse). A selection list showing all aspects of the selected object will pop up.
  c. Select an aspect to open the related application in a separate window.

  Remark: If you select a cell within the column “Source Object” the aspects of the Source Object will be shown. The Source Object can either be an object instance or an object type.
- based on Property References
  a. Select a cell containing a Property Reference to the name of an object.
  b. Perform menu entry “Aspect List” in the context menu (right mouse).

In order to navigate to the structure, the object is located in, select the related structure aspect (e.g. Functional Structure aspect). This will open the Plant Explorer displaying the related structure and highlighting the object within the structure. From there it is possible to continue to navigate to related objects or to other structures.

If the selected cell contains no valid object identification or the object identification is ambiguous, no aspects are shown.

If the Bulk Data Manager was not started from a Document Aspect or from the “Advanced” menu of the Aspect Object context menu, the aspects are listed but they are disabled because no cross navigation is possible.

Adding Constraints to a Worksheet

In some cases there is a need to support the user in filling in the values of cells and to perform consistency checks like:

- show a list of allowed values
  Example: The values allowed for the property “Unit” of an Analog Input Signal are “%, mA, kPa, ...”
- check the values of one column in relation to another column
  Example: HighLimit2 >= HighLimit1 >= LowLimit1 >= LowLimit2
- check that the number of characters of certain cells are within the limits
  Example: the name of signal objects must not be longer than 20 characters
- etc.

Constraints can be entered to individual cells (typically in formatted templates) or complete columns (typically in data areas). In order to create a constraint, follow the steps described below:

Enter a Constraint for a Single Cell

- Select the cell.

- Perform menu entry "**Data > Validation**" in Microsoft Excel. This results in the following dialog.

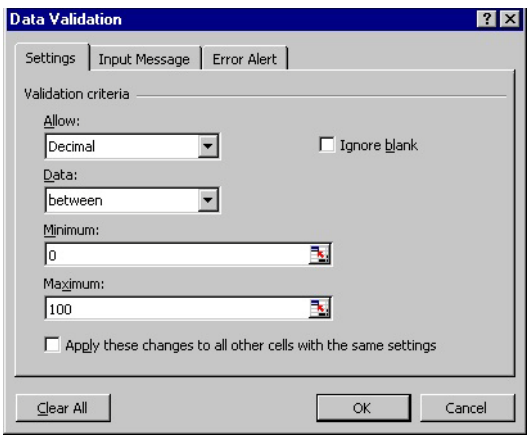


Figure 51. Microsoft Excel Data Validation Dialog

In this example the values for the selected cell must be between "0"and "100".

The constraints are by default enforced. This means no other values are allowed. If you want the constraints to be of an advisory character, giving the user just a hint in case of constraint violation, set the style "Warning" in the "Error Alert" tab.

Please refer to the Microsoft Excel documentation or on-line help to get more information about the possible constraints that can be defined. In the Appendix A, Engineering Templates there is an example included (Signal List) that has several constraints defined.

Enter a Constraint for a Complete Column

- Select the first cell of the column.
- Perform menu entry "**Data > Validation**" in Microsoft Excel displaying the same dialog as described above. Define the constraints according to the required business rules. If you need to refer to other columns, refer to the first cell of the related column.

- If there are cells that have to be excluded from the constraint check (like the headline cell), select these cells, open the same dialog again and set the field “Allow” to “Any value”.

List of Values

There is a special feature for suggesting values provided by the Bulk Data Manager which is a dynamic pick list of values. Consider the following example of a Signal List:

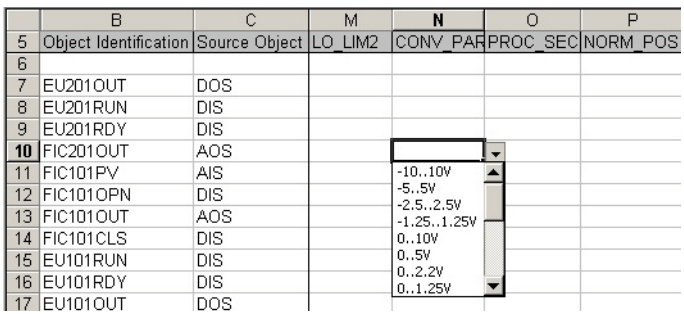


Figure 52. Selection Box: List of Values

The pick-list provided for a property depends on the property itself and on the object type specified in column Source Object.

So when a cell of the column “CONV_PAR” is selected that refers to an Analog Output Signal the values proposed are “-10..10V,...”. However, when the selected cell refers to a Digital Output Signal the CONV_PAR property makes no sense. In this case the list of values should be empty.

This behavior can be defined in a so called “List Of Values” (LOV) workbook that is a separate Excel file. The following example shows a part of the list of values for the SignalList:

	A	B
1	AIPTS_SIGNALPARAMETER_CONV_PAR	40,20,12
2	AIPTS_SIGNALPARAMETER_CONV_PAR	40,20,13
3	AIPTS_SIGNALPARAMETER_CONV_PAR	40,30,12
4	AIPTS_SIGNALPARAMETER_CONV_PAR	40,30,13
5	AIPTS_SIGNALPARAMETER_CONV_PAR	40,50,12
6	AIPTS_SIGNALPARAMETER_CONV_PAR	40,50,13
7	AIPTS_SIGNALPARAMETER_CONV_PAR	40,60,12
8	AIPTS_SIGNALPARAMETER_CONV_PAR	40,60,13
9	AOS_SIGNALPARAMETER_CONV_PAR	-10..10V
10	AOS_SIGNALPARAMETER_CONV_PAR	-5..5V
11	AOS_SIGNALPARAMETER_CONV_PAR	-2.5..2.5V
12	AOS_SIGNALPARAMETER_CONV_PAR	-1.25..1.25V
13	AOS_SIGNALPARAMETER_CONV_PAR	0..10V
14	AOS_SIGNALPARAMETER_CONV_PAR	0..5V
15	AOS_SIGNALPARAMETER_CONV_PAR	0..2.2V
16	AOS_SIGNALPARAMETER_CONV_PAR	0..1.25V
17	AIS810_SIGNALPARAMETER_CONV_PAR	0..20mA
18	AIS810_SIGNALPARAMETER_CONV_PAR	4..20mA
19	AIS810_SIGNALPARAMETER_CONV_PAR	0..10V
20	AIS810_SIGNALPARAMETER_CONV_PAR	2..10V
21	BDM_SOURCE_OBJECT	AIPTS
22	BDM_SOURCE_OBJECT	AOS
23	BDM_SOURCE_OBJECT	AIS810

Figure 53. Definition of List of Values

Figure 53 defines 3 pick lists (List of Values). Column “A” holds the key to identify the lists and column “B” holds the list of values itself. Row 1 to 8 defines the first list, row 9 to 16 the second etc. The key of column “A” is a combination of an aspect property and the related object type (Source Object). For example the key called “AIPTS_SIGNALPARAMETER_CONV_PAR” identifies the list of values for the Property “CONV_PAR” of a SignalParameter aspect of an object of type “AIPTS”. Similar, the list of values for an Analog Output Signal is defined in row 9 to 16 etc.

The keys must be named according to the following schema:

SourceObject_AspectCategory_Property

SourceObject, AspectCategory and Property have to be written in uppercase, concatenated with sign “_” and in addition all space characters in SourceObject, AspectCategory and Property have to be changed to sign “_”.

A special case of a list of values is shown in row 21 to 23. The used key “BDM_SOURCE_OBJECT” defines a pick list for the object types (column “Source Object”) with the values “AIPTS”, “AOS” and “AIS810”.

The list of values can be maintained centrally in a separate Excel file and must be stored in the same directory where the related Excel Workbook is stored. For example the list of values for the file “SignalList.XLS” is called “SignalList_LOV.XLS” and is stored in the same directory where “SignalList.XLS” is stored. In general, the list of values has to be named in the following way <FileName>_LOV.XLS (where <FileName> is the identifier of the related Excel file the list of values is defined for).
When the SignalList is opened the related LOV file is also opened and included into the SignalList. So when mailing the SignalList to other sites this feature still works even if there is no connection to the Aspect Server.

The following conditions must be fulfilled to get List of Values work:

- Sheet name of the list of values workbook “...._LOV.xls” must be "Data_LOV".
- In Source Column the object type name has no path, only the name value.
- For the first time the cell value must be changed. After any modification the "list of values" button will appear.

Auto-open Related Workbooks

In some cases it is required to open two or more Excel files. If you apply the following naming convention, one or more files will automatically be opened and the windows are arranged in “Tiled mode”.

Naming Convention:

first file is called:	<FileName.XLS>
second file is called:	<FileName_1.XLS>
third file is called:	<FileName_2.XLS>
etc.	

Now, when opening the first file the other files will automatically be opened as well. This allows to e.g. create a shortcut of the first file to the desktop to open all required files in one action.

Creating Macros

Many of the features of the Bulk Data Manager described in this manual can also be controlled by macros created in Microsoft Excel using Visual Basic for Applications. No special knowledge of the COM-interfaces of the System 800xA platform is required as all information can be retrieved and written by manipulating cells in Microsoft Excel.

Macros can be used for example to fill in data in a data area and save it afterwards. This allows the user for example to:

- create objects
- duplicate objects
- instantiate object types
- rename objects
- place objects in structures
- provide values for properties of aspects
- delete objects by setting the property “Command” (see also list of available commands in Saving Data to System 800xA Applications)

To save the data stored in data areas use the procedure “SaveObjects”.
Example: Application.Run(“LBEMacros!SaveObjects”)

To update a selected Auto-update Data Area use the procedure “Update”
Example: Application.Run(“LBEMacros!Update”)

To update all Auto-update Data Areas use the procedure “UpdateAll”
Example: Application.Run(“LBEMacros!UpdateAll”)

Macros can also be used to **react on changes of selected process values**. As a prerequisite the cell including the Property Reference to a certain property (e.g. process value) must have been inserted having “Subscribe for life data” activated (see Inserting Property References). Whenever the property referred to changes its value the cell in Microsoft Excel automatically updates its contents. To react on the changes of a cell add the required code (Visual Basic) into the “Worksheet_change()” event procedure of the actual Worksheet.

Macros can be recorded by performing all steps interactively and than adapt the code to fit the requirements.

The functions ReadRef and WriteRef described in Using Functions to Read/Write Data, can be used in macros as well.

Audit Trail Events

Bulk Data Manager creates Audit Trail Events for “save” operations if the Audit Trail feature is enabled within the current 800xA system:

- Before a Bulk Data Manager “save” operation has been started.
- After a Bulk Data Manager “save” operation has been finished.
- While a Bulk Data Manager “save” operation failed.

Authentication

The 800xA system supports Re-Authentication or Double Authentication, which can be configured for:

- Aspect Categories
- OPC properties

The related authentication dialog appears (if authentication is enabled) and the required authentication can be done.

Authentication dialogs will also appear during a “save operation” in Bulk Data Manager. You can disable the Authentication feature in the Admin Structure.

E-Signature Properties

Bulk Data Manager can read a number of E-Signature properties of an aspect after this aspect has been signed in the 800xA system.

First Signature Properties:

- SignTime1
- SignUser1

- SignFullName1
- SignReason1
- SignComment1
- Signature1
- SignStatus1

Second Signature Properties:

- SignTime2
- SignUser2
- SignFullName2
- SignReason2
- SignComment2
- Signature2
- SignStatus2

All these E-Signature properties are “read only” properties.

Executive Summary

The Bulk Data Manager in its current version supports simple properties in various ways as well as so called structured properties as described in this manual.

Definition of Structured Properties

Before we focus on structured properties an example for simple properties is presented in the figure below, in order to differentiate simple and structured properties.

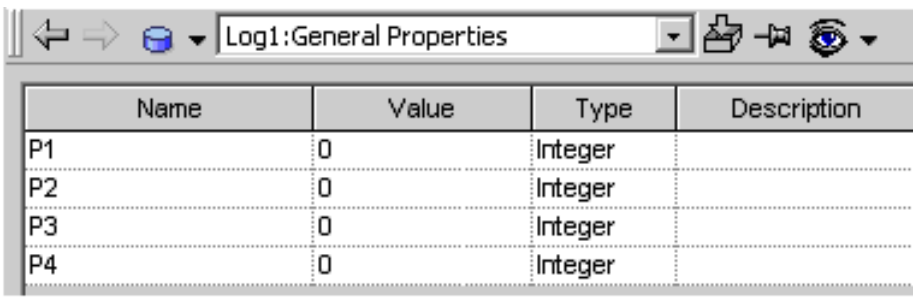


Figure 54. Simple Properties

The properties "P1" to "P4" are of simple data type like integer, string, etc. and do not have a substructure.

Structured properties - as opposed to simple properties - have a more or less deep substructure as described below using examples based on Log Configuration aspects.

The "Log Configuration" aspect cannot be described as a set of simple properties but rather as a set of structured properties as illustrated in the figure below:

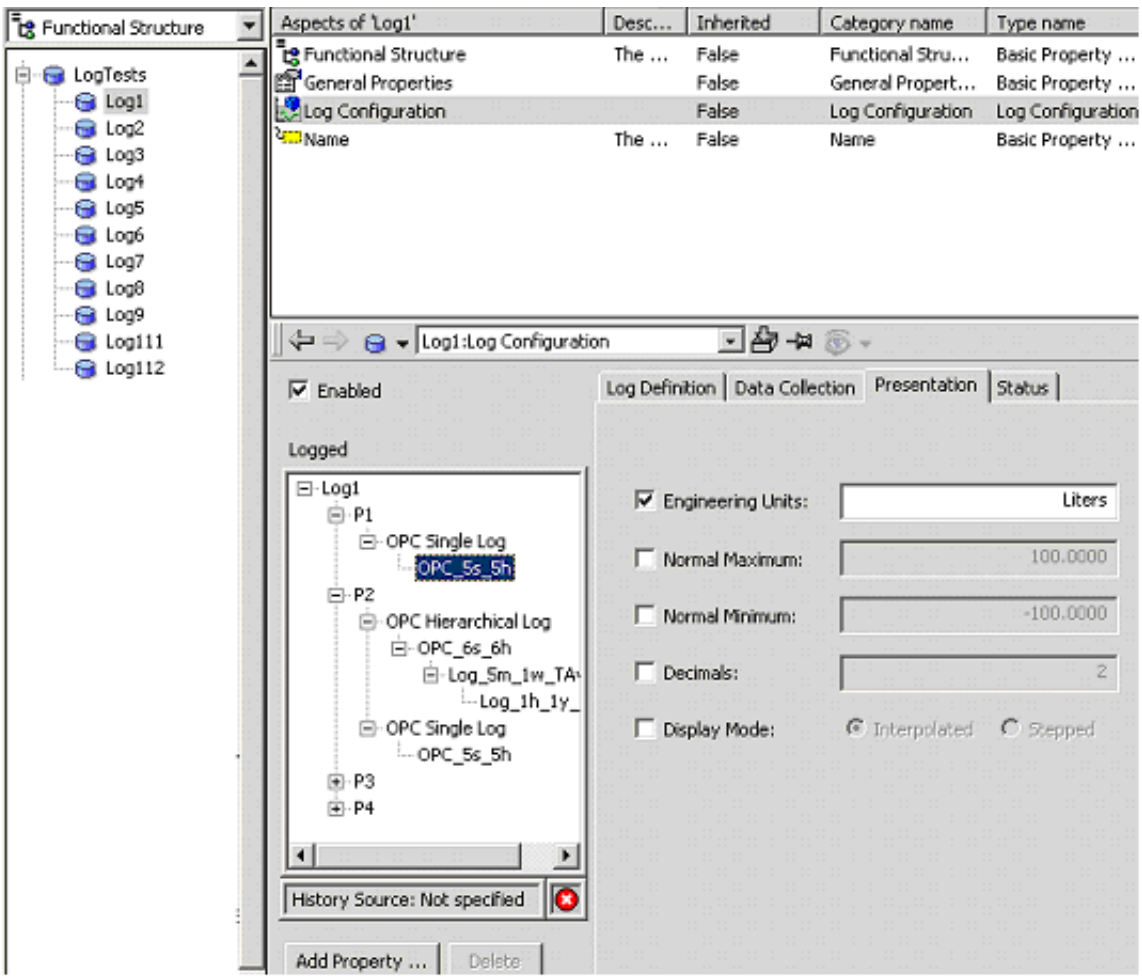


Figure 55. Example for Structured Properties

In the example above the Log Configuration aspect has one structured property "Log1", which has a substructure consisting of "P1","P2","P3", and "P4". This level

is called "PropertyLog" and represents the properties of the object to be logged. Each "PropertyLog" is sub-structured into "LogTemplates". A property, as for example "P2", can be logged based on different "LogTemplates". A "LogTemplate" again is sub-structured into "Logs" (e.g. "OPC_5s_5h"). A "Log" again is sub-structured and composed of the properties "Engineering Units", "NormalMaximum", etc. These properties finally represent the leaves in the hierarchy of structured properties. Hence they are not further sub-structured.

Aspect Categories Supporting Structured Properties

The following aspect categories support structured properties:

- Log Configuration
- Trend Configuration
- Control Builder M
  - Applications
  - Programs
  - Control Module Types
  - Control Modules
  - Single Control Modules
  - Function Block Types
  - Function Blocks
  - Hardware Units
- Parameter Manager Categories
  The Parameter Manager allows the user to define aspect categories either using simple data types like string or integer as well as structured properties. These aspect categories are automatically enabled to co-operate with the Bulk Data Manager

Most of these aspect categories are supported by pre-defined templates available in the folder “Engineering Templates” placed at the desktop.

Please refer to the description of the installed components to check if they support structured properties. As a prerequisite to work with structured properties in the

Bulk Data Manager an aspect system must support the so called Bulk Data Interfaces.

Presentation of Structured Properties in Bulk Data Manager

Mapping structures like a Log Configuration aspect to a grid like an Excel sheet has pros and cons. The advantage is on bulk operations on these data as described below while on the other side however the structure partly dissolves while mapped to a 2-dimensional Excel sheet.

The following picture illustrates how structured properties are mapped to MS Excel:

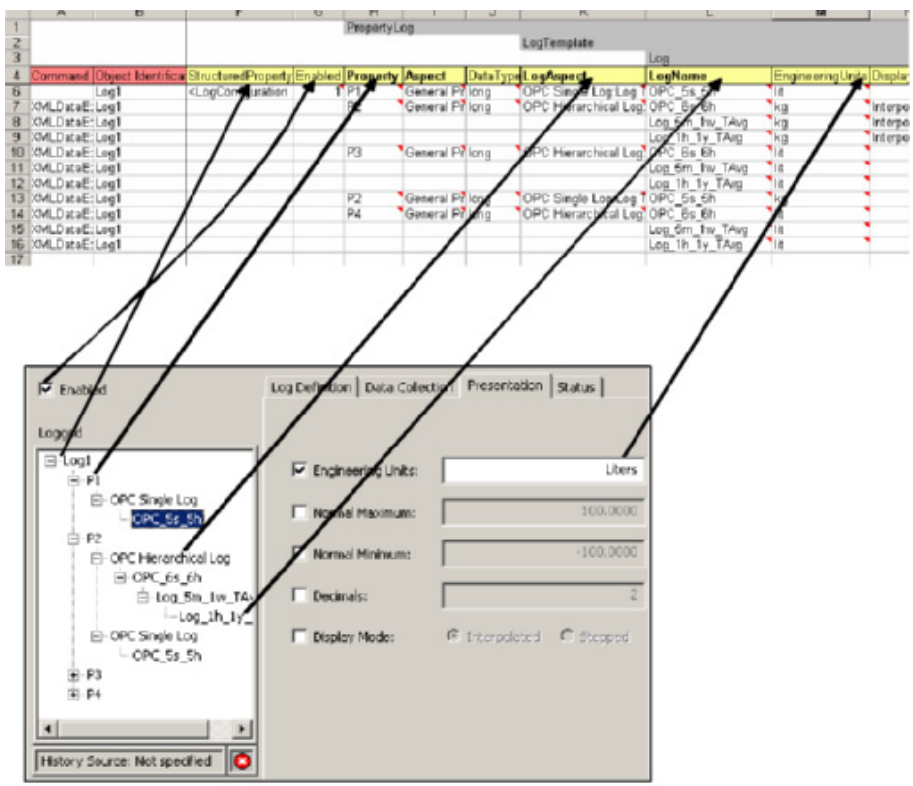


Figure 56. Mapping Structured Properties to MS Excel Sheet

Which structured properties or simple properties respectively are mapped to the Excel sheet depends on the configuration of the headline (see Configure Headline).

As an aid for the user the structure of the Log Configuration aspect is indicated at the top of the sheet.

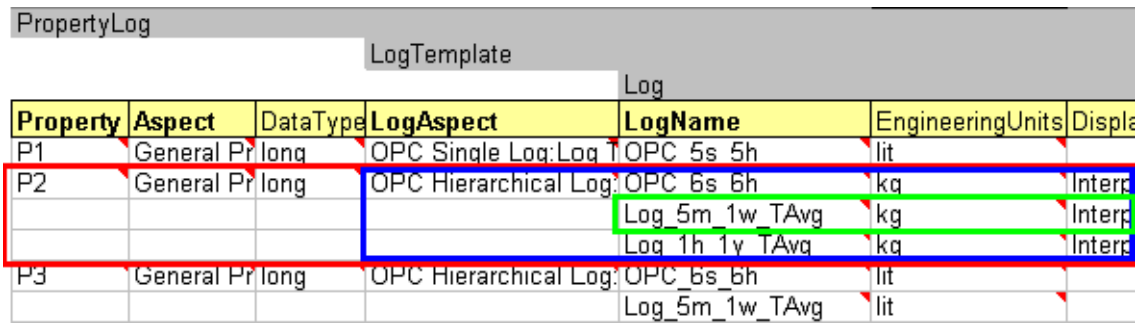


Figure 57. Indicating Structures in the MS Excel Sheet

- The read frame (outer frame) indicates a complete PropertyLog that is identified by its Property and Aspect column.
- The blue frame (middle frame) indicates a LogTemplate. In this case a hierarchical log template is used that has three logs related to it.
- The green frame (inner frame) indicates a Log with its related simple properties like Engineering Units, etc.

Reading and Writing Data

Once the headline has been configured (see Configure Headline), Aspect Objects with a Log Configuration aspect can be dropped into the Bulk Data Manager. The complete data of the aspect is stored in a cell of a column titled "XMLData". After data have been dropped the individual properties as configured in the headline will automatically be populated. This is indicated by a red comment indication in the upper right corner of the effected cells.

If you want to switch the comment indications off open the MS Excel options dialog (menu "Tools/Options"), select the tab "View" and select option "None" for the "Comments" section. Do not delete the comments!

After objects have been dropped into the sheet data can be changed and saved back to the System 800xA platform by e.g. pressing the button "Save" in the Bulk Data Manager toolbar.

Working with Structured Properties

The Bulk Data Manager has been enhanced by functions dedicated to structured properties. In addition most of the powerful MS Excel features can be used to efficiently work on structured properties. The following operations are offered:

- Modify operations
- Insert operations
- Copy operations
- Delete operations

Toolbar for Structured Properties

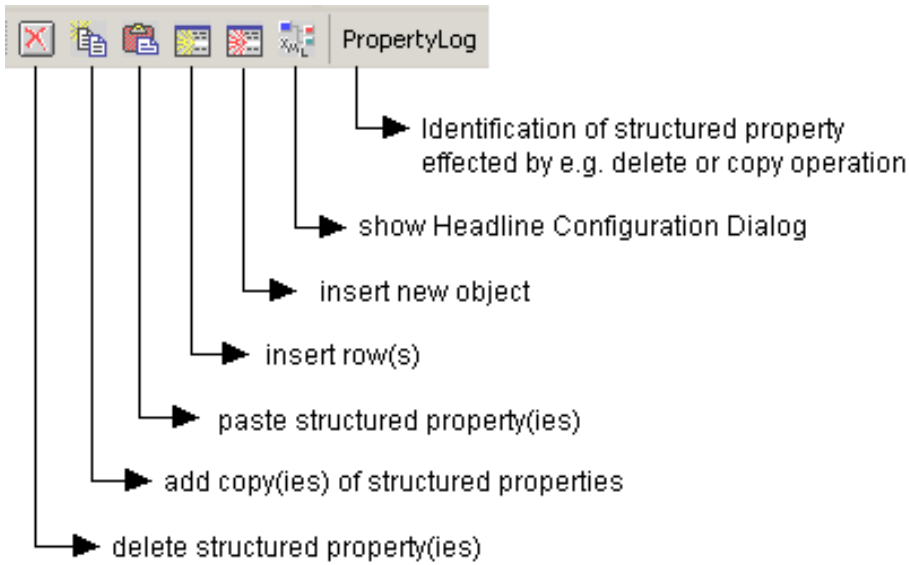


Figure 58. Toolbar for Structured Properties

Additionally a Clear Sheet button is available.

Context Menu for Structured Properties

The functions offered by the toolbar depicted above are also available via the context menu (right mouse click). The context menu offers in addition to the toolbar the following function:

- Select Substructure <structured property>
  this command highlights the cells on the MS Excel sheet that belong together as a part of a structured property. If, for example, the cell "K12" has been selected and the context menu "Bulk XML Editor/Select Substructure

LogTemplate" has been performed the related area is highlighted in the sheet as illustrated in the figure below.

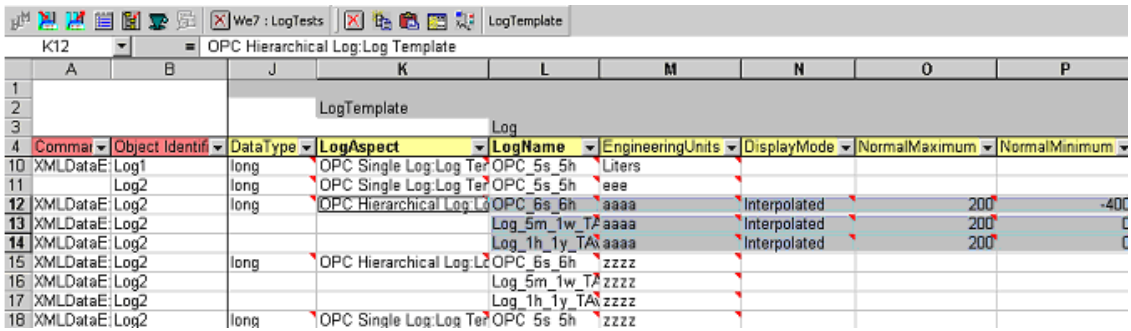


Figure 59. Highlight Substructure

A possible use case is described in Delete Operations.

Modify Operations

In order to modify the value of an existing property select the cell and change its value. All Excel functions, like “replace a range of cell” or “pulling down a cell” can be used.

Insert Operations

Insert operations can be classified as

- inserting new structured properties on any level
- inserting new objects/aspects with structured properties

Inserting new Structured Properties

Assuming we want to insert a new "LogTemplate" for the existing property "P4" of object "Log2" (see Figure 60 below). In this case use MS Excel to insert a new empty line below row “15” and type in the required information for the LogTemplate. As the data entered in the new row “16” is placed below the PropertyLog of property “P4” it will become a sub-property of “P4” when data is saved back to the System 800xA platform. During the save operation the missing

entry for the column “Object Identification” and “Command” are automatically entered.

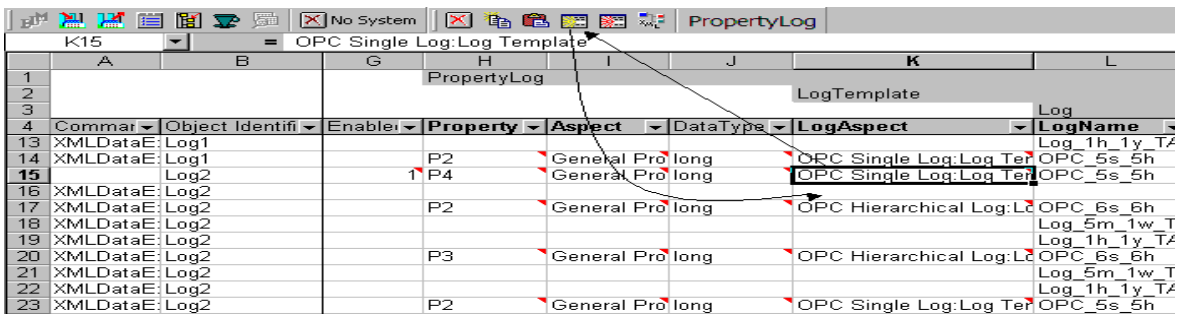


Figure 60. Insert LogTemplate

Analog a new “PropertyLog”, “Log”, etc. can be added. For example if property “P5” had been entered in cell “H16” the data of row “16” would have been related to a new PropertyLog for property “P5” (this scenario is not depicted in Figure 60).

If several structured properties (e.g. several “Logs” for a “Hierarchical Log Template” have to be entered, several rows have to be inserted at the desired location.

Alternatively you can use the button “Insert...” of the structured property toolbar in order to create empty rows. In this case the columns “Object Identification” and “Command” are filled in immediately.
The "Insert" function is an operation that can be applied to ranges (selection of several cells). For example if the range "K14 - K17" has been selected and the button "Insert" is pressed an empty row is inserted at the end of each selected "LogTemplate". If you select however the range "L14 - L17" an empty row would be inserted after each individual "Log". So the insert function (as other operations) is sensitive to the cells/columns selected.

If structured properties have to be entered at the end of a data area they simply can be filled in into the related cells.

Insert New Objects with Structured Properties

In order to insert a new object with a LogConfiguration aspect enter a new Object Identification in column “Object Identification” and continue to fill in the data for

the LogConfiguration aspect. If you need to enter data in several rows (as usual for structured properties) you do not have do repeat the object identification. This will be done automatically when data is saved back to the System 800xA platform.

Alternatively, you can use the button “Insert New Object” in the structured properties toolbar. This adds a new Object with a LogConfiguration aspect at the end of the data area. Just replace the automatically created object name “New Object” with the required name.

Copy Operations

- **Copy/Paste Cell**
  A single cell or a range of cells can be copied and pasted to other cells using the MS Excel menus "Cut", "Copy" and "Paste" as used to when working with MS Excel.
- **Pull Down Cells**
  Place the cursor in the cell the value of should be applied to the cells below. Put the cursor to the right lower corner of the cell (the cursor changes its shape to a cross). Left click the mouse and pull the cell down. This way the contents of a cell can be copied to other cells even crossing object borders.

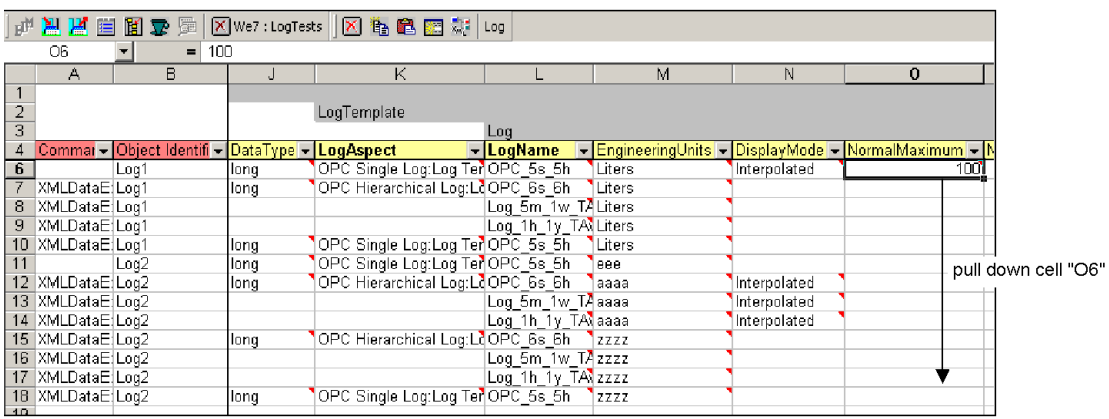


Figure 61. Pull Down Cells

- **Logical Copy of Structured Property**
  As an example we consider property "P2" of object "Log2". This property has a single log template defined. If we want, for the sake of explanation, to add a hierarchical log template definition we can copy the related log template from e.g. property "P3" of the same object. This is done in the following steps:
  - Select e.g. cell "K15" and perform MS Excel Menu "Copy".
  - Select cell "K19" and press the button "Paste" in the structured property toolbar.

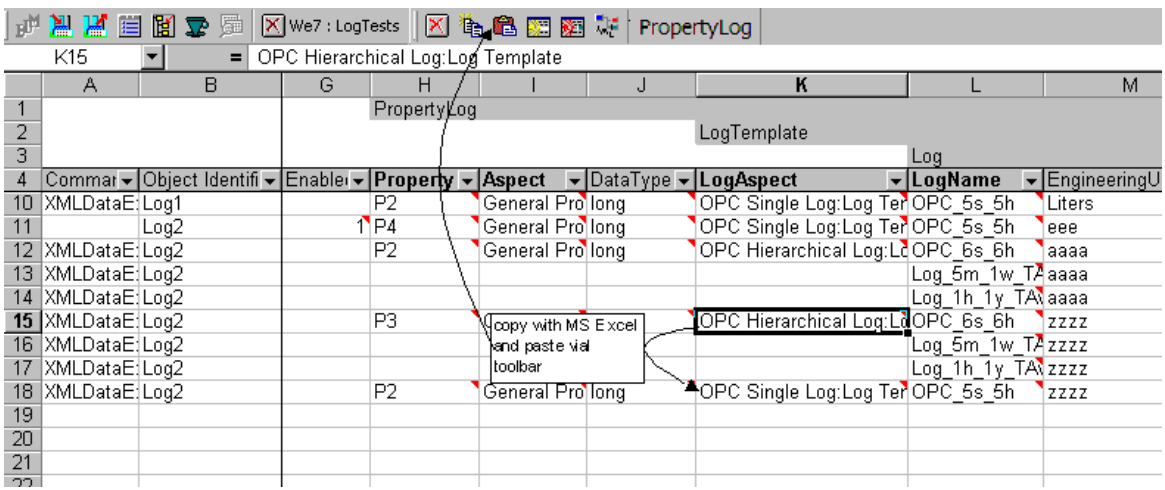


Figure 62. Copy a Structured Property

  - This results in copying the complete log template as specified for property "P3" to property "P2" (see Figure 63).

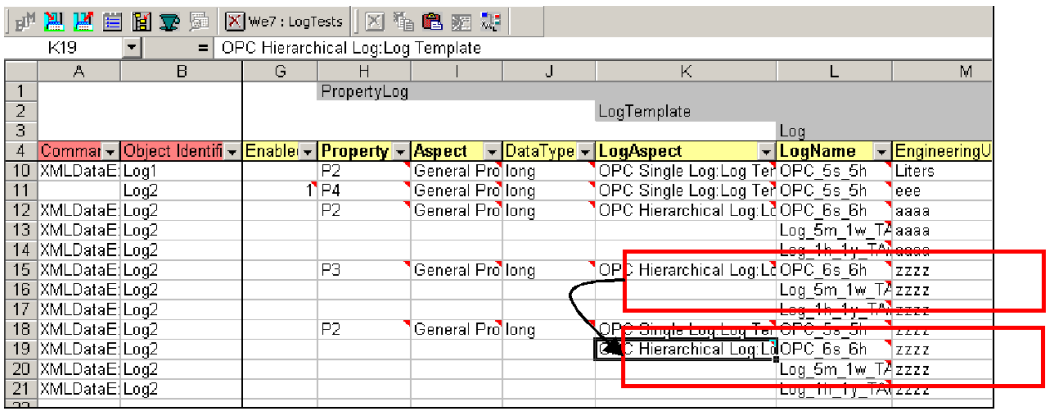


Figure 63. Copy of Structured Property

If, for example, the hierarchical log template of property "P3" is to be copied to property "P2" of object "Log1" we first need to create an empty line below property "P2" (see “Insert Operations” before) before the hierarchical log template can be copied.

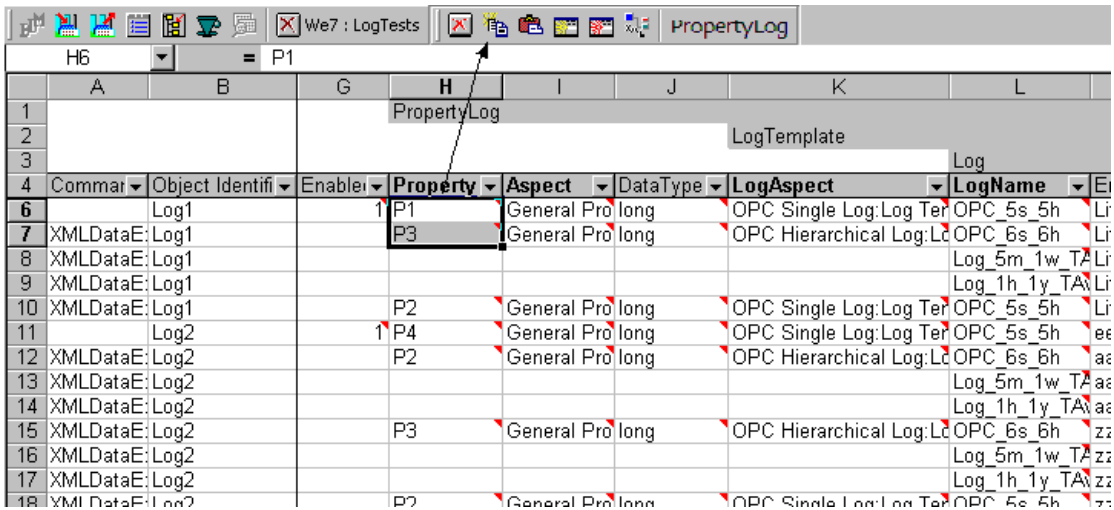
The copy/paste function is an operation that can be applied to ranges (selection of several cells). For example if the cells "K11 - K12" are selected the single log template of property "P4" and the hierarchical log template of property "P2" are copied and can be pasted afterwards.

If a workbook with Structured Properties e.g LogConfig.xls is open and the Structured Properties command “Paste <Parent Node> has been performed at a row representing a new set of Structured Properties for another Aspect Object (column “ObjectIdentification” has an entry different than the row above): These Aspect Objects are not saved during a Bulk Data Manager “save” operation. Make sure that the first row representing another Aspect Object has no entry in the column “Command”. If this cell contains “XMLDataExpanded” then clear the cell.

AddCopy of Structured Property

Similar to copy/paste of structured properties there is also a possibility to add copies of structured properties right after the selected property. For example select the

properties "P1" and "P3" (cells “H6:H7” in Figure 64) of Object "Log1"and press the button "AddCopy" in the structured properties toolbar.



	A	B	G	H	I	J	K	L
4	Commar	Object Identifi	Enable	Property	Aspect	DataType	LogAspect	LogName
6		Log1	1	P1	General Pro	long	OPC Single Log:Log Ter	OPC_5s_5h
7	XMLDataE:	Log1		P3	General Pro	long	OPC Hierarchical Log:Lo	OPC_6s_6h
8	XMLDataE:	Log1						Log_5m_1w_TA
9	XMLDataE:	Log1						Log_1h_1y_TA
10	XMLDataE:	Log1		P2	General Pro	long	OPC Single Log:Log Ter	OPC_5s_5h
11		Log2	1	P4	General Pro	long	OPC Single Log:Log Ter	OPC_5s_5h
12	XMLDataE:	Log2		P2	General Pro	long	OPC Hierarchical Log:Lo	OPC_6s_6h
13	XMLDataE:	Log2						Log_5m_1w_TA
14	XMLDataE:	Log2						Log_1h_1y_TA
15	XMLDataE:	Log2		P3	General Pro	long	OPC Hierarchical Log:Lo	OPC_6s_6h
16	XMLDataE:	Log2						Log_5m_1w_TA
17	XMLDataE:	Log2						Log_1h_1y_TA

Figure 64. Add Copy of Structured Property

As a result the complete PropertyLogs of "P1" and "P3" are copied right after their related source.

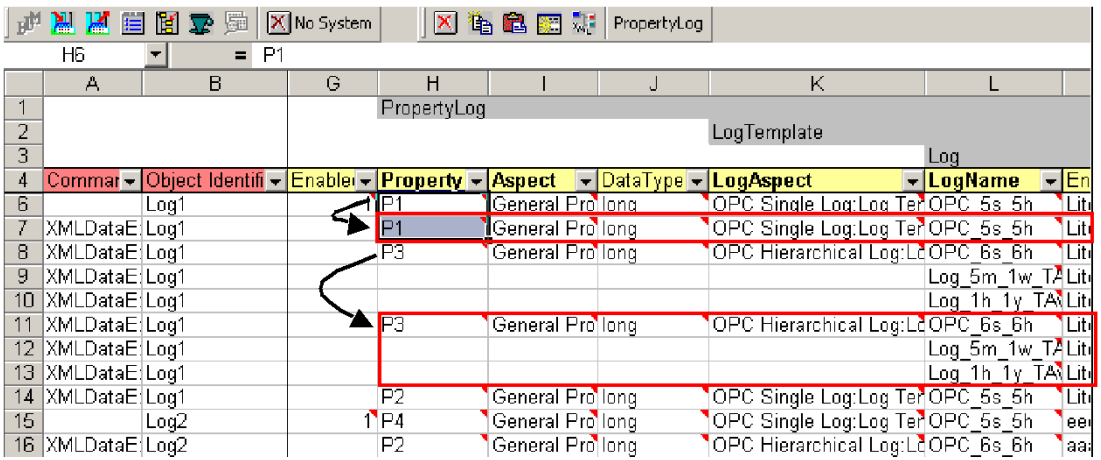


Figure 65. Copies of Structured Properties added

DO NOT USE Delete Cell

It is **not recommended** to use the MS Excel "Delete" operation that can be accessed via the menu "Edit/Delete" or via the context menu. This is because delete operations require to shift cells which results in an visual and sometimes logical disordering of cells and hence destroying the concept of mapping structural information to the cells of an MS Excel grid.

Use Clear Sheet instead.

Delete Operations

- **Clear Cell**
  The contents of a cell or a selected range of cells can be cleared by either pressing the button "Del" or performing the MS Excel menu "Edit/Clear/…"

Do not copy cells or rows by using Excel's <Ctrl> drop operation for Structured Property worksheets (also Engineering Templates).

The copied cells are still connected to the original Aspect Object and the cell comments are not updated. Delete all comments of the copied cells.

- **Delete Line**
  It is possible to delete complete lines using MS Excel functions. The effected data will however **not** be deleted from the aspect.
- **Logical Delete of Structured Property**

  Say for example you want to delete the structured property "Log" of the aspect "LogConfiguration" of object "Log1". In this case select any cell related to the structured property "Log" and press the button "Delete" in the toolbar for structured properties. The structured property that will be deleted is indicated at the right part of the toolbar as depicted in the figure below:

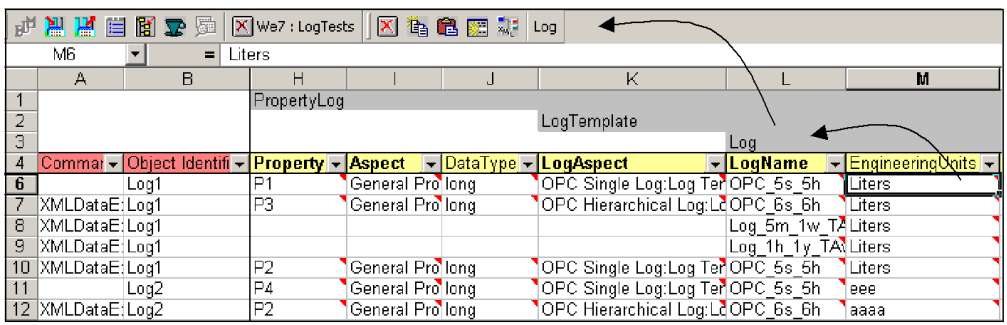


Figure 66. Deleting a structured property

Moreover the range of cells making up the structured property is highlighted and a confirmation dialog pops-up.

The Delete function of the Bulk Data Manager is a logical operation, i.e. it deletes - in this case - the complete "Log" even if only a part of the structured property "Log" is presented in the MS Excel sheet. As shown in Figure 66 above only the property "EngineeringUnit" is present in the MS Excel sheet while the structured property "Log" contains in addition the properties "NormalMaximum", "NormalMinimum", etc. When performing a logical delete on the structured property "Log" all its sub-properties are gone.

The delete function is an operation that can be applied to ranges (selection of several cells). For example if the range "M6 - M11"is selected all "Logs" of object "Log1" and the "Log" of property "P4" of object "Log2" will be deleted.

- **Delete using *delete* curve**

  Follow the steps below to delete signals or variables in a Trend Display aspect through TrendConfig.xls Bulk Data Manager template:

1. Select the specific signal or variable to be deleted.
2. Right click and select **Bulk XML Editor > Delete** curve.
3. Click **Save All Objects** in the Bulk Data Manager toolbar.

- **Delete using Delete Aspect(s)**

  Follow the steps below to delete a aspect through TrendConfig.xls Bulk Data Manager template:

  1. Select the specific aspect to be deleted.

  2. Right click and select **Aspect Commands > Delete Aspect(s)**.

- **Delete using delete Command**

  Follow the steps below in the Command column of Bulk Data Manager:

1. Select **Delete** from the dropdown list in all the respective rows that need to be deleted.
2. Click **Save All Objects** in the Bulk Data Manager toolbar.

In TrendConfig.xlsx template select **delete** under the Command column for all the filled rows and click **Save All Objects** in the Bulk Data Manager toolbar to delete an object with multiple aspects.

Applying Filters

MS Excel Auto Filter (menu "Data/Filter/AutoFilter") can be used to filter out for example only the "OPC Single Log" templates and to set for example the "NormalMaximum" value for these log templates.

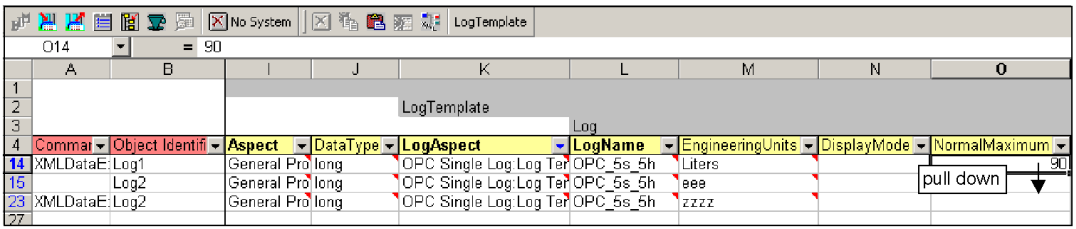


Figure 67. Applying MS Excel Filter

If we want to apply a filter that retrieves only the hierarchical log templates and their related "NormalMaximum" values it wouldn't work based on the current sheet settings as not every row contains the information that it is related to the hierarchical log template (see figure below).

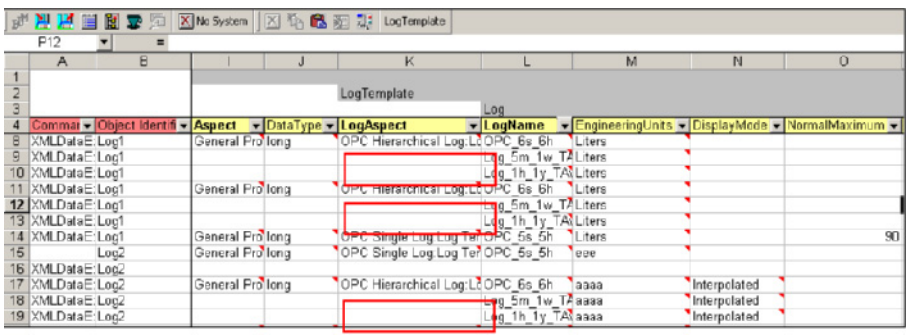


Figure 68. Missing Filter Criteria

The cells framed with the red square do not contain the value "OPC Hierarchical Log:Log Template" and hence would not be retrieved by the filter. In order to retrieve these rows as well we need to redundantly repeat this information for all related cells.

This can be set as an option in the Bulk Data Manager as depicted in the following figure.

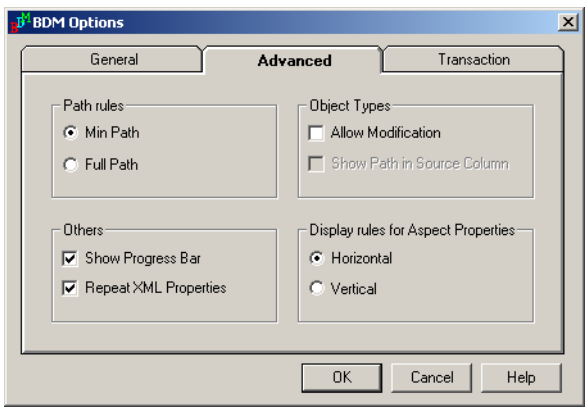


Figure 69. Option to Repeat Properties

When marking the check box "Repeat XML Properties" the values of the property "LogAspect" and other properties would be repeated row by row.

Configure Headline

Before data can be dropped to the Bulk Data Manager the headline must be configured as described in this section.

First select an object having an aspect with structured properties and drop this aspect into the Bulk Data Manager sheet. Select the properties "XSDData" and "XMLData" (if these properties are not offered in the selection dialog the aspect category dropped does not support structured properties).

Finally the Headline Configuration dialog pops-up, which is shown below.

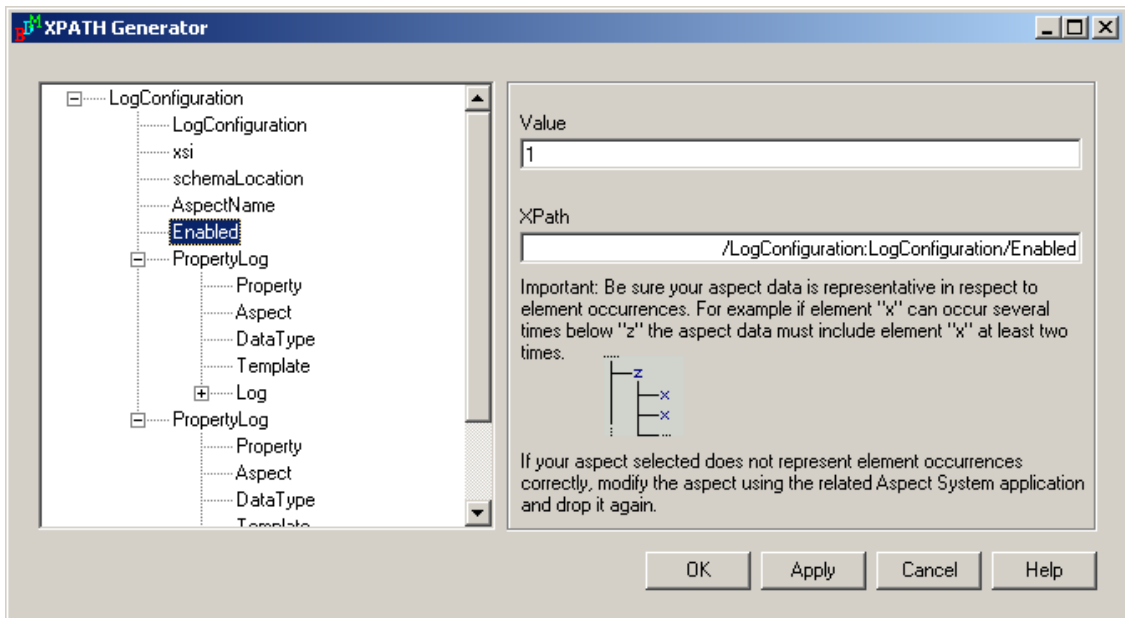


Figure 70. Headline Configuration

Within this dialog the structured property is presented in a tree. Select property by property to be included in the headline and press the button "Apply". Each property selected is automatically added at the end of the headline in the sequence selected.

Remarks:

Usually only properties on the lowest level (leaves) are of interest. If you select for example the property “Log” you get an XML presentation of all data pertaining to the “Log” inserted into the related column in the Excel sheet.

You should select the properties of interest in a sequence mapping to the hierarchy of the structured property. For example, open the node “PropertyLog” and select e.g. the node “Property”, “Aspect” and “DataType”. Continue to penetrate down the hierarchy and open the node “LogTemplate”. Again, select the nodes of interest and continue with the next lower level. After that you may come back to the top level and continue with another sub-structure.

Be sure the aspect data are representative in respect to element occurrences. For example a LogConfiguration aspect can have several “PropertyLogs”. When configuring the headline the LogConfiguration aspect **MUST** include at least two “PropertyLogs”.
Similar, a “LogTemplate” can have several “Logs” on the next level below. Again the LogConfiguration aspect **MUST** include at least two “Logs”.
If you do not adhere to this rule the resulting aspect data is not saved in the expected way

Whenever a property is inserted into the headline a path information (Xpath) is inserted into the headline. As this path information is usually not readable by a user an additional row is inserted on top of the headline having readable property names. This additional row must be inserted manually. The actual headline can be hidden (e.g. select the row and perform the MS Excel menu "Format/Row/Hide"). It is also recommended to use cell colors to indicate the substructures as illustrated by the pre-defined template for LogConfiguration.

If you need to reconfigure the headline (e.g. add additional properties) you are advised to delete the current headline and restart from the beginning.

Schema Information

In addition to the data of an aspect, which is stored in column "XMLData" an aspect system optionally may also provide schema information. Schema information represents metadata describing the actual data of the aspect.

Schema information is not yet evaluated in the current version.

Restrictions

- Only one data area per sheet can include a structured property. If several structured properties have to be dealt with they must be spread over different sheets or different workbooks.
- Column called "XSDData" must be placed directly left to the column called XMLData".
- User defined Bulk Data Manager xml sheets do not support Parameter Manager aspects with more than two levels of categories.
- Bulk Data Manager does not support the writing of OPC values which are being set for double authentication.
- While using BDM to read OPC values, if the response from the OPC server is not received within 15seconds, an error message appears. Hence, it is recommended not to use BDM to read OPC values from slow OPC connects related to field devices.

Frequently Asked Questions

This section gives answers to common questions and problems:

- *No warning or error message if Object Type in column 'Source Object' is changed.*
  It is not checked that the object in the system is of a different object type than specified in the Excel worksheet. There is no warning or error message. The reason why the object type specified in column 'Source Object' is not checked is that 'Source Object' can be used as source for copying or merging objects. With checks on, saving two times would fail.
- *Bulk Data Manager does not continue and stops responding.*
  Creation of certain aspects shows a dialog and that could be the reason the Bulk Data Manager does not continue until the user has closed the dialog box of the aspect.
- *"#NULL" does not work for Aspect Properties of data type boolean, float, integer.*
  A workbook has read an aspect property of data type e.g. "Boolean". Bulk Data Manager's string value "#NULL" cannot be used to clear this kind of aspect properties. Use the required data type value e.g. "True" for boolean for those kind of cell values.
- *Toolbar Icons look strange when they are dimmed*
  The "True Color" mode of the screen is not used. Enable the "True Color (32 bit)" mode using Control Panel > Display Properties > Settings.
- *Why can't I drag objects and aspects into an Excel worksheet.*
  An Excel workbook must be activated for working with the Bulk Data Manager. See User Interface how to activate a workbook.
- *I get error messages "Object name is not unique!" logged in the Error sheet when saving data to System 800xA applications.*
  Bulk Data Manager identifies objects and object types by name. This error message is logged if there are multiple objects with the same name in the system. You can make the Object Identification unambiguously by using path names, e.g. Plant/Area 1/Tank3.
- *I get error messages ("Object name is not unique!" and "Source object not found!") logged in the Error sheet when saving object types.*

The Bulk Data Manager option “Object Type Definition: Modification Allowed” must be checked before object types can be changed.

- *Why can’t I read or change property X of aspect Y?*
  The Bulk Data Manager can access only properties that are published by the aspect system. The aspect system must implement certain interfaces for publishing properties. In general, aspects that comply with the requirements of Integration Level “Engineering”, do publish their configuration data. Contact the provider of the involved Aspect System to get more information about the integration level and engineering support.
- *The error message: “Can't create an instance of the object type. (E_AFW_OT_NOT_OUTSIDE_STRUCTURES)” is logged in the error sheet.*
  The object type of the object is configured to enforce that all objects of this type must be placed in a structure. This means, that such objects have to be placed immediately at creation time into the structure specified by the object type. With the Bulk Data Manager, this can be achieved by specifying a path (or an ARD) in the Object Identification column. The path identifies the parent object where the new object is placed at creation time.
- *Leading zeros are missing in names and designations.*
  The cells in Excel have a default format called “General Format”. With this format, numeric text is interpreted and formatted as a number. Therefore, leading zeros are skipped, i.e. a designation “05” is formatted as “5”. In order to keep the original names and designations, you must format the cells (or whole columns) as “Text” **before** you read the data from System 800xA applications.
- *Cell-auto-format in Excel does not fit to Control Builder M data type definitions for data types like bool and string.*
  The value for a property of a string-like data type needs to be escaped in an Excel cell in case of doubling the first single quote e.g. "mystring'. During read operation into Excel cells this is done automatically by BDM. But when the user enters values he must be aware of this formatting problem and has to escape the entered strings. (There is no way that BDM can solve this.)
  For boolean-like data types BDM solves escaping the values in both in the read and the write case.
- *An error message “Formula too long” appears while using Excel’s replace feature.*

This can happen while a cell contains too many characters. Do not select the entire Excel sheet and do not select multiple rows / columns containing hidden rows / columns (which may be in between a selection).

- *Reading sizable Structured Property data (also with Engineering Templates) creates an error message.*
  The error message is "11030: Internal BDM error! -1072896657: Load Document failed : G4 Switch from current encoding to specified encoding not supported...". This happens for UTF encoded aspect data with more than 32.000 characters. Try to reduce the amount of aspect data to be displayed or try to remove the UTF encoding of the aspect data.
  Background: UTF (Unicode Transformation Format) is a way to encode unicode characters. In UTF-16, every character takes two bytes. In UTF-8, most characters (for example English characters) take one byte each and other characters (such as Asian or Arabic) take up to three bytes.
- *Toolbar buttons are empty and do not have bitmaps after startup.*
  Use the tooltip to identify the toolbar button's functionality.
- *After de-installation of Bulk Data Manager, DM & PM Application´s aspect menu "Open Properties" does not work any more.*
  Only the Engineering Platform component "Bulk Data Manager" has been de-installed. Re-install Bulk Data Manager via Engineering Platform setup.
- *In rare cases Excel or Bulk Data Manager starts with error message and /or missing toolbar buttons even though Bulk Data Manager has been installed successfully without any installation error messages.*
  - An error message like "...Cannot find COM-Addin: BDMUIAddIn..." appears.
  - An error message like "...521:Cannot open Clipboard." appears.
  - Not all Bulk Data Manager tool bar buttons appear.

  First try to register Bulk Data Manager's Excel Addins by execution of application "..\ABB Industrial IT\Engineer IT\Engineering Studio\Engineering Platform\Bulk Data Manager\bin\LBEInstallAddins.exe".
  If this does not help then check disabled Excel Addins:

Microsoft Excel 2007 / 2010 includes an option to enable the unavailable items. Follow steps below to enable the unavailable items:

1. Click **Add-Ins** from **Office Button.**
2. Select **Excel Options**.
3. In **Manage**, select **Disabled Items**.
4. Click **Go**. The **Disabled Items** window appears.
5. Select the required items.
6. Click **Ok**.

For previous Excel versions check the registry key "HKEY_CURRENT_USER\Software\Microsoft\Office\12.0\Excel\Resiliency \DisabledItems" and look for values of type "REG_BINARY" containing value data like "...bulk data manager\bin\..." and delete those kind of values.

- *Re-installing BDM in different directory and using another user account shows errors*. Bulk Data Manager was installed on a PC, later un-installed, and then re-installed in a different directory. After re-installation, Bulk Data Manager is used from a user account, that is not the account used for installation and not member of the user group "Administrators".
  When Excel is started, immediately a number of error message boxes containing the following texts appear:
  - "LBEMarcos.xla could not be found. Check the spelling of the file name, and verify that the file location is correct."
  - "11030: Internal BDM error!"
  - "11012: Cannot find LBEMacros Addin.

  Please shutdown Excel and start LBEInstallAddins.exe" Quit Excel after the warning messages described above and run the installation helper program 'LBEInstallAddins.exe' located in "<installation path>\Bulk Data Manager\bin".
  Ignore the following 6 error messages:
  * "Write access to registry denied. (HKLM\SOFTWARE\ABB\Engineer IT\Engineering Studio\SystemModules\BulkDataManager)"
  * "Registering ABBLBEGlobal failed."
  * "Registering LBEPropsAddIn failed."

* “Registering BDMUIAddIn failed.”
* “Registering LBEAddIn failed.”
* “Registering mAllAddIn failed.”

The Excel Add-in “LBEMacros.xla” is now installed for the current user. This procedure is necessary once a time for each new user account.

Error Messages

If errors occur during loading and saving data from/to System 800xA applications, these error messages are logged into a separate “Error Sheet”. The logged error messages include a short description, a link to the cell causing the problem, and the error codes and descriptions delivered by the System 800xA platform.

Problem Reporting

Provide the following additional information while reporting problems (refer to Select the Version check box.) regarding Bulk Data Manager:

- The involved Excel workbook including the “Errors” sheet that may have been generated by the Bulk Data Manager.
- A screen-shot of the Bulk Data Manager with any error dialog.
- An AFW file (created by the Import / Export tool of System 800xA with the objects and object types that might be involved in this error.

Section 4 Bulk SPL

Feature Pack Functionality

Overview

Bulk SPL is a component of Engineering Workplace.It is an Excel template (*.xltm*) with predefined add in to configure steps, transition, jump, parallel step, and selections. It is used along with Function Diagram (FD) and Control Builder M (CBM).

Bulk SPL template is used for:

- Configuring and modifying steps and transitions in Sequence1D/Sequence2D in Function Diagram.
- Configuring logic for the steps and transitions.
- Configuring SFC overview diagram in SCM/Programs in Control Builder M (CBM).

User can configure SFC or SPL using Bulk SPL template even if System 800xA is not installed in a system. For workflow refer Bulk SPL for Offline Engineering on page 174.

Prerequisite

Install **Visual Studio Tools for the Office system 3.0 Runtime** and **Service Pack 1**.

Features

- Template supports Microsoft Office 2007/2010.
- Add in is installed along with the installation of the Engineering Studio.
- SPL_SFC_Overview and SPL_Detailed sheet support for the Function Diagram.

- SPL overview diagram support for CBM.
- User can select and insert step, transition, parallel branch, jump and selection from the toolbar.
- Steps, transition and jump are represented with various color codes.
- Template supports copy paste functionality for steps and transitions.
- Option to identify the start and end of the column sequence.

Directory Locations

Bulk SPL template stores files in the following directories:

- For 32 bit system:
  <drive>:\Program Files\ABB Industrial IT\Engineer IT\Engineering Studio\Function Designer\bin\BulkSPLbin

  For 64 bit system:

  <drive>:\Program Files (x86)\ABB Industrial IT\Engineer IT\Engineering Studio\Function Designer\bin\BulkSPLbin
- *<drive>:\Program Files\ABB Industrial IT\Engineer IT\Engineering Studio\Engineering Templates*

Short cut for **Engineering Templates** folder is available on the Desktop.

Template Settings

Keep the macro security setting as **Enable all Macros for Bulk SPL** (Microsoft Excel 2007 / 2010).

1. Launch Microsoft Excel 2007 / 2010.
2. Click **File** button placed at the top left corner of the Excel application.
3. Click **Excel Options**.
4. Click **Trust Center** in the opened **Excel Options** dialog window.
5. Click **Trust Center Settings**.
6. Click **Macro Settings**.

7. Check **Trust access to the VBA project object model.**
8. Select **Enable all macros**.
9. Click **OK**.

User Interface

Context Menu

Right-click on SPL_SFC_Overview sheet to access the context menu:

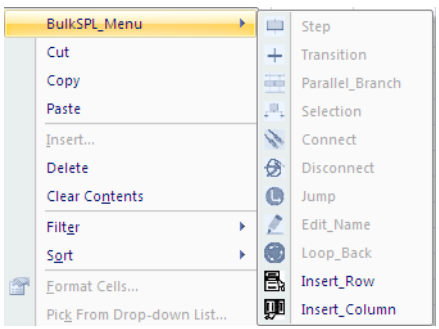


Figure 71. Context Menu

Table below describes the Bulk SPL context menu:

Table 12. Bulk SPL Context Menu

Menu Command	Description
Connect	Connects to different steps and transitions. Right click on a step/transition and select the connect option to display the list of transitions/steps.
Disconnect	Disconnects steps or transitions. Right click on a step/transition and select the disconnect option to display the list of steps or transition to which the current component is connected.

Table 12. Bulk SPL Context Menu

Menu Command	Description
Jump	Defines the conditional transfer to step.
Edit_Name	Allows user to change the name, number and description of a step or transition.
Loop_Back	Connects any transition to the initial step.
Insert_Row	Adds two new rows in Bulk SPL template (SCM/Programs).
Insert_Column	Adds a new column in Bulk SPL template.

Toolbar

A subset of context menu is available in the toolbar command.

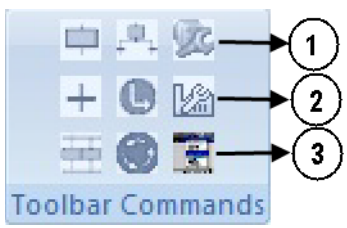


Figure 72. Toolbar

Following Table describes the Toolbar icons:

Table 13. Toolbar Commands

Sl. Number	Command	Description
1	WriteToCBM	To write the final diagram on Control Builder M (SCM/Programs).

Table 13. Toolbar Commands

Sl. Number	Command	Description
2	WriteToFD	To write the final diagram on Function Diagram.
3	Insert_Obj_Path	To insert a new object path.

Working with Bulk SPL Template

The following sections describe:

- Different template sheets (Bulk SPL Template Sheets on page 161).
- Working with SPL_SFC_Overview sheet (SPL_SFC_Overview sheet on page 162).
- Working with SPL_Detailed sheet (SPL_Detailed sheet on page 167).
- How to work with Bulk SPL template using offline engineering (Bulk SPL for Offline Engineering on page 174).

Bulk SPL Template Sheets

- **SPL_SFC_Overview**- is used to configure the SPL/SFC Overview. It is supported only if it is configured in Function Diagram or Control Builder M. User can configure step, transition, jump, parallel branch and selection branch of the sequence diagram.
- **SPL_Detailed**- is used to configure logics inside the steps and transitions. It is supported only if it is configured in Function Diagram. All the creations and modifications of logics in steps and transitions are made in the detailed sheet.
- **BDM_SPL_ErrorSheet**- is displayed only when there are validation errors. A separate error sheet is generated in the template to display the basic User Interface errors and basic configuration errors.

Current version of Bulk SPL template does not support CBM in SPL_Detailed sheet.

SPL_SFC_Overview sheet

Features of overview sheet:

- Modifying name, number and description for steps/transition.
- Adding steps, transitions, and jumps.
- Deleting steps and transition.
- Navigating to the detailed step/transition.
- Copy paste of steps and transition.
- Cut paste of steps and transition.

Steps to work with SPL_SFC_Overview sheet:

1. Create the Function Diagram in SCM or FUD code block with Sequence1D or 2D.

The Bulk SPL supports diagrams only with single Sequence (1D or 2D). Multiple Sequences (1D or 2D) in a single Function Diagram are not supported.

2. Right click on an object in the Functional Structure and select **Advanced >Engineering Templates**.

The Engineering Template folder opens.

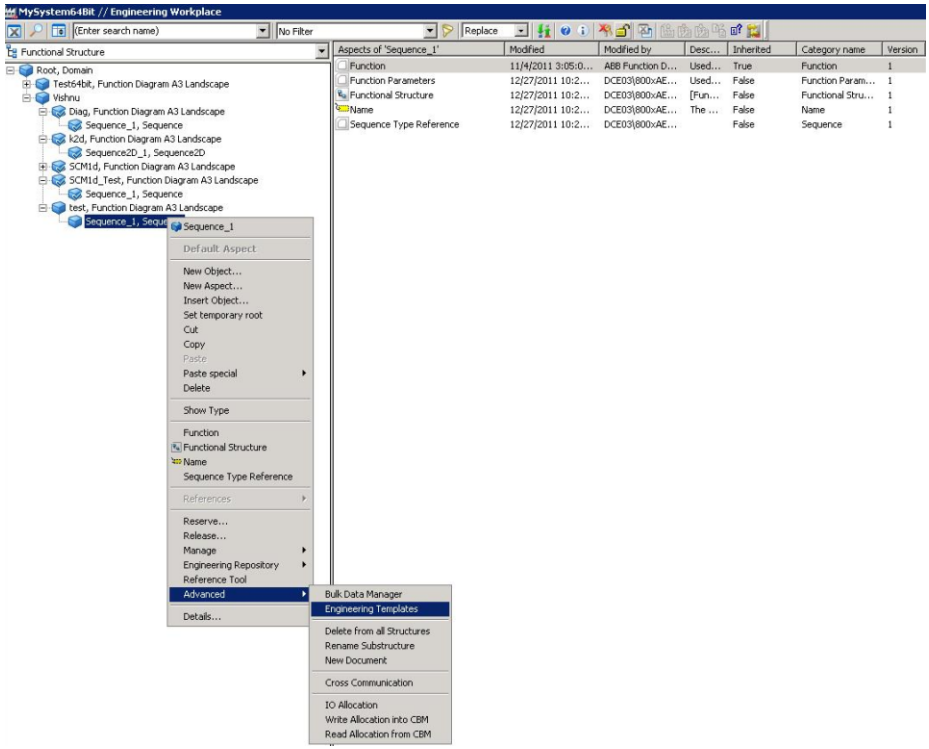


Figure 73. Bulk SPL Template

3. Select the Bulk SPL template and click **Open**.

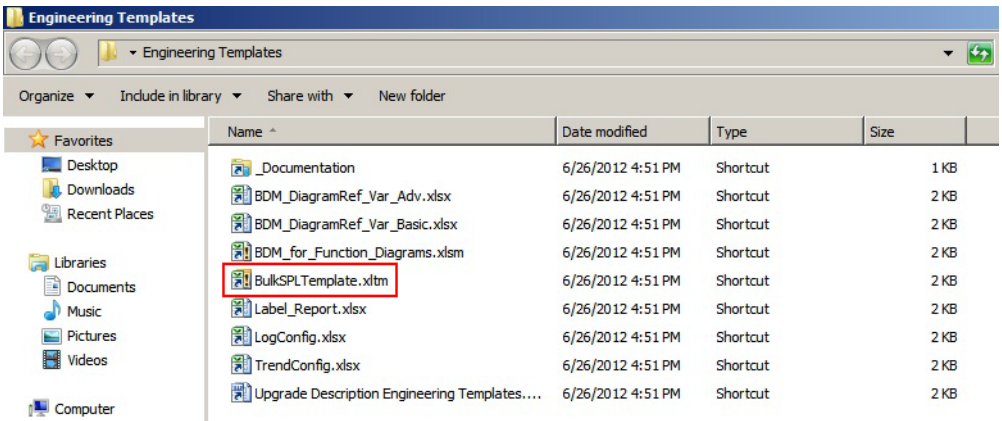
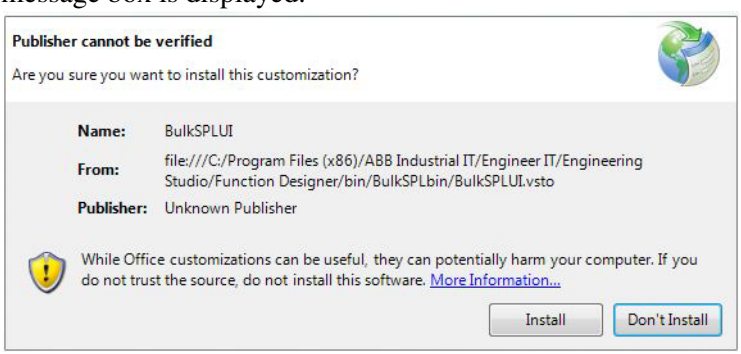


Figure 74. Engineering templates

If a different user opens the Bulk SPL template for the first time following message box is displayed:



Click **Install** to continue

4. To insert a new object path, in the Bulk SPL template click **Insert_Obj_Path** icon.
5. Select the object or the Function Diagram with sequence in the **Insert ObjectPath** dialog.

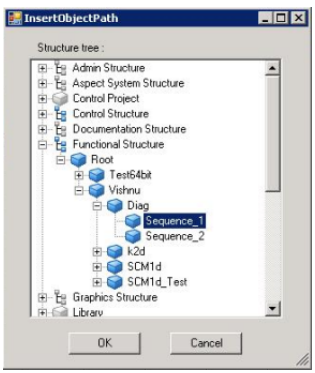


Figure 75. Insert Object Path

6. Select and click step, transition, jump and selection from the toolbar.

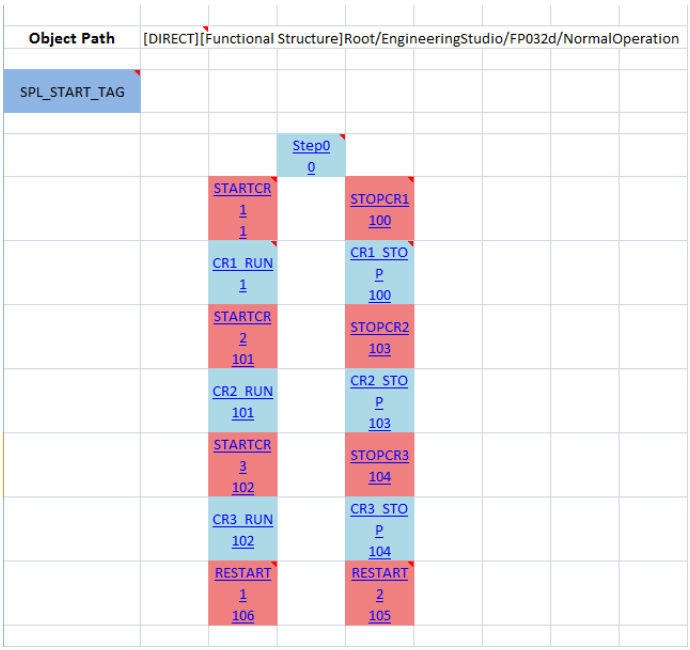


Figure 76. Sequence 2D

The steps, transition and jumps is displayed with three different color codes (blue, red and green for step, transition and jump respectively).

Renaming is possible only for steps and transitions. Renaming of jumps is not possible.

7. Click the **Loop_Back** icon to connect any transition to initial step.

To navigate to the SPL_Detailed step/transition click the step and transition from the SPL_SFC_Overview sheet.

8. Click **Write to the Function Diagram** to write the final diagram on the Function Diagram.

After performing **Write to the Function Diagram**, reopen the modified sequence.

User cannot change the predefined initial step in Bulk SPL template.

If SPL Overview diagram have minor routing issue (connections are slightly overlapping the other symbols), right-click on the page and select **Rerouting page > Reroute**.

Sequence 2D does not display **On Off** labels when written from Bulk SPL Template to Function Diagram. This does not affect the functionality and can be ignored.

Creating SPL Overview diagram in CBM

Creating SPL in Control Builder M is similar to creating the SPL in Function Diagram.

Steps:

1. Create a Single Control Module (SCM) in application.
2. Open the SCM editor and right click on code tab.
3. Change language to **Sequential Function Chart (SFC).**
4. In Bulk SPL template, select **Insert_Obj_Path** icon.
5. Select the desired SCM and configure.

Click **Write to the CBM** to write the final diagram on Control Builder M.

If a jump is configured for a selection of two branches, then there has to be a dummy transition between the two transitions.

SPL_Detailed sheet

SPL_Detailed Sheet is used only for the sequence created using the Function Diagram. If a new steps or transition is inserted in the SPL_SFC_Overview sheet, details of this step and transition are created in the SPL_Detailed sheet.

Following sequences are supported in the SPL_Detailed sheet:

- Sequence 1D

- Sequence 2D

Sequence 1D

After inserting the steps and transitions in the SPL_SFC_Overview sheet, the steps and transitions are created in the SPL_Detailed sheet with default actions, Tmax and Tmin rows.

Comp Name	Input Reference DataType	Logic Operand 1	Logic Operand 2	Logic Operand 3	Logic Operand 4	Logic Operand 5	Tr/Diagram Variable	Output Diagram Reference	Output Reference DataType
WF3_CMD							WF3_CMD		
Tr_Logic	bool					and(bool)(5)	Tr		
	bool								
	real				le(real)(2)				
	bool								
	bool								
Step13							Step13		
P0_Action							BC1BC2STOP	BC1BC2STOPAII	bool
							BC3BC4STOP	BC3BC4STOPAII	bool
N_Action	real					gt(real)(2)	BC1_IT01_OK	BC1ITOK	real
	dint					add(dint)(2)	BC1BC2TOTALIT	ITTOTALBC12	dint
	dint								
P1_Action							CR1_NOTRUN	CR1NOTRUN	bool
							CR2_NOTRUN	CR2NOTRUN	bool
TMin									
TMax									

Figure 77. Sample 1D Step and Transition

Sequence 2D

If a new Sequence 2D is inserted, each transition is inserted with one AND block and one input connected to default variable. Each step is inserted with one StepInfo block with predefined values for the input ports.

Configuring Steps

A	J	K	L	M	N	O	Q	R	S
Comp Name	Logic Operand 3	Logic Operand 4	Logic Operand 5	DefaultPort_Value	DefaultStepComponent	Tr/Diagram Variable	Output Diagram Reference	Output Reference DataType	
Step0						Step0			
P0_Action									
N_Action				Sequence2D_1_FromSFC.StepInfo1	SFCStep				
				0					
				Step0.X					
				Step0.T					
P1_Action									
TMin									
TMax									

Figure 78. Configuring Steps

Under step name five default rows are displayed for Sequence 2D:

- P0_Action
- N-Action
- P1_Action
- TMin
- TMax

Output variables should not be defined next to the SFCStep and in N_Action output reference variables should be defined by inserting a new row.

To insert a new row, right click on the row header and insert a new row.

User can edit and connect Diagram Variable, Communication Variable to the TRmin and TRMax. The default properties and functions are provided with the respective information next to the cell.

Tmin and Tmax should be defined either at step level or action level.

Configuring Transition

A	B	C	D	F	G	H	I	J	K	L
Comp Name	Description	InPort_Value	Diagram Variable	Input Diagram Reference	Input Reference DataType	Logic Operand 1	Logic Operand 2	Logic Operand 3	Logic Operand 4	Logic Operand 5
Tr1										
Tr_Logic		Sequence2D_1_ToSFC.OnSeq								andlbool(2)
Tr100										
Tr_Logic		Sequence2D_1_ToSFC.OffSeq								andlbool(2)

Figure 79. Configuring Transition

In transition, **AND** block is inserted in **Logic Operand 5** column with two inputs. Each transition is inserted with default two rows in case of Sequence 2D.

Logical operands has to be defined in decreasing order starting from Logic Operand5.

Variable type **Crosscomm** and **DiagVar** displayed in SPL_Detailed sheet (SCM) are referred as **CommunicationVariable** and **DiagramVariable** in CBM/FD respectively.

In Function Diagram, in transition detailed logic page if the components are placed outside the boundary, user has to change the template settings (**Menu > Measurement & Size**) and move the blocks within the template limit before saving the diagram.

Working with Operands

Logic Operands should be configured from right to left. In sequence, five logical operands can be configured. In parallel there is no limit for number of logical operand.

To add operands, right click the logical operand column, select **Operands_Menu** and click on **Add Logical Operands.**

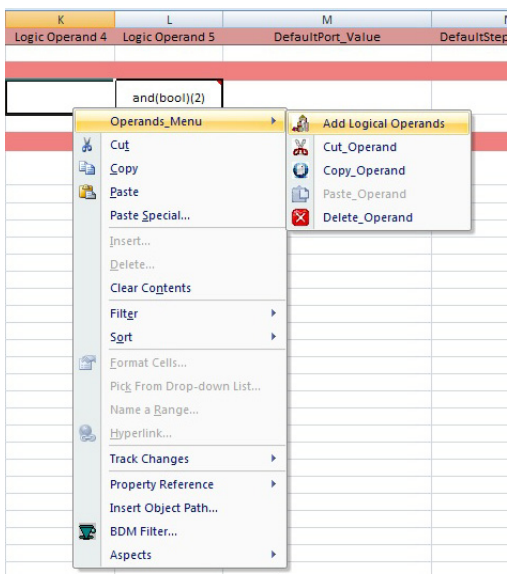


Figure 80. Operands Menu

Table below describes the Operands_Menu:

Table 14. Operands Menu

Menu Command	Description
Add Logical Operands	To add a logical operand. Figure 81 displays various operands.
Cut_Operand	Cut a particular operand.
Copy_Operand	Copy a particular operand.
Paste_Operand	Paste the cut/copy operand.
Delete_Operand	Delete an operand.

Add Logical Operand dialog box.

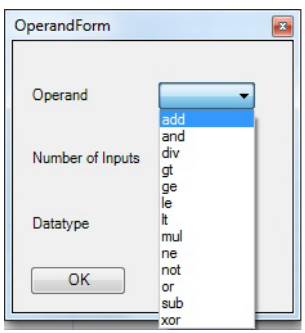


Figure 81. Add Logical Operand

Number of Inputs: Based on number of inputs entered, number of rows will be added in the SPL_Detailed sheet.

The block containing block name is taken as output for next block.

After writing sequence from Bulk SPL template to Function Diagram, the step (action page) and transition detailed logic page is truncated due to display of components outside the boundary. To overcome this user has to change the template settings (**Menu > Measurement & Size**) and move the blocks within the template limit before saving the diagram.

Modifying existing Sequence

Modification is possible only for the sequences which are created in Function Diagrams.

User can modify the existing sequence by giving the path of the sequence in the template using **Insert Path** option.

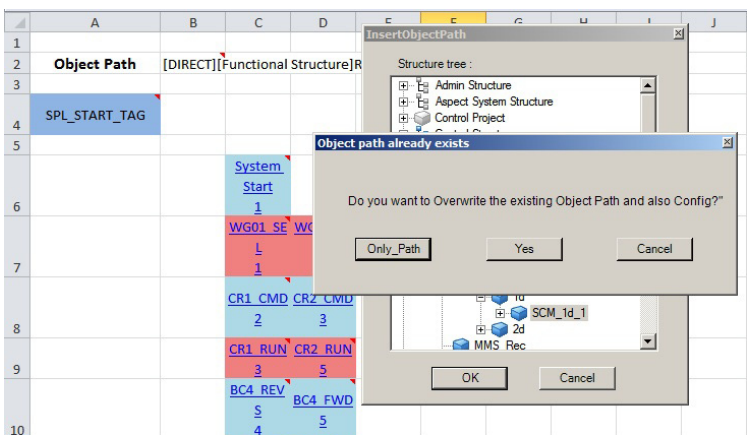


Figure 82. Modification of Existing path

Following three options are displayed if the existing path is modified.

Table 15. Overwriting of existing path

Option	Description
Only_Path	Object path is changed, but existing sequence in the template is not overwritten.
Yes	Object path is changed and existing sequence in the template is overwritten.
Cancel	To cancel the modification of existing path.

Bulk SPL for Offline Engineering

Configuring Bulk SPL without installing System 800xA Engineering Studio.

Prerequisite:

Install **Visual Studio Tools for the Office system 3.0 Runtime** and **Service Pack 1** where offline engineering is to be performed.

Steps:

1. Create a Sequence 1D/2D using Bulk SPL template in the system where System 800xA Engineering Studio is installed. Open the Bulk SPL template and refer to the Sequence 1D/2D for which offline engineering has to be performed.
2. Save the templates in.*xltm* format.
3. Copy the bin folder from the system where System 800xA Engineering Studio is installed and paste at any location in the system where Offline Engineering for Bulk SPL template is to be performed.

Location of the bin:

For 32 bit system:

C:\Program Files\ABB Industrial IT\Engineer IT\Engineering Studio\Function Designer\bin\BulkSPLbin

For 64 bit system:

C:\Program Files (x86)\ABB Industrial IT\Engineer IT\Engineering Studio\Function Designer\bin\BulkSPLbin

4. Copy the template which was saved in step2 and paste it in the same bin folder in the system where offline engineering has to be performed.
5. Run the *BulkSPL_OfflineSupport.exe* from the bin folder.
6. After working with Bulk SPL template offline, save the template in.*xltm* format.
7. Copy the bin folder from the offline engineering system and paste at any location in the system where System 800xA system is installed and running.

8. Run the *BulkSPL_OfflineSupport.exe* present in the bin folder copied in step 6 from System 800xA Engineering Studio bin folder.

User can use Bulk SPL template in System 800xA Engineering Studio.

Section 5 IO Allocation

IO allocation is the process of allocating signal objects to free channels of compatible I/O boards.

The allocation function:

- supports IO allocation of signal objects in bulk as well as in single signal quantities
- allows signal engineering by entering signal data at an early stage without the necessity to create the specific I/O boards at that time. The allocation of signals to boards can be done in a later step.

Starting with 800xA 5.0 SP2 IO Allocation also supports allocating signal objects to channels of modules of PROFIBUS devices.

Signal engineering including IO allocation can be divided in six activities:

1. Creation of signals.
   For example from object types of CBM_Signal object type group.
2. Signal data entry.
3. Creation of I/O boards.
4. IO allocation of signals to I/O boards.
5. Write allocation data to Control Builder M.
6. Read allocation data from Control Builder M.

Supported Hardware Libraries

IO Allocation by default supports the following AC 800M Connect hardware libraries:

- CI856S100HwLib
- S800CI801CI854HwLib
- S800CI830CI851HwLib

- S800CI830CI854HwLib
- S800CI840CI854HwLib
- S800IoModulebusHwLib
- S900IoCI851HwLib
- S900IoCI854HwLib
- CI851PROFIBUSHwLib
- CI854PROFIBUSHwLib

IO allocation support for new / additional hardware units can be added by using the HWDProcessor utility.

IO Allocation Functions

The allocation function:

- places the allocated signals below the I/O board hardware unit in Control Structure visible in Plant Explorer of Engineering Workplace
- optionally creates an application global variable (depends on allocation scenarios)
- adds the variable(s) as connection(s) in the I/O board hardware unit
- writes the signal data into the corresponding channel of the I/O board hardware unit in Control Builder M (CBM), containing Settings, Connections and Scaling information.

Additionally the allocation function supports to:f

- build and use own generic signal objects
- configure mapping of IO properties to signal properties
- add IO allocation support for new I/O board, PROFIBUS device and HART device hardware units (using the HWDProcessor utility).
- allocate HART devices to free channels of compatible I/O boards (HART Multiplexers)

- create variables and connecting them to modular PROFIBUS devices.

To be able to use instances of HART and PROFIBUS device object types (installed via the Device Library Wizard) with the IO Allocation function these object types have to be pre-processed using the HWDProcessor utility.

Applying the HWDProcessor utility options to a non appropriate library may damage the library.

Before You Start

System extension *Signal Extension for AC 800M Connect* has to be loaded before doing IO allocation for AC 800M controllers. This in turn requires the following system extensions to be loaded:

- AC800M Connect
- DM & PM Application
- Engineering Base

The system extension *Signal Extension for AC 800M Connect* supply all necessary board, channel, and signal information on predefined object types.

Getting Started

IO-Allocation User Interface

IO Allocation is a view (see Figure 83), which gives effective support for doing IO allocation. Its user interface displays all allocation relevant data. Typically all operations can be done via drag and drop.

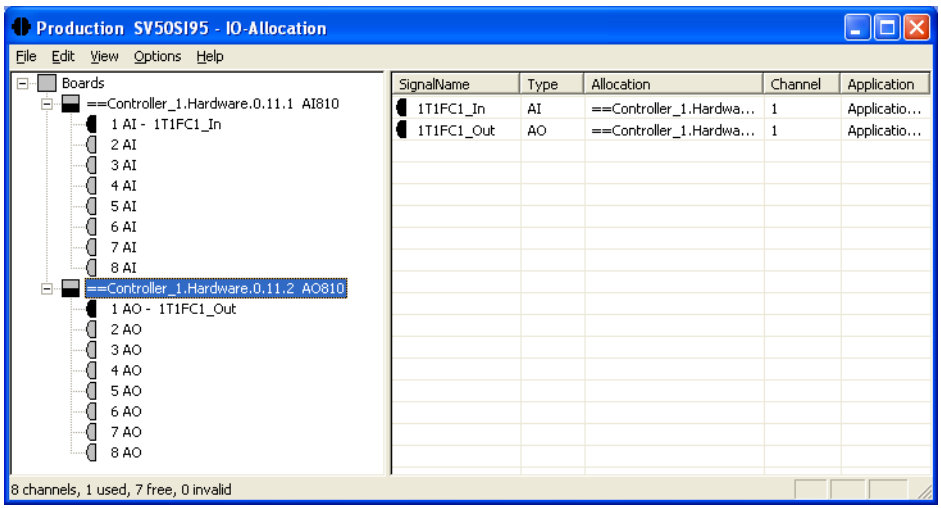


Figure 83. IO Allocation View

It displays the objects (boards, channel, and signals or devices) controller independent, only focusing on the IO allocation data.

Additional to the simple IO allocation with single signals/channels it gives the possibility to do bulk and smart IO allocations in an easy and fast way, as for example finding suitable free channel(s) for one or more given signal(s).

Moreover the view detects typical invalid allocations, like missing or duplicate data, and allows to correct this data.

DP910x I/O boards offer discontinued channel numbers. The actual channels to use are displayed in the Allocation tool.

The IO Allocation tool is environment sensitive: The window title shows the environment and the system the tool is started in.

Starting IO-Allocation

Select a board object in the Control Structure, click the right mouse button and select menu item **Advanced > IO Allocation**. The IO Allocation view will be launched, displaying the selected board and its channels in the left view and the allocated signals in the right view.

Start IO Allocation on a node or project object. All boards, channels, and allocated signals below the starting object will be displayed at startup. Basically you can start IO Allocation from any object in any structure.

Signal Engineering and IO Allocation

The following subsections describe a basic workflow for signal engineering including IO allocation.

Creation of Signals

In Plant Explorer of Engineering Workplace create your signals in the Functional Structure, either manually by instantiating the object types of the CBM_Signals object type group or by using the Function Designer.

User may as well use the Bulk Data Manager to create the signals including all the CBM_SignalParameter settings.

Available signal object types are:

Table 16. Signal Object Types

Signal Object Type Name	Description	Data Type
CBM_AIS	Analog Input Signal (input)	RealIO
CBM_AOS	Analog Output Signal (output)	RealIO
CBM_DIS	Digital Input Signal (input)	BoolIO
CBM_DOS	Digital Output Signal (output)	BoolIO

Table 16. Signal Object Types (Continued)

Signal Object Type Name	Description	Data Type
CBM_Pulses	Pulse Signal	Pulse
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

Signal Data Entry

Select the CBM_SignalParameter aspect of the created object.

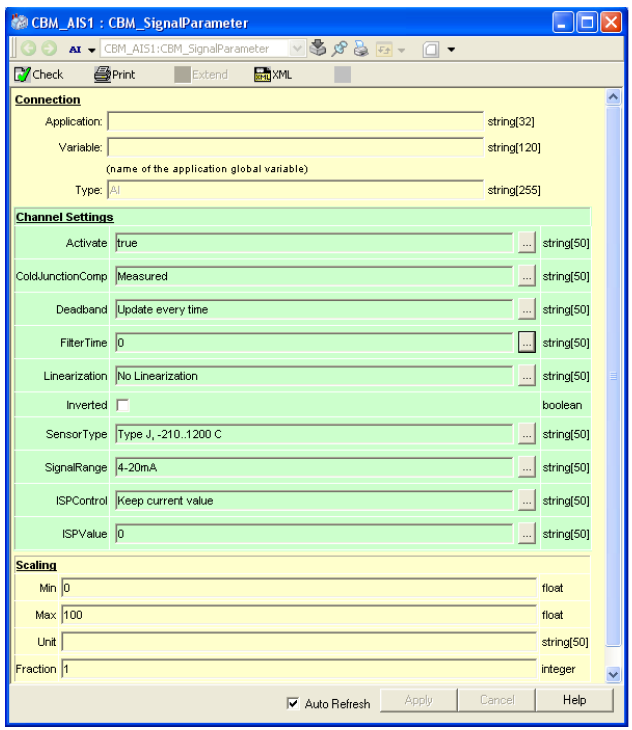


Figure 84. CBM_SignalParameter Aspect for CBM_AIS (not yet Allocated to a Board).

In the preview window of the Plant Explorer or in the popup window on this aspect you can fill in the values for various properties like Min and Max value, Signal Range or Application. These values will be written into the Control Builder M at a later stage. The values can also be written using the Bulk Data Manager. You can also edit the CBM_SignalParameter aspect from within the IO Allocation tool, see Editing Signal Parameters

For **SignalRange** property as part of the CBM_SignalParameter aspect, 4-20mA or 4..20mA can be used interchangeably without any impact on the functionality.

.

For additional information on the properties and allowed values see the Tool Tips and the "..." button in this preview.

The properties and their picklists offered in the user interface for the aspects CBM_SignalParameter and CBM_PulseSignalParameter are context sensitive regarding the IO Board the signal is allocated to.

Before the signal is allocated the aspect shows the superset of the properties of all board types. Further the picklist of a property shows the superset of values for all board types having this property.

The allowed hardware properties and the actual values allowed for these properties can be looked up from the CBM Hardware Unit dialogs at that time.

The default values provided in the properties of the aspects CBM_SignalParameter and CBM_PulseSignalParameter are defined with reference to a generic board type. During allocation of a signal object to a concrete board instance these default values are not automatically adapted to the values allowed by the concrete board type. You have to check yourself if the default values fit, otherwise you may get corresponding error messages during writing the allocation data into CBM.

Creation of Boards

In the Control Structure, create the IO boards below the related controller / interface objects. Again, this can be done using the Bulk Data Manager.

Remember to connect the application(s) also to the controller object.

IO-Allocation of Signals to Boards

After starting the IO Allocation tool (see Starting IO-Allocation) depending on the start context you either see boards and / or signals or you insert boards into the boards view (left pane) and signals into the signals view (right pane).

Then you can select / filter the not allocated signal objects. If needed you can edit the application the signal is used in and the signal parameters.

To allocate the signal you drag and drop them from the left pane onto the Boards object or on a single board object in the left pane.

Write Allocation Data to Control Builder M

After you have done the necessary allocations you have to write the data to Control Builder M using the corresponding context menu item, For more details refer to Writing Allocation into CBM.

Different allocation scenarios as described in Signal Information are supported.

Read Allocation Data from Control Builder M

During commissioning properties of the I/O boards, etc. may have been changed. In order to update the signal objects with the changes done directly in the Control Builder M, IO Allocation offers a possibility to read the data back from the Control Builder M. For more details refer to Reading Allocation from CBM.

Signal Information

The IO Allocation function covers different allocation scenarios regarding the connection of a signal to an IO variable. The desired allocation scenario can be set in the CBM_SignalInformation aspect of the signal objects as shown in Figure 85.

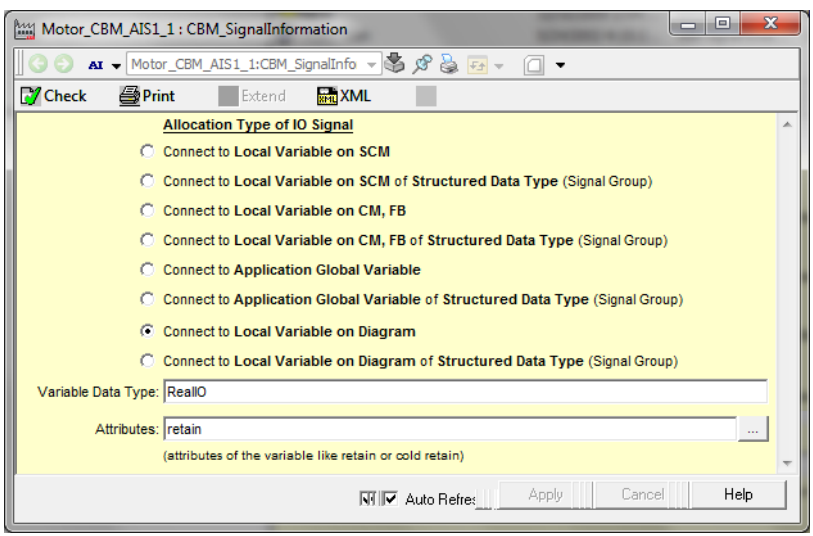


Figure 85. Allocation Scenarios

While inserting CBM signals in newly created Function Diagrams, **Connect to Local Variable on Diagram** is the default selection in the **CBM_Signal Information** aspect. While working with restored Function Diagrams, the **Connect to Application Global Variable** is selected by default.

- The property "**Variable Data Type"** defines the data type of the variable created by Allocation (in case of a variable is created)
- The property "**Attributes**" represents the identically named property related to a variable in an application.

Connect to a Local Variable on Single Control Module

In this case it is assumed that the variable used to connect to the IO does already exist in a Single Control Module. The variable name must match the object name of the signal as depicted in the following example.

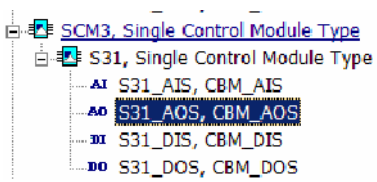


Figure 86. Connect Local Variable on SCM

In the example above there are 4 signal objects related to the Single Control Module S31 (the closest parent object of type Single Control Module is considered).

It is assumed in this example that there are 4 variables defined in the Single Control Module “S31” which are called “S31_AIS”, etc. as depicted below.

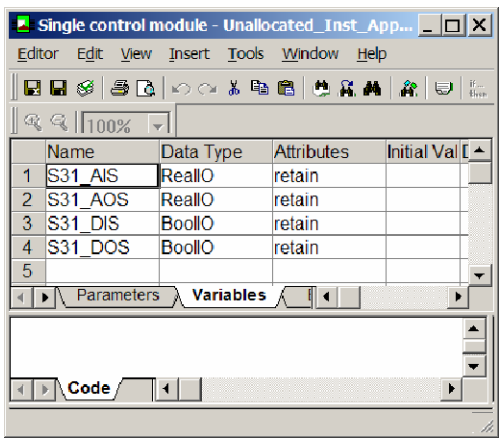


Figure 87. Variable Definition on SCM

Once the signals have been allocated and the allocation data have been written to the Control Builder M the **connection string** entered in the IO Board looks for example like “Application1.SCM3.S31.S31_AIS”.

If the Function Designer is in use:

- A Control Builder Name is created. In this case the variable name must match the Control Builder Name. This allows to de-couple the variable name (subjected to IEC 61131 restrictions) from the object name.
- The local variables within the Single Control Module are automatically created.

Connect to a Local Variable on Single Control Module of Structured Data Type

This case is equivalent to the case Connect to Application Global Variable of Structured Data Type. There is only one difference:

- The variable to connect to must be defined in the Single Control Module. The variable name must match the name of the SignalGroup object.

The signal group objects must be located below the Single Control Module in the Functional Structure. More than one signal group can be used.

If the Function Designer is in use:

- A Control Builder Name aspect is created for a signal group. In this case the variable name must match the Control Builder Name. This allows to decouple the variable name (subjected to IEC 61131 restrictions) from the object name.
- The local variable within the Single Control Module is automatically created

Connect to Local Variable on Control Module or Function Block

In this case it is also assumed that the variable used to connect to the IO does already exist in a Function Block Type or a Control Module Type. Each signal object (which must be placed below an instance of a Function Block / Control Module Type) maps to one local variable. The mapping is defined in the aspect “Relative Name”, i.e. the Relative Name must match the variable name (see figures below).

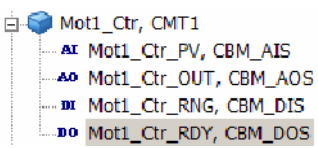


Figure 88. Motor Control with its Related Signal Objects

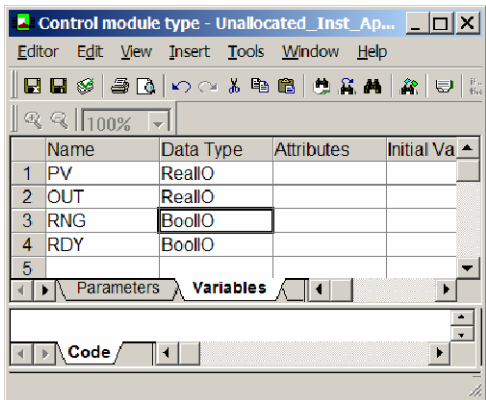


Figure 89. Variable Definition in Control Module

The **connection string** entered in the IO Board looks for example like "Application1.Mot1_Ctr.PV".

Connect to Local Variable on Control Module or Function Block of Structured Data Type

This case is equivalent to the case Connect to Application Global Variable. There are two differences:

- The variable to connect to must be defined in the Control Module / Function Block
- The signal group object must have a Relative Name that maps the name of the local variable in the Control Module / Function Block related to the signal group.

The signal group objects must be located below the Control Module / Function Block in the Functional Structure. More than one SignalGroup can be used.

Connect to Application Global Variable

In this case a global variable is created (if not already existing) in the application identified in the aspect CBM_SignalParameter or CBM_PulseSignalParameter. The name of the variable is derived from the object name. If the Function Designer is in

use there may optionally be a Control Builder Name created. In this case the variable name is derived from the Control Builder Name.

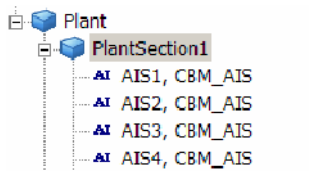


Figure 90. Connect to Global Variable

In this example 4 global variables will be created called AIS1, etc.

The **connection string** entered for the allocated IO Board looks for example like "Application1.AIS1".

If the Function Designer is in use:

- A Control Builder Name aspect is created for each signal. In this case the variable name is derived from the Control Builder Name. This allows to decouple the variable name (subjected to IEC 61131 restrictions) from the object name.

Connect to Application Global Variable of Structured Data Type

A signal group is a composition of individual signals as depicted in the following figure:

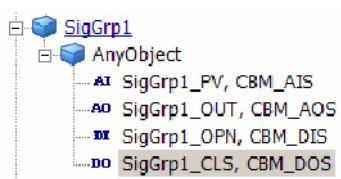


Figure 91. Example of a Signal Group

The signal group can be created as instance from scratch or instantiated from a pre-defined composite object type.

It is recommended to implement signal groups based on composite object types. For technical reasons composite object types have to be put into a library, a library extensions or have to belong to a system extension.

Related to the signal group example is a definition of a structured data type with the sub-components “PV, OUT, OPN, CLS” in Control Builder M as shown below.

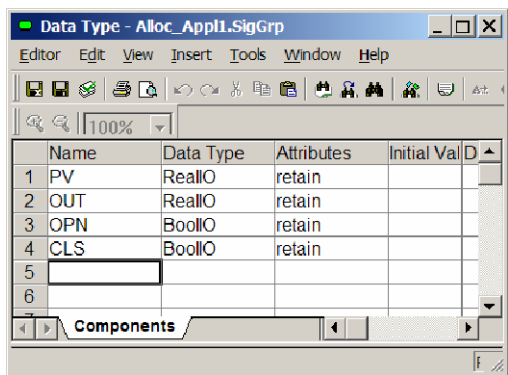


Figure 92. Structured Data Type Definition

Each signal of the signal group maps to a component of the structured data type. There is one variable (SigGrp1 in this example) generated for the signal group. The Application the variable is automatically generated into is defined in the aspect CBM_SignalParameter or CBM_PulseSignalParameter. The individual signals have to be allocated to channels of matching IO boards. According to the signal allocation a connection string is entered into the IO boards for the related channels. The connection strings entered in the IO board will point to the path down to the subcomponent of the structured variable, as for example “Application1.SigGrp1.PV”.

Settings for the different aspects:

- The signal group object must have an aspect of category CBM_SignalInformation with the following settings:

  - Radio button “Connect to Application Global Variable of Structured Data Type”
  - Variable Data Type: data type of the structured variable
  - Attributes: attribute of variable (e.g. retain)
- Within a signal object the CBM_SignalInformation aspect have to be set as:
  - Radio button “Connect to Application Global Variable of Structured Data Type”
  - Variable Data Type: data type of the sub-component of the structured data type that is related to the signal
- Each signal object must have a Relative Name aspect that maps the sub-component of the structured data type the signal is related to, as depicted in the figure below.

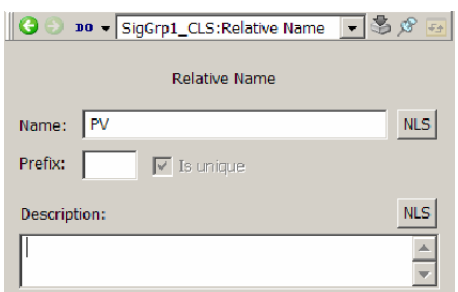


Figure 93. Relative Name Aspect

Based on this structured data type a so called signal group can be created in the Plant Explorer (either from scratch or by instantiating a pre-defined composite object type).

If the Function Designer is in use:

- A Control Builder Name aspect will be created for a signal group. In this case the variable name is derived from the Control Builder Name. This allows to decouple the variable name (subjected to IEC 61131 restrictions) from the object name.

Connect to a Local Variable on Diagram

In this case it is assumed that the variable used to connect to the IO already exists in the Diagram. The variable name must match the object name of the signal as depicted in the following example.

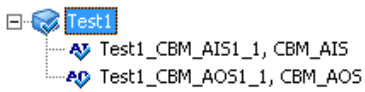


Figure 94. Connect Local Variable on Diagram

In the example above there are 2 signal objects related to the Diagram Test1.

It is assumed in this example that there are 2 variables defined in the Diagram "Test1" which are called "Test1_CBM_AIS1", etc. as depicted below.

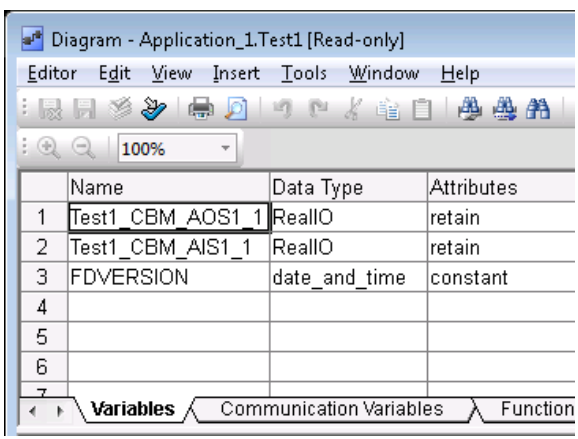


Figure 95. Variable Definition on Diagram

Once the signals have been allocated and the allocation data have been written to the Control Builder M the **connection string** entered in the IO Board looks for example like "Application1.Test1.Test1_AIS".

If the Function Designer is in use:

- A Control Builder Name is created. In this case the variable name must match the Control Builder Name. This allows to de-couple the variable name (subjected to IEC 61131 restrictions) from the object name.
- The local variables within the Diagram are automatically created.

Connect to a Local Variable on Diagram of Structured Data Type

In this case, the variable to connect to must be defined in the Diagram. The variable name must match the name of the SignalGroup object.

The signal group objects must be located below the Diagram in the Functional Structure. More than one signal group can be used.

If the Function Designer is in use:

- A Control Builder Name aspect is created for a signal group. In this case the variable name must match the Control Builder Name. This allows to decouple the variable name (subjected to IEC 61131 restrictions) from the object name.
- The local variable within the Diagram is automatically created.

Working with IO Allocation

Loading Boards

Use drag and drop to load boards into IO Allocation view. Select an object in the Engineering Workplace and drop it into the board view. All board objects, which are below the selected object (including the selected object) are added to the board view, if they are not already there.

Alternative: Start IO Allocation tool on an object in the Control Structure using the object´s context menu item **Advanced > IO Allocation**. The boards available in the Control Structure below the start object will be loaded. If signals are allocated to these boards these signals will be loaded too.

Loading Signals

Use drag and drop to load free (this means not allocated) signals into the signal view IO Allocation. Select an object in the Engineering Workplace and drop it into the signal view. All signal objects, which are below the selected object (including the

selected object) are added to the signal view, if they are not already there and if they are not yet allocated.

Alternative: Start IO Allocation tool on an object in the Functional Structure using the object´s context menu item **Advanced > IO Allocation**. The signals available in the Functional Structure below the start object will be loaded. If signals are allocated to boards these boards will be loaded too.

This also applies if you start IO Allocation from within a Function Diagram containing CBM_Signal objects.

Inserting Boards/Elements

To insert either I/O boards and allocate signals in the **IO Allocation** window:

1. Click **IO Allocation** icon from the **Quick Access** toolbar.
2. In the grid on the right side of the **IO - Allocation** dialog, all engineering signal objects inserted in the Function Diagram are shown.
3. In the tree on the left side of the **IO-Allocation** dialog, right-click the Boards object and select **Insert Board** from the context menu.
4. In the **Insert Board** dialog, navigate to the required location and select **ModuleBus** or select the required IO Board.
5. Click **Insert**.
6. Click **Close** to exit the **Insert Board** dialog.

Alternatively, follow the steps below to use **Insert Elements** option:

1. Click **Edit > Insert Elements**.
2. In the structure browser navigate to and select the element, either a board in the Control Structure or a Signal in the Functional Structure, and select it.
3. Click **Add**.

Editing Signal Parameters

To edit the properties of the CBM_SignalParameters aspect of a signal object:

1. Right-click the signal row in the right pane.

2. Click **Open Properties**.
3. In the CBM_SignalParameter popup window edit the properties.
4. Click **Apply**.

Allocating Signals

To allocate a single signal, select it in the signal view and drop it onto a channel in the board view. If you have chosen valid signal/channel association, the allocation will be executed, if not the error reason is displayed in the status bar.

To allocate multiple signals to multiple channels, select a bunch of signals in the signal view and drop them onto a board or the root object *Boards* in the board view. In this case the smart function works and finds automatically suitable channels for the signals.

Alternatively you can do this operation without drag and drop. After selecting the signal(s) in the signal view, select a target object in the board view, press the right mouse button and select menu item **Allocate**.

Additionally you can select signals using the context menu (right mouse button):

- **Select all** selects all signals in the view
- **Select free** selects only signals, which are not allocated in the view
- **Select allocated** selects all allocated signals in the view
- **Select invalid** selects all invalid signals in the view.

IO Allocation uses the ChannelNumber aspect of the signal objects to store the channel number determined during allocation for later on writing the allocation data to CBM. It is not possible to pre-set the channel number in the ChannelNumber aspect for the allocation function of the IO Allocation tool.

If you want to pre-set channel numbers, for example using Bulk Data Manager, you must not use drag&drop of the signals to boards or the Allocate command in the user interface of the IO Allocation tool. You have to set up the complete prerequisites required for writing allocation data into CBM (see Writing Allocation into CBM) by using Bulk Data Manager. Then you can use the command **Write Allocation into CBM** of the IO Allocation tool or the command **Generate Configuration Data** in Function Designer.

De-allocating Signals

De-allocation (removing existing signal/channel connections) can be executed only in the board view. Select any object (channel, board or the root object *Boards*), click right mouse button and select command **De-allocate**. The selected signal objects will be de-allocated.

When de-allocating all signals of a signal group and using **Advanced >Write Allocation** into CBM of the object context menu in Plant Explorer there are error messages stated in a separate message box. These messages can be ignored

Users are recommended to refresh after de-allocating the signal. To refresh click **View > Refresh** or press F5.

Filtering Signals

Click **View > Filter Unallocated** to filter out the unallocated signals, for example in a large pool of allocated and unallocated signals.

Correcting Invalid IO Allocations

If there is an invalid signal/channel allocation the icons of the affected objects (signal, channel and related board) in the view are marked red. If you want to know what makes a channel invalid, just select the channel in the board view. The reason is displayed in the status bar. Possible reasons are:

- **invalid signal type -** Signal type does not fit to the channel type. For instance an AI signal is allocated to an DO channel. You can correct this only by de-allocating.
- **invalid channel number** - There is a channel number defined, which is not valid for the related board. You can correct this only by de-allocating.
- **channel is multiple allocated** - There are two or more signals allocated to the same channel. You can correct this only by de-allocating.
- **signal is multiple allocated** - One signal allocated to two or more channels. You can correct this only by de-allocating.
- **no application defined -** Only relevant on AC 800M controller. The Application property of signal is not defined. You can correct this by filling in the property (see Editing Application Values below).

Moving from Channel to Signal and Vice Versa

You can easily move from an allocated channel in the board view to the related signal in the signal view. Select the channel, click the right mouse button and select menu item **Show signal**. The selection moves to the related signal in the signal view. The inverse function moving from a signal to the channel works analogous.

Editing Application Values

Select the signals the application should be set for and perform the context menu **Edit Application**. In the subsequent dialog select the application form the drop-down box.

Clearing Views

Board and signal view of the IO Allocation view can be cleared separately or common. You access the command via menu item **View**.

- **Clear All** clears both, board and signal view.
- **Clear Boards** clears the board view.
- **Clear Signals** clears the signal view.

Refreshing Views

Data, which are displayed in board and signal view, are not updated continuously concerning changes done with other tools. Refresh the views via menu item **View > Refresh** or by pressing the **F5-button**.

Writing Allocation into CBM

User can start the function **Write Allocation into CBM** in the board view on a single channel, a board or the root object *Boards*. It works for all selected objects.

The engineered signal data as well as the allocation information needs to be written into the Control Builder M in order to update the properties of the IO boards, set the connection strings, etc.

Write Allocation into CBM command supports up to 400 channels. If the number of channels exceeds this limit, then an error message is displayed while performing this command.

However, the error message is not displayed, if the signals are allocated through Bulk Data Manager.

In order to write the signal data into the Control Builder M the following pre-requisites must be fulfilled (this is automatically guaranteed if the IO Allocation tool is used):

- the signal object must be placed in the Control Structure below a hardware unit object
- the ChannelNumber aspect of the signal object specifies a valid channel of the hardware unit
- the signal type has to fit to the channel type
- an application must be entered in the CBM_SignalParameter / CBM_PulseSignalParameter aspect

Ensure that the control module and the application have different names, before performing **Write Allocation into CBM** operation.

To start writing allocation select an object in the Control Structure, click the right mouse button and select menu item **Advanced > Write Allocation into CBM**. Depending on the selection either data of an individual signal, a complete IO board or a controller with all IO boards is written into the Control Builder M. If the project object is selected all controllers with their IO boards are written into the Control Builder M.

Do not select an object above a project object as it may drastically decrease the performance due to switching between the projects.

In an IO Allocation window **Write Allocation into CBM** command throws “Invalid Application” error if CBM_Signals are inserted on a Function Diagram, which is already allocated to an application and configuration data generated.

Also user can not force values online to these CBM_Signals even though the traffic lights in the Function Diagram shows fully allocated.

The function can also be performed within the IO Allocation tool using the context menu item **Write Allocation into CBM** on an object in the board view of the tool.

While writing data into Control Builder M a progress window is presented (see Figure 96), displaying the progress and messages (info, warnings and errors).
The progress can be stopped anytime by pressing the **Stop** button.
When finished, the Stop button turns into a **Close** button, which has to be clicked for closing the window. Click the **Save Log** button, to save all messages in a file for reuse or printing.
If **Close automatically after successful progress** is checked, the window closes automatically if no errors or warnings are displayed.

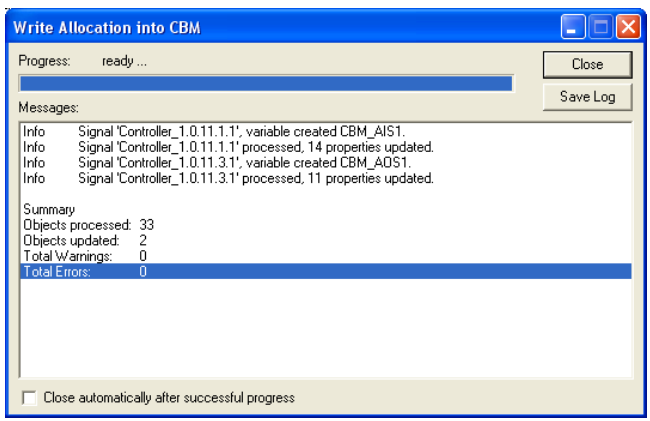


Figure 96. Write/Read Allocation Progress Message Window

- When writing data for a DP840 board to CBM there are error messages generated related to property 'Signal Range'.
  Similarly, the property "Function mode channel x" of DSDP 170 IO Boards causes error messages.
  Open the respective board in CBM and change the value of property, "Signal Range" / "Function mode channel x", save it, reset (clear the existing value and enter a new value) the property value and save again.
- While allocating the pulse signals to DSDP 160 IO Boards, user needs to manually enter the datatype in the signal information aspect.
- After allocation of multiple pulse signals to DP910*(F12 P) card and on performing **Write Allocation into CBM**, errors are generated. IO Allocation cannot be used to allocate multiple pulse signal to DP910*(F12 P) card.

Automatic Update of CBM

If this function is enabled, all changes concerning allocation data are written automatically into CBM, independent if the changes are made in IO Allocation, Engineering Workplace, Bulk Data Manager, or any other tool. Extra calling function **Write Allocation into CBM** is not longer needed.

Enable the function with menu item **Options -> Automatic Write Allocation into CBM**. A message box is launched, which confirms that Automatic Write Allocation into CBM is enabled. It also asks for running synchronization. If you confirm the question with Yes, **Write Allocation into CBM** is performed for all signal objects in the Control Structure.

Disable the function with menu item **Options -> Automatic Write Allocation into CBM**. A message box is launched, confirming that **Automatic Write Allocation into CBM** is disabled.

If the function is currently enabled or disabled is displayed at menu item **Automatic Write Allocation into CBM**, which is checked or not.

In order to update properties of CBM_SignalParameter with respect to CBM hardware properties, deselect **Automatic write allocation to CBM** and perform **Config Data Generation** or **Write Allocation to CBM** from IO allocation.

For improved performance, always disable function **Automatic Write Allocation into CBM** in IO Allocation tool before performing import operation and also before allocating IO signals through Bulk Data Manager.

Reading Allocation from CBM

The function **Read Allocation from CBM** acts the opposite way of **Write Allocation into CBM**, see Writing Allocation into CBM. This operation will overwrite your signal object data with data currently stored in Control Builder M.

If for any reason Control Builder Name and Object Name of a CBM_Signal object is out of synchronization, Read Allocation from CBM will de-allocate the CBM_Signal object without notice.

This is typically performed at the end of a commissioning phase in order to reconcile the signal object data with the data stored in the Control Builder M.

This function mainly does the following:

- update the aspects CBM_SignalParameter / CBM_PulseSignalParameter and CBM_SignalInformation
- update the channel number consumed by the signal in the aspect ChannelNumber
- optionally the signal may be re-allocated in the Control Structure.
  In case of deleting the connection string of an IO board the related signal is removed from the control structure.
  If a connection string is entered in an IO board the related signal object is created (if not already existing) and placed in the Control Structure (if not already placed there)

To start reading allocation select an object in the Control Structure, click the right mouse button and select menu item **Advanced > Read Allocation from CBM**. The scope of signal objects to be updated depends on the user's selection (see also Writing Allocation into CBM).

A progress window is launched (see Figure 96), displaying the progress status and messages (info, warnings and errors).
The progress can be stopped anytime by pressing the **Stop** button.

When finished, the Stop button turns into **Close** button, which has to clicked for closing the window. Click the **Save Log** button, to save all messages in a file for reuse or printing.
If **Close automatically after successful progress** is checked, the window closes automatically if no errors or warnings are displayed.

Messages

During Writing Allocation into CBM / Reading Allocation from CBM text messages written to a corresponding message window. They are divided into 3 categories: Info, Warnings and Errors. Follow the hints given by warning and error messages.

Property Mapping

The mapping of properties as defined in the aspect CBM_SignalParameter / CBM_PulseSignalParameter depends on the signal type (AIS, AOS, DIS, DOS, Pulse, etc.) and the type of the IO board.
The actual mapping is basically illustrated by the user interface of the aspects CBM_SignalParameter / CBM_PulseSignalParameter:
The aspect page adapts their content and layout to the actual signal type, and if allocated to a board, to the definitions given in the PropertyDefinition aspect of the IO board.

Pick lists for properties adapt to the IO board the signal is allocated to.

Configuring Property Mapping

The mapping of I/O board hardware unit properties to signal properties (to properties of the SignalParameter aspect category) is configurable.

For example AO845 which has a property “OutputFilter” which is not supported. To include support for this property:

1. Open the IOMapping aspect of the CBM_Signals object in Object Type Structure.
2. Right click on the **IOMapping** field and click **New PropertyMap** entry.
3. For **SignalProperty** enter OutputFilter.

4. For **IOProperty** enter OF % (name OF is used in the Control Properties aspect).
5. Press **Apply**.
6. Open the Categories aspect of the CBM_SignalParameter object in the Aspect System Structure.
7. Right click on the **Categories** field and click **New CategoriesProp**.
8. For **Property** enter OutputFilter.
9. Press **Apply**.

Please be aware that such additional definitions will be overwritten when you upgrade to a new version, minor version or revision of the corresponding system extensions Signal Extension for AC800M Connect and DM & PM Application.

Building Signal Object Types

IO Allocation enables a flexible signal concept. It recognizes an object as a signal if the object contains the aspects

- CBM_SignalInformation (aspect category CBM_SignalInformation)
- CBM_SignalParameter (aspect category CBM_SignalParameter)
- ChannelNumber (aspect category Control Designation).

Therefore it is possible to build own signal object types and instances by adding these aspects.

To use such own signal objects in Function Designer a Function aspect with a Function Component has to be added, for example as given in the CBM_Signal object types. For further information on Function Components see *System 800xA Engineering, Engineering Studio Function Designer (3BDS011224*).*

HART Device Support

Preparing Device Types

Use the utility HWDProcessor.exe to pre-process the HART device (object) types.

This utility is available in the Engineering Studio installation path <installdrive>:\Program Files\ABB Industrial IT\Engineer IT\Engineering Studio\Engineering Base\bin.

1. Ensure the relevant Control Project is opened and the required hardware library is inserted.
2. Double click on HWDProcessor.exe
3. Press radio button HART device.
4. Press **OK**.
5. In the structure browser shown navigate in the Object Type Structure to the HART device object type and select it.
6. Select the HART device type, for example basic HART device type AI or AO.

7. Press **OK**.
8. After processing press **Close** in the progress message window.

Needed aspects are added to the selected device type. Instances created from the pre-processed object types can be used with IO Allocation.

Working with Devices in IO Allocation

Regarding allocation / de-allocation you can work with HART devices in the same way as with CBM_Signal objects.

If the HART devices are connected to PROFIBUS the allocation information is synchronized with the FieldbusManagement aspect view.

PROFIBUS Device Support

IO Allocation supports modular PROFIBUS devices.

Preparing Device Types

Preparation of PROFIBUS Devices.

PROFIBUS specific device libraries can be installed via Device Library Wizard. User has to prepare these libraries using HWDProcessor to use them in Engineering Studio for allocating or de-allocating signals to the channels of PROFIBUS modules. Follow the steps below to prepare these libraries using HWDProcessor.

1. Ensure the relevant Control Project is opened and the required hardware library is inserted.
2. Launch HWDProcessor. *(<Engineering Studio Installed Drive>:\Program Files\ABB Industrial IT\Engineer IT\Engineering Studio\EngineeringBase\bin\HWDProcessor.exe).*

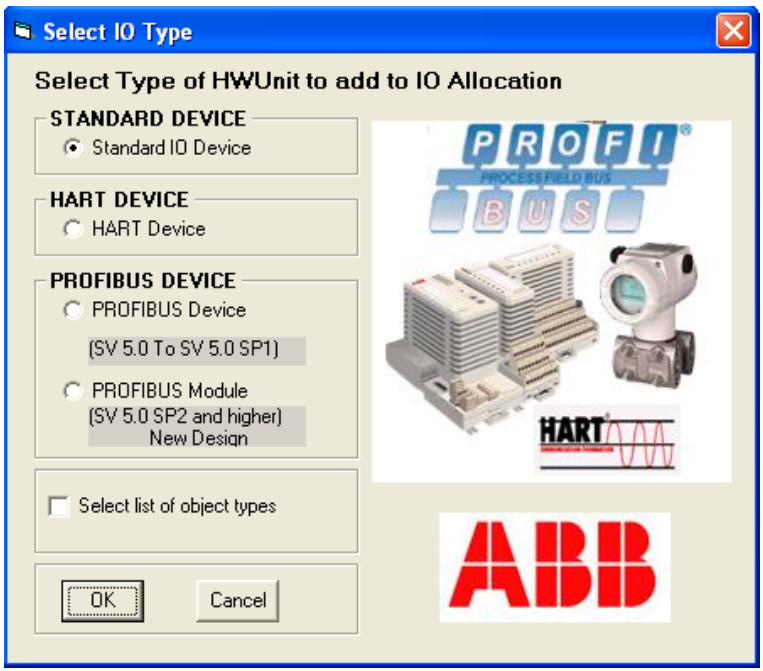


Figure 97. HWD Processor

3. Select appropriate PROFIBUS DEVICE radio button.
4. Click **OK** to open the structure browser.
5. In the opened structure browser, navigate to Object Type Structure and select required PROFIBUS device object types from the PROFIBUS Libraries.

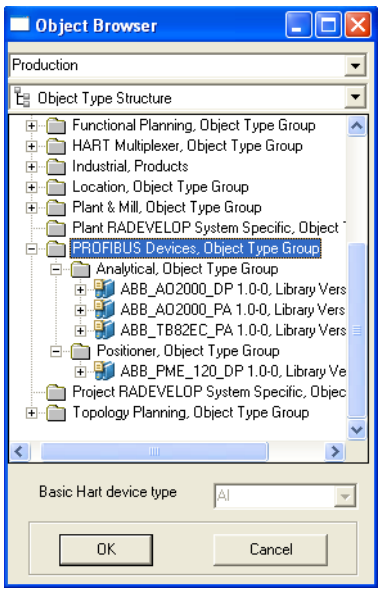


Figure 98. Sample PROFIBUS Device Library

6. Click OK.
7. After processing, click Close in the progress message window.
8. Repeat steps 3 to 6 for all other required PROFIBUS device types.

Up to 800xA 5.0 SP1, PROFIBUS devices are represented as boards in IO Allocation. 800xA 5.0 SP2 onwards PROFIBUS Modules are represented as boards.

User has to select one of the two options in HWD Processor (PROFIBUS Device or PROFIBUS Module), while preparing PROFIBUS device libraries. For compatibility reasons the option Profibus Old continues to support representing PROFIBUS devices as boards and the related way to create connection variables with the IO Allocation tool. For new projects it is recommended to use the option PROFIBUS Module to present modules of PROFIBUS devices and the corresponding way to allocate CBM_Signal objects in the IO Allocation tool.

HWDProcessor allows a user to select either PROFIBUS Device or PROFIBUS Module at any point of time. Extra care should be taken if an existing user of 800xA 5.0 SP1 is selecting PROFIBUS Module option in HWD Processor. Because selection of PROFIBUS Module option will replace all existing old design related aspects with new aspects according to the new design of PROFIBUS device support. This process can not be rolled back.

Prerequisite for PROFIBUS Module Naming

PROFIBUS devices that are imported into CBM using the Device Import Wizard may have modules or sub-modules which have names that include AI, DI, DP, AO, DO. These are keywords for IO allocation tool to detect and classify the type of signals that are to be allocated to the modules/sub-modules.

For example if the PROFIBUS module includes the string DP as part of its name, IO allocation tool expects its channels to be of Pulse data type. The allocation of a non Pulse type signal to these channels is not allowed and the following error message is displayed: *Please insert the signal under compatible board.*

Refer to Table 17 for the string and the type of CBM signal that is expected by IO Allocation tool:

Table 17. String and type of CBM Signal

String	CBM Signal
AI	CBM_AIS
AO	CBM_AOS
DI	CBM_DIS
DO	CBM_DOS
DP	CBM_PulseS

For modules with the above keywords, allocation of required signals to the channels can be done by changing the module names in the Device Import Wizard. An example is explained below:

1. Browse and select the .gsd file in Control Builder to start the Device Import Wizard.
2. Check the Device Information and click **Next**, refer to Figure 99.

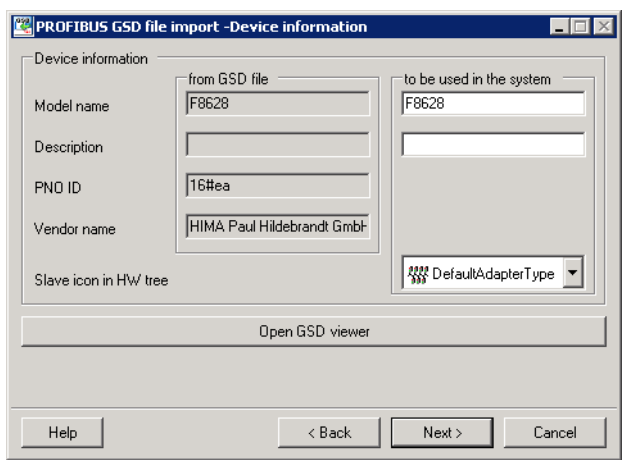


Figure 99. Device Information

3. Select the modules in the left pane. The default name of a module is updated under **Original name from GSD**. The user can change the default name in the **Name to be used in the system** field.

   In this example the module name includes the keyword ***DP***, refer to Figure 100.

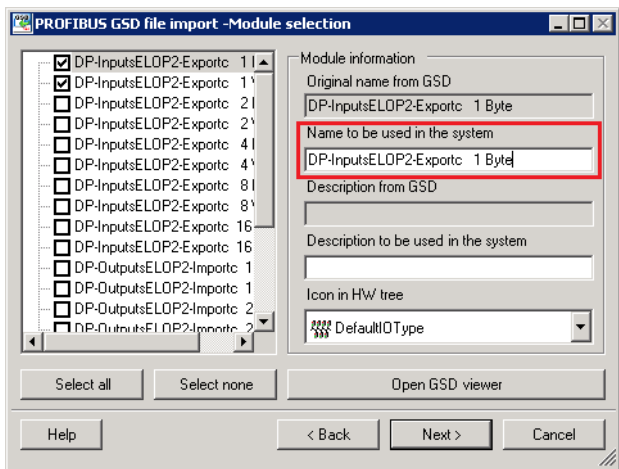


Figure 100. Module Selection

4. Delete the keyword or replace it with an alternate name.

In this example the keyword DP has been deleted, refer to Figure 101.

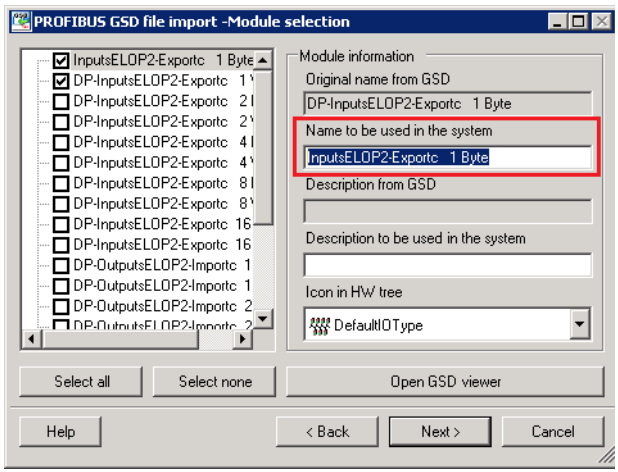


Figure 101. Deletion of Keyword

5. Continue with the import workflow which creates HWD files with modified module names.

For more information on Device Import Wizard refer to, *System 800xA Control AC 800M Configuration (3BSE035980*) and AC 800M PROFIBUS DP Configuration (3BDS009030*).*

IO Allocation

After creating signals (see Creation of Signals)and PROFIBUS modules (see Creation of Boards), the signals need to be allocated to the PROFIBUS modules using IO Allocation tool.

The channels of PROFIBUS IO boards start from '0' in the Hardware Editor of Control Builder. But in IO Allocation tool, the channels of PROFIBUS IO boards start from '1'. If the user intents to allocate a PROFIBUS signal to channel number ‘x’ in Control Builder, then always allocate the signal to channel number ‘x+1’ in IO Allocation tool, where x = 0,1,2,…n.

If no tagnames are entered into the Name aspects of the PROFIBUS device and module the corresponding information fields in the Allocation column of the right pane and in the left pane of IO Allocation view show the given device number and the module number.

Be aware that they just repeat the last two fields of the absolute reference designation.

Regarding allocation / de-allocation of CBM_Signals and reading allocation data from Control Builder M / writing allocation data to Control Builder M, user can work with PROFIBUS modules in the same way as with Standard IO board objects.

Feature Pack Functionality

PROFINET Device Support

Preparation of PROFINET IO Devices

Libraries are prepared using **HWDProcessor,** to use them in IO allocation tool as well as in Function Diagram for allocating or de-allocating signals to the channels of PROFINET modules.

PROFINET specific device libraries cannot be installed through Device Library Wizard.

To prepare these libraries using HWDProcessor.

1. Ensure the relevant Control Project is opened and the required hardware library is inserted.

2. Launch HWDProcessor. (<*Engineering Studio Installed Drive*>:*Program Files\ABB Industrial IT\Engineer IT\EngineeringStudio\EngineeringBase\bin*).

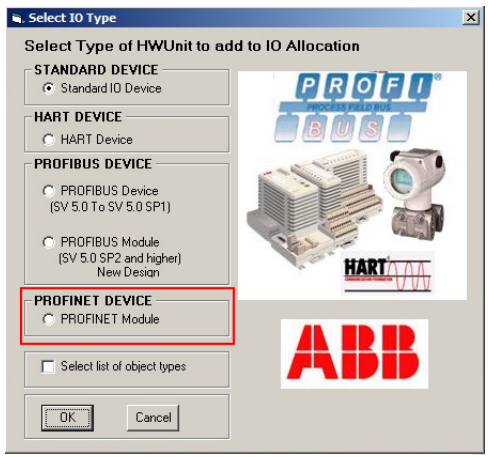


Figure 102. HWD Processor

3. Select PROFINET Module and click OK.

   The Object Browser window appears.

4. In the Object Browser, navigate to the Object Type Structure and select the required PROFINET device object types from the PROFINET Libraries.

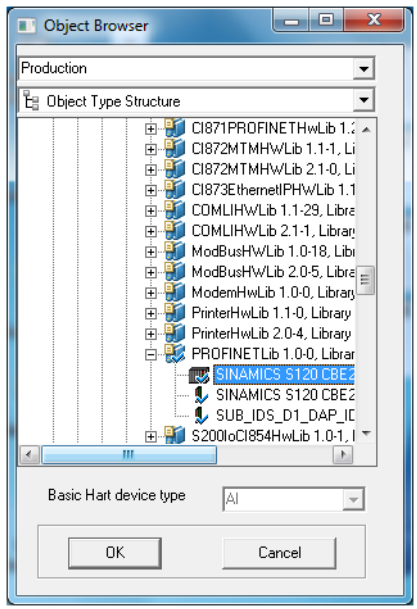


Figure 103. Object Browser

5. Click OK.
6. After processing, click Close in the progress message window.

IO Allocation

After creating signals (see Creation of Signals on page 181) and PROFINET modules (see Creation of Boards on page 184), the signals need to be allocated to the PROFINET modules using IO Allocation tool.

Section 6 Parameter Manager

This section describes the Parameter Manager, with its functions, commands and utilities. For more information, refer to the Release Notes delivered with the product.

Parameter Manager is a component of Engineering Workplace.

The Parameter Manager is the tool for sharing data within an IIT system. Data is stored and maintained at a single place by using the Parameter Aspects. The data can be used in different applications through the referencing mechanism.

Parameter Manager Functions

Parameter Manager comprises:

- Creation of user defined parameter aspect categories of several kinds:
  - Table-like category.
  - Structured category.
  - Extendable category.
- Entering, modifying, and storing dedicated engineering data into properties of parameter aspect instances of these categories.
- Providing the data of the properties for data sharing.

Through property (parameter) references the property values can be used as parameters in aspects of other aspect systems. Property references can be made to properties of parameter aspects in the same or in other Aspect Objects.

For example, common project parameters can be stored in a parameter aspect of the Aspect Object representing a project. A property reference in a Microsoft Word document of a document aspect and a property reference in a control properties aspect of a Control Module or Function Block Aspect Object of Control Builder can refer to the same parameter aspect property and can be updated with the actual value

of the property on request by the user. Data sharing between different aspect systems can be organized in this manner.

Parameter Manager stores the parameter aspect property data and the parameter aspect category definitions in an aspect blob.

Before You Start

Recommended User Group Settings

Concerning Parameter Manager activities three user profiles have been identified:

- Setup users who execute the Parameter Manager product setups must be included in the Windows Administrator group.
- Configuration users who create and modify Parameter Aspect Categories must be in the Windows IndusrialITAdmin group.
- Users who create Object Types together with Parameter aspects and users who instantiate object types and who add, copy, move, delete Parameter aspects must be included in at least one of the following Windows User Groups:
  - IndustrialITUser and IndustrialITApplication or
  - IndustrialITAdmin which includes additionally the rights for creation of aspect categories.

System / Project Creation

After creation of a new system or project the system extension '**DM & PM Application'** must be loaded.

Do not change system name (aspect Name of Admin Structure.Administrative Objects.domains.<system name>) after system creation. Such a modification will impact the functionality of Document Manager and Parameter Manager. If the user has already changed the system name, undo this modification and stop and re-start the system in Configuration Wizard.

Project / System and Directory Structures

Parameter Manager has two areas where it stores files on your PC. The product files which are stored in the product directory structure and the system/ project files which are stored depending on your definition at system creation when you define, where the project data are stored.

Product Directory Structure

<anydrive:>\Program Files\ABB Industrial IT\Engineer IT\
Engineering Studio\Engineering Platform\
DocumentParameter Manager**add**
Parameter Manager aspect configuration files

<any drive:>\Program Files\ABB Industrial IT\Engineer IT\
Engineering Studio\Engineering Platform\
DocumentParameter Manager**bin**
installed Parameter Manager product files

<any drive:>\Program Files\ABB Industrial IT\Engineer IT\
Engineering Studio\Engineering Platform\
DocumentParameter Manager**import**
Parameter Manager afw-files

On system/ project creation, the Configuration Wizard presents a dialog to enter the directory path where user stores the **Server Data, System Data** and **Workplace Data** specific data.

Product User Interface

The Parameter Manager displays its data in two ways.

- in the **Preview Area** of Plant Explorer where users gets a table like data presentation for configuration of the category data and for instance data
- additionally, in connection with the **Open** command on Parameter aspects - **Aspectview Area** - user gets in an Excel like bulk data sheet the instance data of the selected Parameter. If user clicks **Show All,** all Parameter aspects (of that selected category) included in the start object (all the children) of the current Structure is displayed. The following Figure 104 shows an example for

Parameter presentation in the **Preview Area**.

User Interface and Description of Buttons

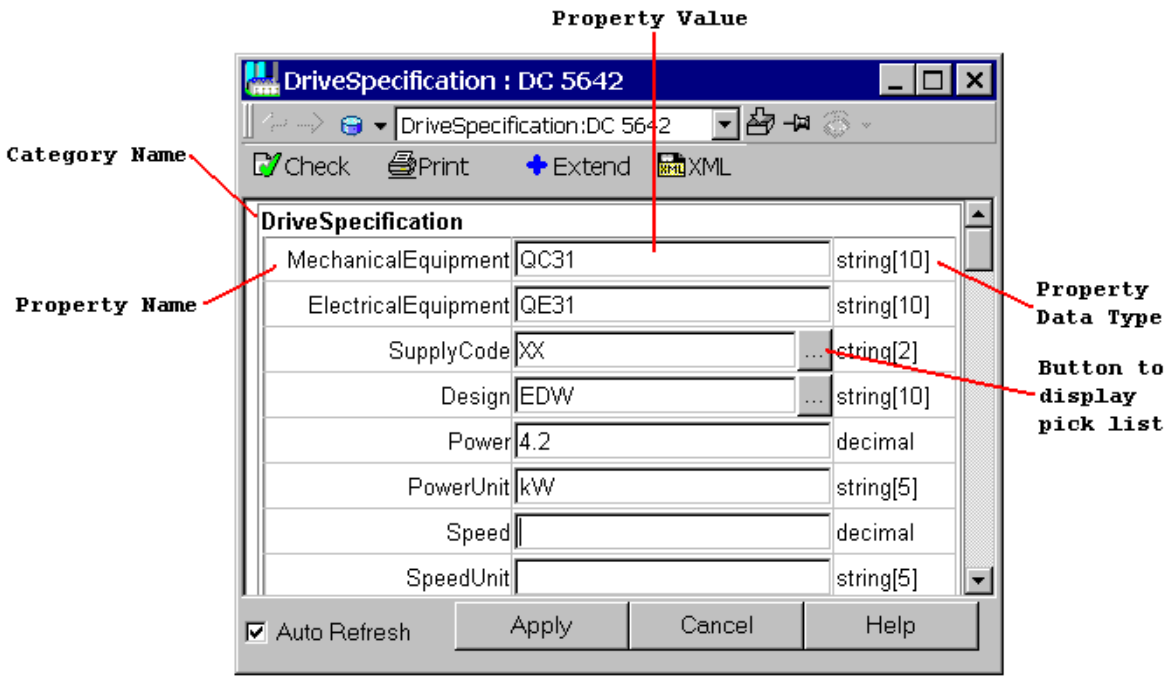


Figure 104. User Interface Overview

The following table describes the function of the different buttons.

Table 18. Functions

Name	Function
Check	Checks if the entered data is valid. This check also determines that the value of each property corresponds to the data type of that property.
Print	Prints the table.
Extend	Adds a new instance specific property to the aspect. This command is enabled for Extendable Categories.
XML	Opens a context menu that permits the user to access an XML representation of the aspects data.
XML>Copy XML	Copies the XML representation of the aspects data into the clipboard.
XML>Paste XML	Replaces the aspects data with data from the clipboard. The command expects valid XML data (as text) in the clipboard.
XML>Copy Schema	Copies the XML schema definition of the aspects data into the clipboard.
Auto Refresh	If selected, the user interface will automatically be updated in case another user changes the aspect data.
Apply	Writes the data into the aspect. The data will automatically be validated before writing it to the aspect (see Check function).
Cancel	Discards your modifications and reloads the user interface with data from the aspect.
Help	Displays the help text.

User Interface and Description of Buttons

The user interface of the Parameter Manager supports a context menu. Elements that have a context menu are highlighted if user moves the cursor to such an element. Figure 105 shows an example.

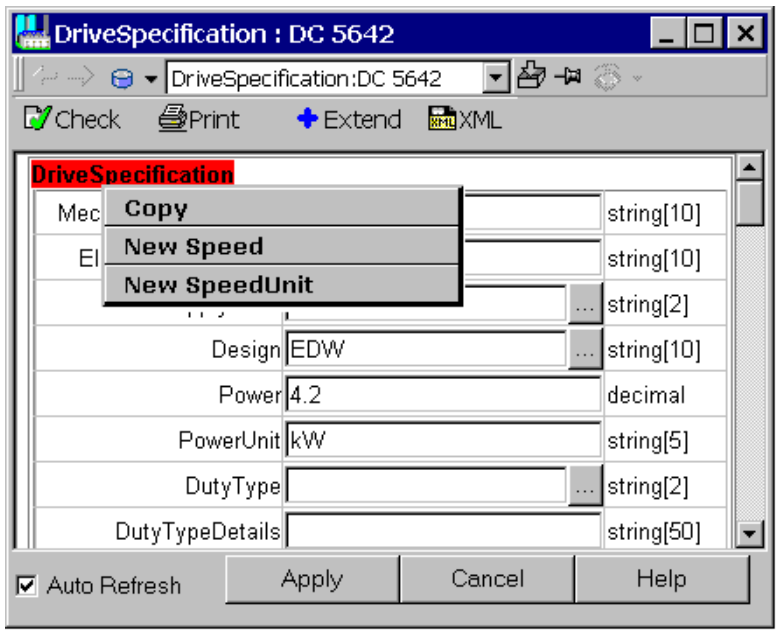


Figure 105. Context Menu

The following table lists all possible entries in the context menu an their function.

Table 19. Functions

Menu Entry	Function
Cut	Copies the data of the element into the clipboard and removes the element from the table. Note: the element is stored as XML in the clipboard.
Copy	Copies the data of the element into the clipboard.
Paste	Insert the element from the clipboard into the table.
Delete	Removes the element from the table.

Table 19. Functions (Continued)

Menu Entry	Function
New NNN	Create a new element (NNN) in the table. The context menu will include all properties that can be created for the selected element in the user interface.
New Instance Property	The menu entry is only available for Extendable Categories. It will add a new instance specific property to the aspect.
Edit Instance Property	The menu entry is only available for Extendable Categories. It allows you to edit a instance specific property.

User Interface with Sub Properties

Properties of a Parameter Manager aspect can have sub properties, or, if user thinks in terms of a relational database, the property is a key to a row in a different table (see Creation of a Structured Category). The user interface of the Parameter Manager displays this sub properties as well.

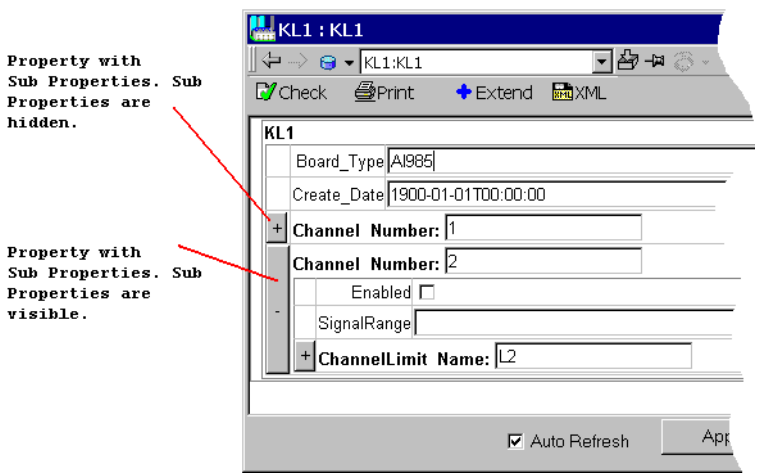


Figure 106. Display of Sub Properties

User can display or hide the sub properties by clicking the button to the left of the property that contains the sub properties.

Read-Only Mode

If the viewed parameter aspect is inherited, or the user does not have the privileges to alter the data of the parameter aspect, the user interface will not allow the user to save any changes to the aspect. The **Apply** button is always disabled.

Customizing of the User Interface

The user interface for the Parameter Manager can be adopted to present the data of the individual parameter category in a different style. The parameter categories in the system will display the data in a different layout than what has been described in this manual. Figure 107 shows an example.

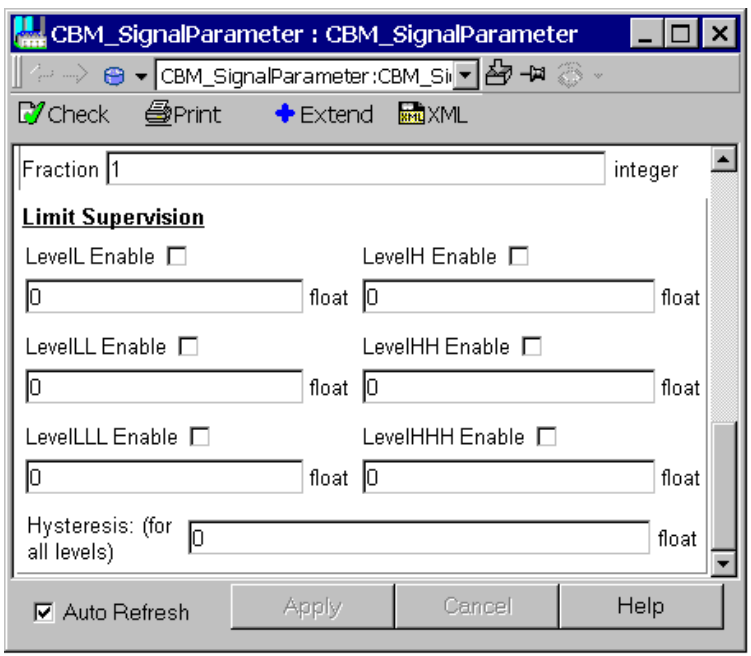


Figure 107. Customized View

Parameter Aspect Category Configuration

The Different Kinds of Parameter Aspect Categories

User can configure different kinds of Parameter Aspect Categories:

- **Table-like Categories**: All Parameter Categories having only simple properties will be called Table-like Categories (see Appendix C Example 1: Table-like Categories)
- **Structured Categories**: All Parameter Categories which have one or more subcategories as structured properties will be called Structured Categories. (see Appendix C Example 2: Structured Categories)

Version enabled systems do not support subcategories.

- **Extendable Categories**: Parameter Categories created without any property but having the possibility to add instance specific properties will be called Extendable Categories. (see Appendix C Example 3: Extendable Categories)

It is possible to mix these kinds. Parameter Categories having instance properties as well as structured properties will be called **Customized Categories**.

A Parameter Category can be defined as **System Category.** Once defined System Categories and their properties are protected and can not be changed. For example Categories needed for internal use such as ‘Categories’ and ‘GeneralProperties’ are System Categories. They must not be changed by the user.

Predefined Parameter Aspect Categories

Predefined Parameter aspect categories of Parameter Manager are:

- ArticleData
- DriveSpecification
- TagData

These categories are prepared for special engineering templates (Excel documents) which are based on the Parameter instances of those categories (see folder **Engineering Templates**).

Configuration of Parameter Aspect Categories

Parameter Category Configuration is done within the Plant Explorer. Parameter categories are displayed as objects in the *Aspect System Structure*. Create, copy, rename, and delete operations can be performed on these objects. Copying these objects with the Import / Export tool is also possible.

Parameter Category objects are placed below the Parameter Manager Type Object.

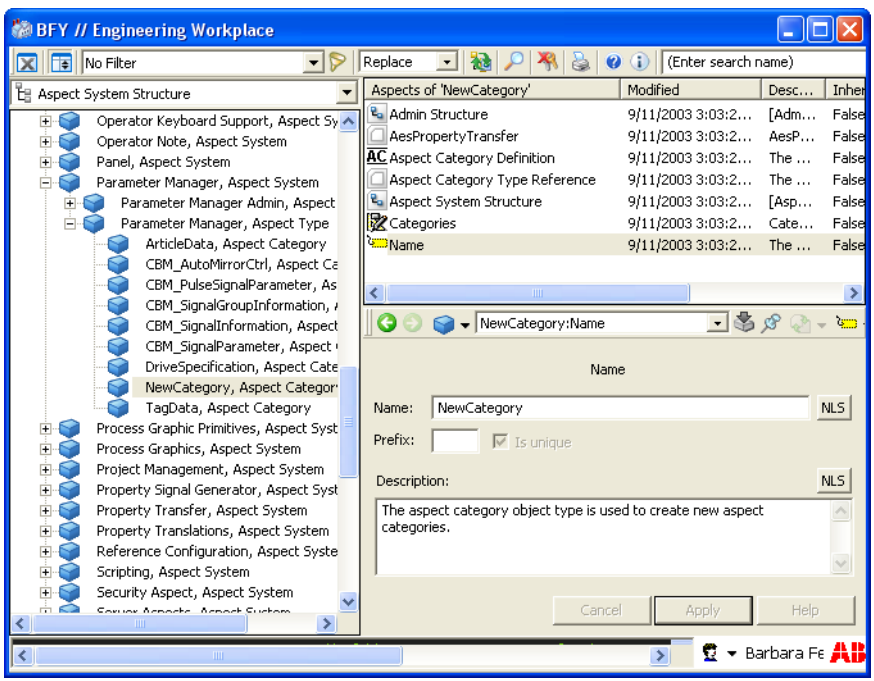


Figure 108. Objects of Parameter Manger in Aspect System Structure

General Remarks on Parameter Category Configuration

Configuration of properties for a Parameter category is performed using the Main view on a Categories aspect. The tool presenting this view is named Property Editor (PED).

For creation/modification of categories the user must belong to the user group IndustrialITAdmin. If not you will get a message “permission denied on object...”

Features and Restrictions for Creating User-defined Aspect Categories

The **Name** of a category must not contain blanks or special characters and must be unique.

Subcategory of: This item defines a category as sub category of another category. See Structured Categories in The Different Kinds of Parameter Aspect Categories.

Instance Properties: This flag must be set to indicate that this category has the possibility to add instance specific properties. See Extendable Categories in The Different Kinds of Parameter Aspect Categories.

Add a Parameter Aspect Category

Creation of Table-like Category

To create a new Parameter Aspect Category follow the steps:

1. Select the **Parameter** object in the Aspect System Structure (see Figure 109), right-click the mouse and select **New Object...**.
2. Select object type **Aspect Category**, enter a name and click **Create**.

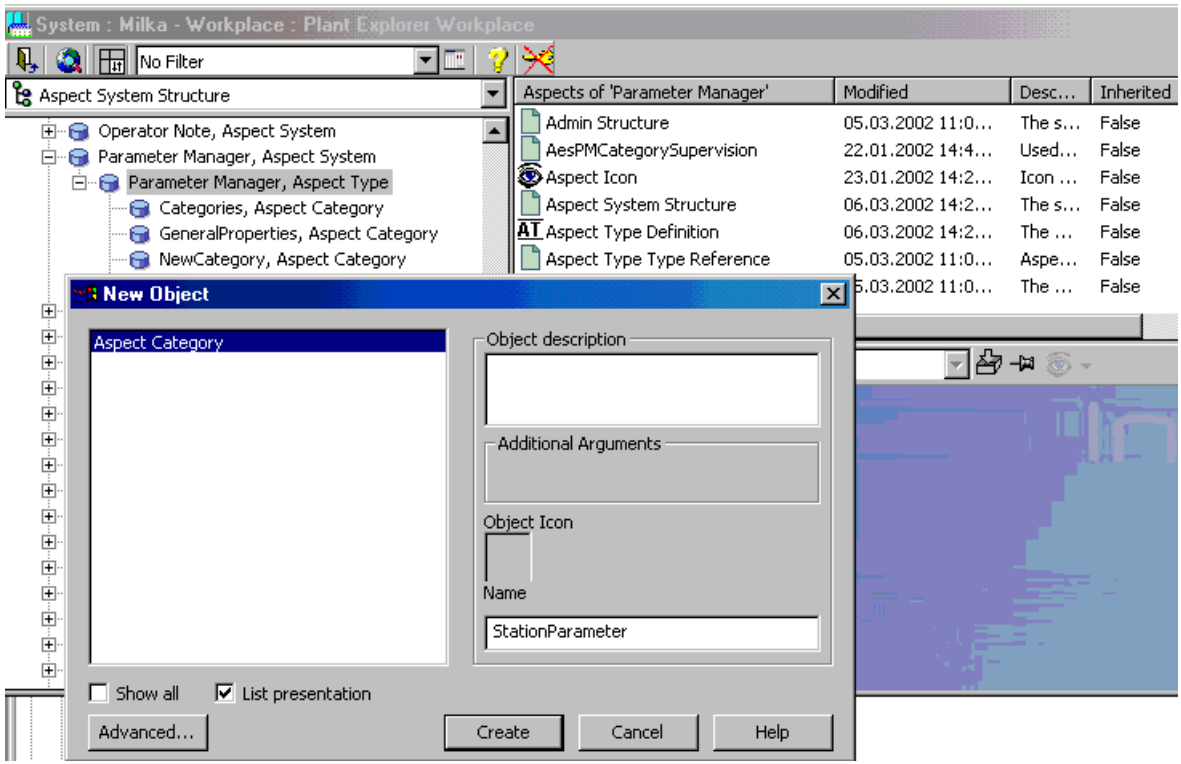


Figure 109. Create a New Parameter Category Object

A new parameter category is created which has no attributes but is usable.

For details on how to add properties, see Add Properties to a newly created Parameter Category.

The name of an Aspect Category must not contain blanks or special characters. The name must be unique

Creation of an Extendable Category

Follow the steps to create a category with instance properties:

1. Select **Parameter Manager, Aspect Type** object in the Aspect System Structure, right-click the mouse and select **New Object...**.
2. Enter a name and click **Create**. A new parameter category is created (see also Figure 109)
3. Select the **Categories Aspect** of this Parameter Category to show the **Main View** in the **Preview Area**
4. Select the **Instance Property** check box.

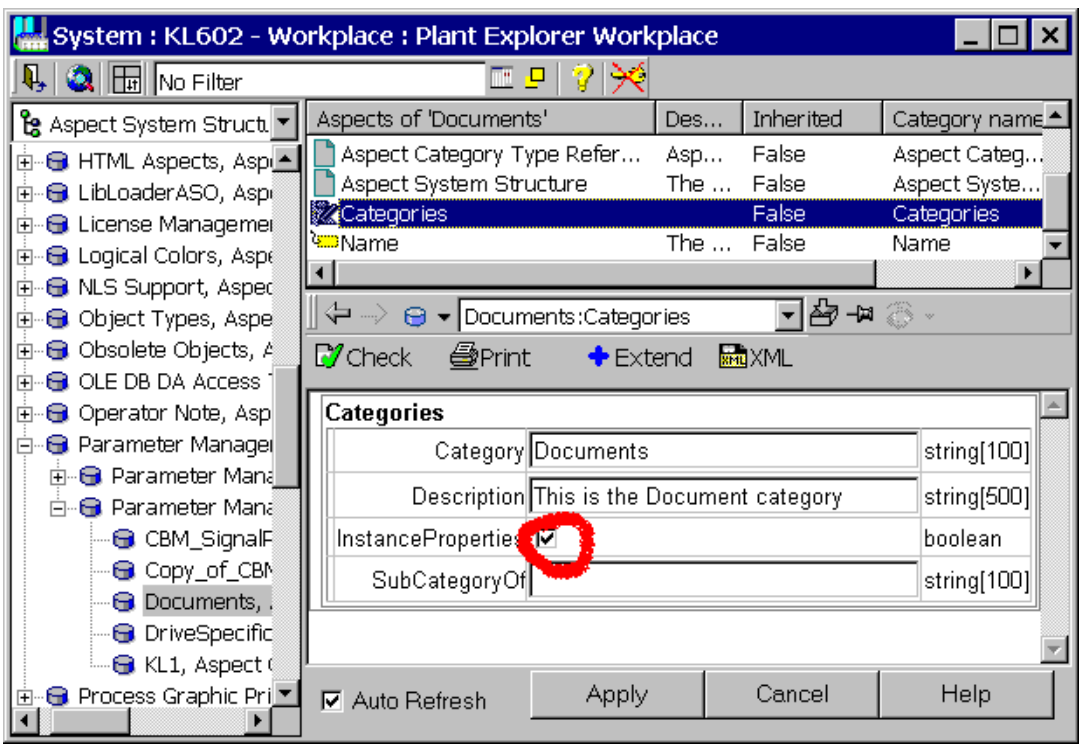


Figure 110. Extendable Category

An extendable category has been created. Refer to Add Instance Properties to a Parameter Aspecthandle aspect instances of an extendable category and to define the instance specific properties.

Creation of a Structured Category

Follow the steps to create a structured category:

1. Create a **Parameter Aspect Category** object in the Aspect System Structure as described in Creation of Table-like Category, which will be the top category (see below “Example for a Structured Category”).
2. Enter the **Aspect Area**, right-click the mouse, **New Aspect** dialog comes up. In aspect categories list search for **Categories.** The **Categories** aspect has the **AspectDescription** = Category Definition (Properties) for Parameter Manager aspects. Use this to create Parameter Manager subcategories.
3. Enter a name for this sub category and click **Create**.
4. Select the just created category aspect, the **Preview Area** offers beside others the property **SubCategoryOf**, here enter the name of the ahead created top category and press Apply button.
5. An additional property is called **Category** where “MyCategory” as value is inserted. Here insert the sub category name too (see also, “Example for Structured Category”Example 2: Structured Categories Figure 326).
6. Proceed in that way for more sub categories.
7. How to add properties to the just created categories see Add Properties to a newly created Parameter Category.

An Example of a structured category and how to create it is shown in Appendix Example 2: Structured Categories.

Creation of Parameter categories with sub categories where the name of any property of the main category matches the name of any sub category results in an error. The naming convention within the Category Object must be unique.

User has to completely configure a subcategory before creating any subcategories within the first subcategory.

Add Properties to a newly created Parameter Category

To define your required parameter aspect category properties follow the steps:

1. Select the **Categories** aspect of this parameter category, the **Main View** will be presented in the **Preview Area.**
2. Open the Context menu of the top element (the Categories element).

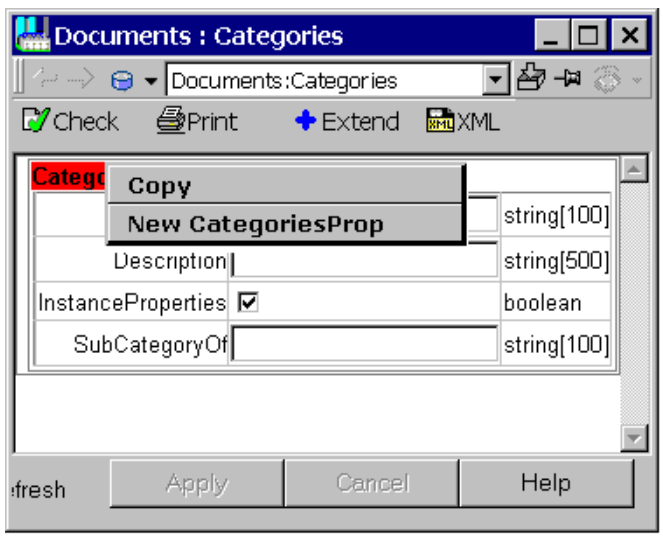


Figure 111. Categories Main View

3. Select **New CategoriesProp**.

The system will now add the new property to the display.

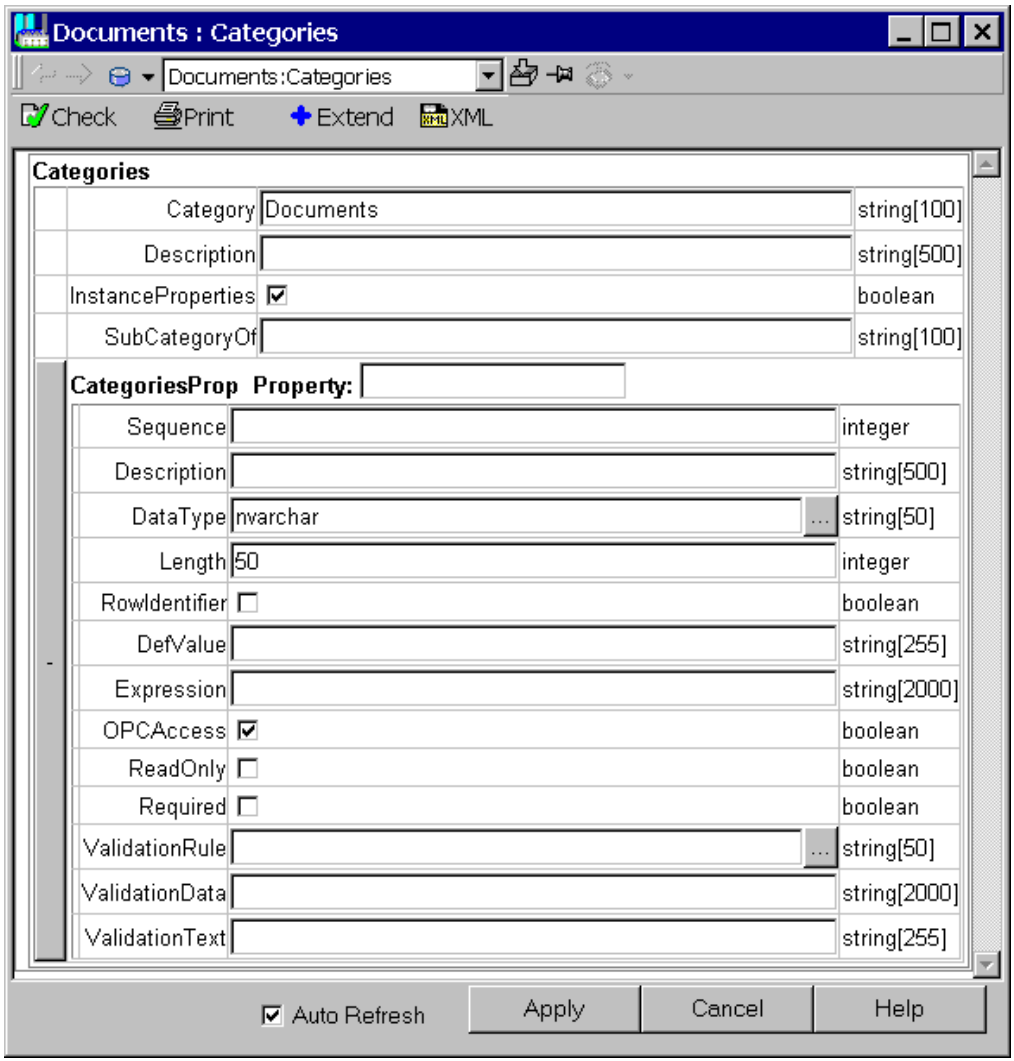


Figure 112. Add Property

Fill in the data for the new property and repeat this process until all properties have been added.

Features and Restrictions of Category-Properties

During definition of the properties of a category the following characteristics of properties could be set (see Figure 112)

Name: The name of a property must not contain blanks or special characters and must be unique for one category.

Do not use numbers in the first position of property names.

Do not name your property ‘System’ or ‘FilePath’.

If a property is named as ‘System’ or ‘Filepath’, exit the Property Editor, delete the just created Parameter category, and re-introduce it with corrected property names.

Description: A descriptive text for this property.

Sequence: The entered number defines the order of the presented properties in the property editor.

It is recommended to use sub property Sequence to define the creation order of properties. In cases where you build up expressions with properties of that category define the sequence order number in a way that a property which will be used in an expression is already created.

DataType and Length: The supported data types are shown in Table 20. The Length must only be given for ‘char’-types. The default will be ‘nvarchar(50)’

Default Value: A value which will be automatically inserted for this property at creation of a parameter aspect instance.
For entering Default Values you have to consider: During configuration it will not be checked, if the Default Value is correct due to Datatype and Validation Rules. Be sure to give a correct Default Value.

If you use default values, do not use in parallel an expression. If you need default values and expressions for your solution, create expressions in your Parameter Category and use Object Types where you add a Parameter aspect of that Category and fill in your needed “default” values.

Expression: The property-value will be calculated according to a given expression; a property with given expression will be set to read-only.

If you want to define an expression to fetch data from any aspect in your project use the Absolute Reference type to define your reference string. Example Absolute Reference to any object like: ((Name))Motor:ArticleData:Category.

Expressions are not supported in Aspect Type Categories. The Category Definition for Parameter Manager aspects are used for creating Parameter Manager sub-categories.

ValidationRule: A rule to validate data of this property. The rules implemented are: GREATERTHAN, LESSTHAN, BETWEEN or NOTBETWEEN.

ValidationData: If a ValidationRule requires data, have to be entered here. Example:

ValidationRule: LESSTHAN
ValidationData: 300

ValidationRule: BETWEEN
ValidationData: 10;100

Required: Set this flag, if you want this property must have a value anyway. Then a default value must be given.

Read Only: Set this flag for a read-only property. Not yet supported.

OPCAccess: If checked, property can be accessed via OPC.

Rowidentifier: Set this flag to indicate, that this property will be part of the primary key. Then a default value must be given. This flag can only be set, if the category is a subcategory of an other category.

LOV Value: Supported for all date types except integer and bit. Enter value and description as much you need.

Description for List of Value Data is not presented in PED.

Use of List of Values in connection with LimitToLOV flag: When you define for your property of any category a list of values and you want to force that only values

from that list should be used to fill in, then you have to check the LmitToLOV flag.If this flag is checked only values define in LOV list will be accepted.

If the LmitToLOV flag is set, take care that the DefValue property (default value, to be inserted into the corresponding property at create of aspect) holds a value out of the List of Values, otherwise aspect create will fail.

Table 20. Data Types

Data Type	Description
nchar	Fixed-length Unicode data with a maximum length of 4,000 characters
nvarchar	Variable-length Unicode data with a maximum length of 4,000 characters
bit	Integer data with either a 1 or 0 value (True/False)
smallint	Integer data from -2^15 (-32,768) through 2^15 - 1 (32,767). Storage size is 2 bytes.
integer	Integer (whole number) data from -2^31 (-2,147,483,648) through 2^31 - 1 (2,147,483,647). Storage size is 4 bytes
float	Floating precision number data from -1.79E + 308 through 1.79E + 308.
decimal	Fixed precision and scale numbers. When maximum precision is used, valid values are from - 10^38 +1 through 10^38 - 1
datetime	Date and time data from January 1, 1753, through December 31, 9999, with an accuracy of three-hundredths of a second, or 3.33 milliseconds.

List of Values for Property of data type Decimal are not supported.

Use instead data type float or integer to add list of values.

The validation rule BETWEEN on properties of data types decimal is not supported.

Use data type integer or float instead of decimal.

The validation rule NOTBETWEEN on properties of data types like integer, float, and decimal is not supported.

Presentation of additional description in Property Editor (PED) data view:

On right side of the PED aspect data presentation, next to the data type, an area is prepared to present for example the data unit (see below, hPa or m/s).

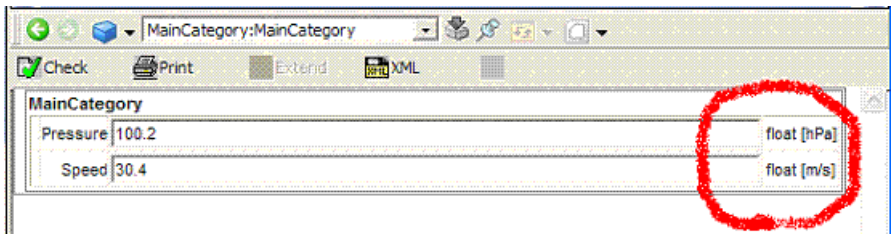


Figure 113. Presentation of a Data Unit

To use this functionality, enter in the Aspect System Structure; select your Category Aspect Object, select aspect Categories. PED presents in pre-view window the properties of your category. Open the property which should get a Unit text. Enter into the Description property on any place the "["and "]". The text you include between the brackets and the brackets will be presented. Additional text stored in the Description property will be omitted (see below, property Pressure, entered text [hPa]).

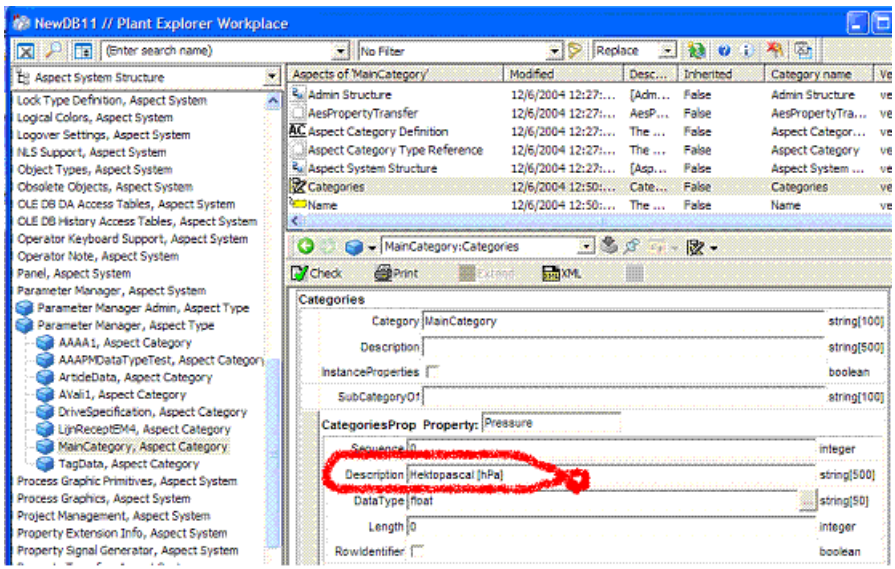


Figure 114. Configure Data Unit

Scaling/rounding in Property Editor (PED) data view for data type decimal:

Scaling is a flag for properties of data type decimal where Scaling defines the number of digits right of the decimal point which will be accepted at the time when you enter the data. Note, if you define Scaling 2 it means you allow entering numbers only with two digits right of the decimal point, like: 234.66, 2.01. If you enter 234.98702 it will be rejected.

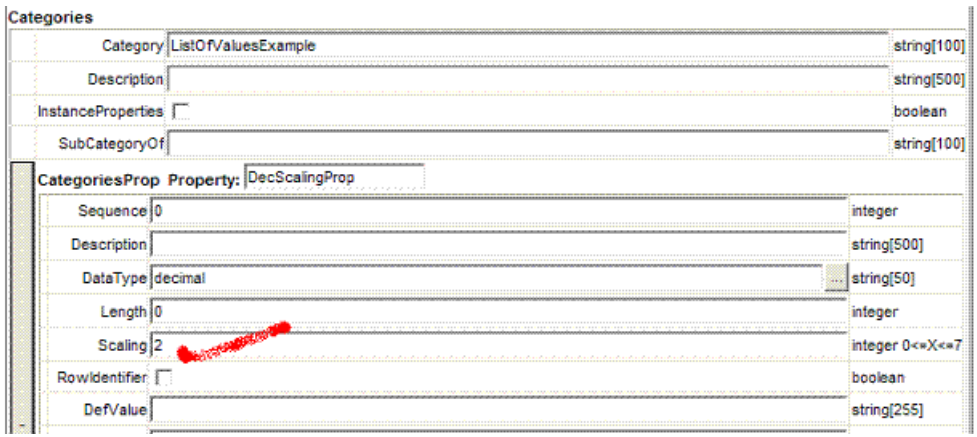


Figure 115. Configure Scaling

Additionally to the Scaling flag there is a keyword to round the value of a data type decimal. If you enter into the Description property the keyword %p3 the entered number will be rounded to show only 3 digits right of the decimal point. Valid precisions are: 0... 6, %p0...%p6. Note that the use of scaling and %pn has to be coordinated; to get the full functionality, scaling value has of course to be greater than round value, otherwise scaling prevent entering values to round. For example, you define scaling to 7 and %p3; it means you will be able to enter values which has max 7 digits right of the decimal point and at Apply the values will be rounded to 3 digits right of the decimal point. But keep in mind the entered value will be stored in the data repository, the rounded value will only presented in your view.

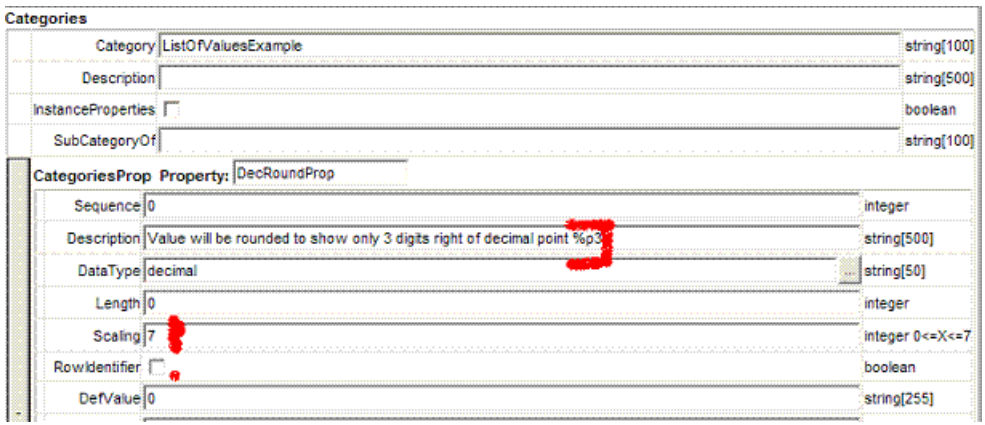


Figure 116. Configure Rounding

Delete a Parameter Aspect Category

Select the parameter category object to delete in the Aspect System Structure, click the right mouse button and select operation **Delete**.

The parameter category is deleted in the Aspect System Structure.

Delete operation succeeds only if **no** instances of the category exist.

Deletion of the whole category object in Aspect System Structure works only without instances of that category. An error message at delete will inform you how many instances are still in your system. You first have to delete all the instances before delete of category object will be executed.

You cannot delete a Parameter Aspect Category which is defined as a System Category

Modify a Parameter Aspect Category

You can add, change or delete properties of a Parameter Aspect Category using the **Properties Configuration** view of the Categories Aspect in the **Preview Area** (see Figure 112)

If the category has a property which is set to read-only or a expression is defined the required flag must not be set.

If instances exist for this aspect category regard the following rules:

- **Changing property definition** is in principle only possible if the category has no instances. This means:
  - If no values are filled in for a property, changing of Datatype will be ok.
  - If category has instances RowID must not be changed
  - If property has an expression remove the expression to change the definitions.
  - If validation rules are used for properties with for example data type float, like BETWEEN, LESSEQUAL, GREATEREQUAL, etc., it is possible to change the property name even if aspect instances of that category already exists.
  - If the property data type is changed from number data types (e.g. float, integer,...), or bit to string types like nvarchar, this will always be possible (Length size) even if aspects of that category already exists. The database transforms on its one into the new datatype (sometimes with truncation) or the change will be rejected and you will be informed by error message.
- **Change of Property Data Type:** It is possible to change the property data type even if aspects of that category already exists. To change from number data types (float, integer,...), or bit to string types like nvarchar is always possible. The database transforms the new data type (sometimes with truncation) on its own or rejects the change and informs you by error message.

If you have already created parameter aspects with properties of data type string, for example nvarchar, and no value is inserted (empty), and you change that property from string to decimal or float, then such a modification will be rejected. First enter any value into the string property and then modify the property to data type decimal or float.

- **Deletion of a property** is possible, existing data in this column will be lost. But if the property is a ‘system property’ or the category is a ‘system category’, deletion is not allowed.
- **Addition of a property** can always be done, you can even add a new property to a ‘system category’.
- **Deletion of a sub category**: It is possible to delete a sub category even if instances of that category object already exists. After delete of the child category, the already instantiated aspects are still there but without the child category instance data.

Copy a Parameter Aspect Category

Select the Parameter Category Object to copy in the Aspect System Structure, click the right mouse button and select operation **Copy**.

Select the **Parameter** Object, click the right mouse button and select **Paste**.

A new parameter is created having the same attributes as the source parameter. If the parameter name conflicts with existing parameter names (they have to be unique in a system), a new name is generated automatically, based on the source parameters name (Copy_of_<name>). Copy parameter category can be performed only inside one system. Between different systems copying of parameter has to be done by using the **Import / Export** tool

Copy of Parameter Aspect categories **which hold sub categories** is not supported.

Rename a Parameter Aspect Category

Select a Parameter Category Object in the Aspect System Structure and click on its **Name** Aspect.

Modify the name in the preview area.

This function is useful, if automatic name generation was done during a copy operation.

Rename of Parameter Aspect categories **which hold sub categories** is not supported.

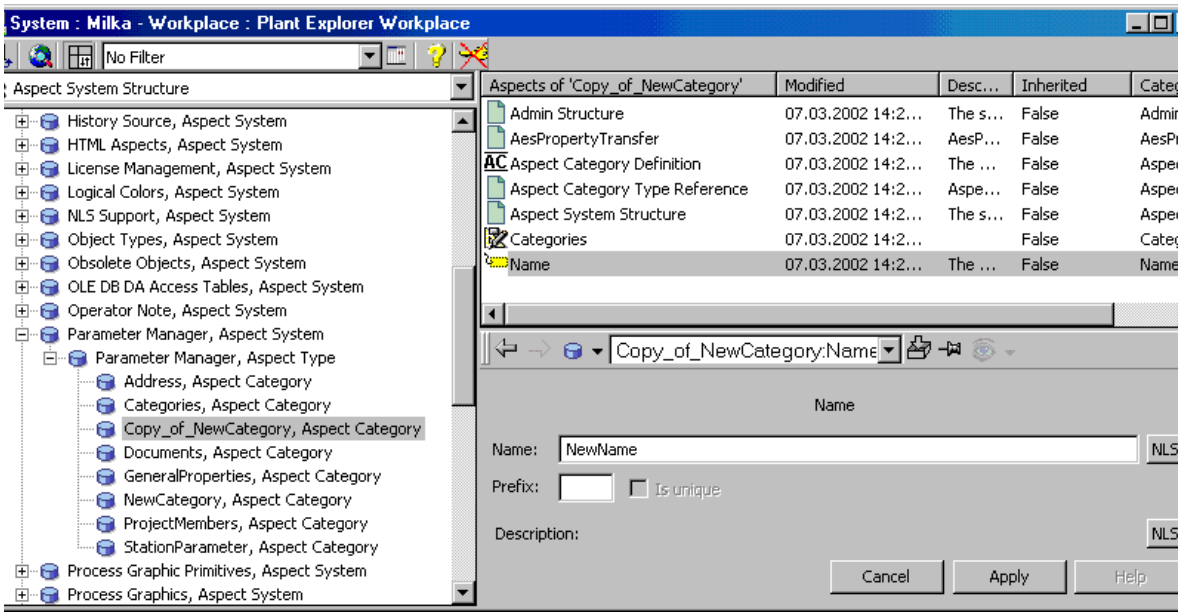


Figure 117. Rename a Parameter Aspect Category

It is not possible to rename a Parameter Aspect Category which is defined as a ‘system category’

Working with Parameter Manager

The following section shows how to work with Parameter Manager.

Viewing and Modifying Parameter Aspects

Parameter Aspects can be presented and modified basically in two ways:

- In the Property Editor

  Within the Property Editor one aspect at a time is presented. The property editor is flexible in respect to viewing and editing any Parameter Category defined by a user. The user interface provides support in terms of data type checks, limit checks, pick-lists, default values – provided this information is defined for the related Parameter Category. Data presented in the Property Editor can also be printed in the same layout
  The Property Editor is described in more details in Add a New Parameter Aspect and Add Instance Properties to a Parameter Aspect.

When saving aspect property data in Property Editor the changes are cached, they do not trigger Script Manager scripts subscribing for the changes immediately. You can use Data Sheet to edit and save changes direct to data repository

- In a Data Sheet

  Any Parameter Aspect can alternatively be opened in a Data Sheet view which is based on the Bulk Data Manager. When an aspect is opened in the Data Sheet view only the aspect selected is presented. However the user is free to apply filters in order to retrieve for example all aspects of a certain category within the sub tree of the starting object. The Data Sheet (Bulk Data Manager) is automatically configured to show the properties defined for the related category. Within the Data Sheet view the user can create additional objects having a Parameter Aspect, store them in the structures like the Functional Structure, delete objects, modify data and store the results back into the Parameter Management System or the System 800xA platform respectively. Data presented in the Data Sheet can also be printed
  The Data Sheet view is described in more details Working with the Data Sheet View of Parameter Aspects.

How to Handle Parameter Aspects in the Plant Explorer

In the Plant Explorer your can perform several actions upon aspects. The following considerations are focusing on parameter aspects. The functions you can perform on parameter aspects within the Plant Explorer are:

- Add a new parameter aspect
- Add instance properties to an parameter aspect of an extendable category
- Delete a parameter aspect
- Copy a parameter aspect
- Override a parameter aspect
- Open a parameter aspect in a Data Sheet View
- Print a parameter aspect

These functions will be described in the following subsections.

Add a New Parameter Aspect

Select the **New Aspect...** item of the context menu on an object in the Plant Explorer's tree view.

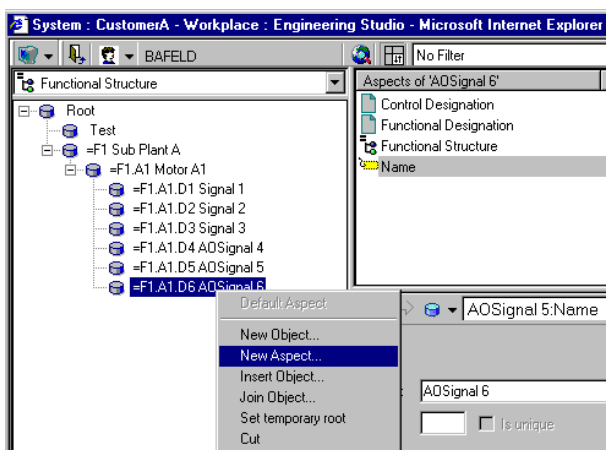


Figure 118. Create New Parameter Aspect

This will popup the “New Aspect dialog”

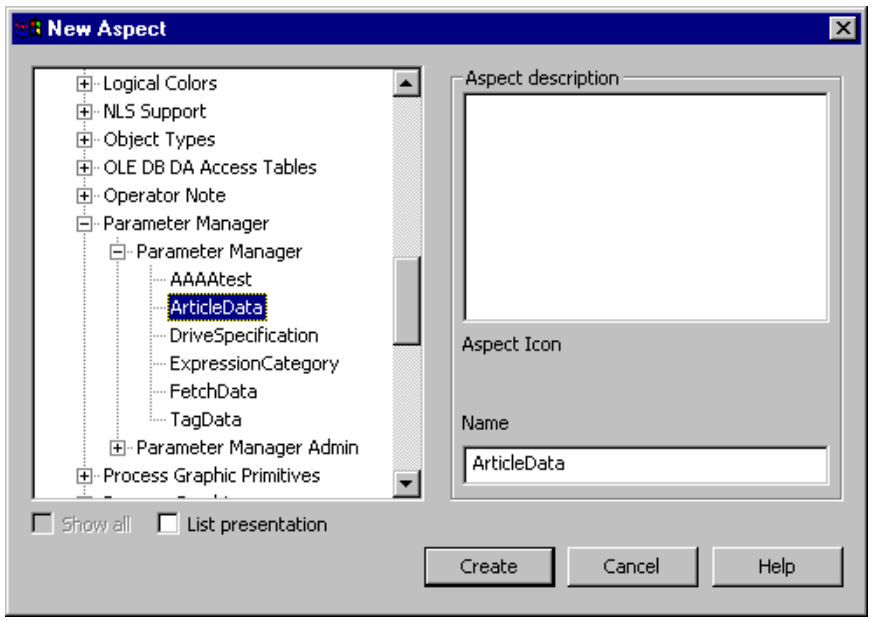


Figure 119. Select Aspect Category

Select the parameter aspect type and then the parameter aspect category. Enter a name and press the **Create** button and the parameter aspect will be inserted

Selected this new created aspect, the properties of this aspect are shown in the Aspect Window. You can now modify the values according to the data type of the property:

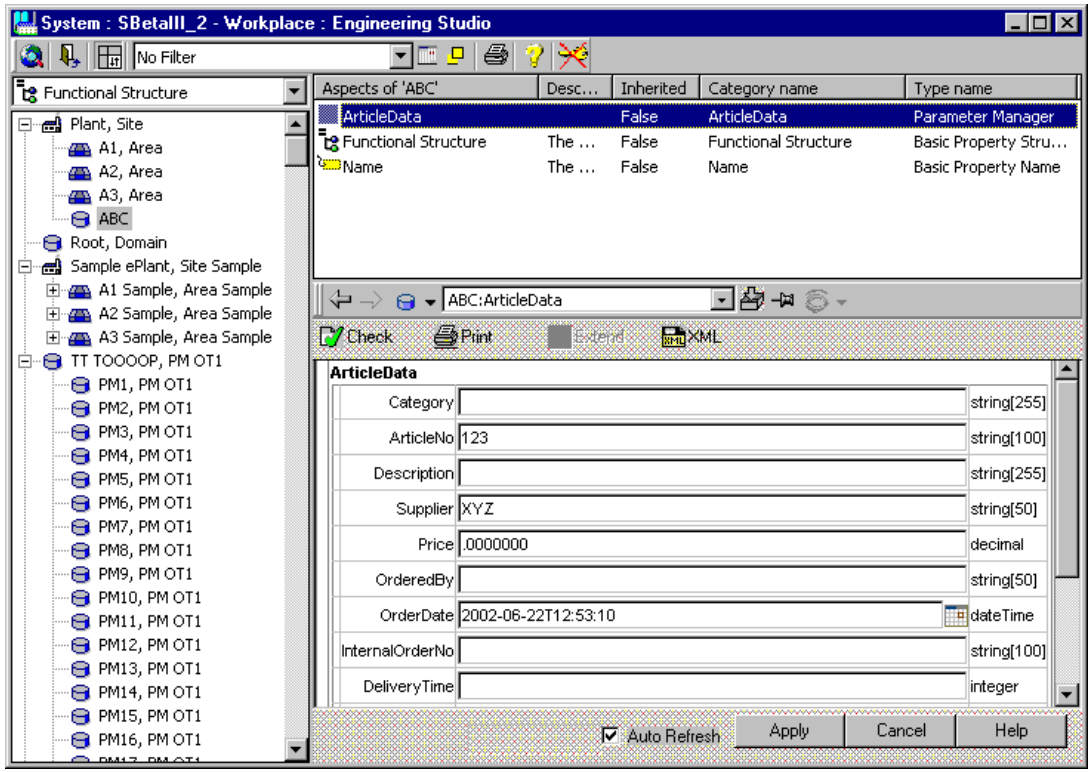


Figure 120. Properties of Parameter Aspect 'Article Data'

Add Instance Properties to a Parameter Aspect

Select the **New Aspect...** item of the context menu on an object in the Plant Explorer's tree view.

Select the parameter aspect type and then the Parameter aspect category (must be an **extendable category**). Enter a name press the **Create** button and the Parameter aspect will be inserted

Selected this new created aspect. Open the context menu for the top element (Documents in this example). Select New Instance Property from the context menu.

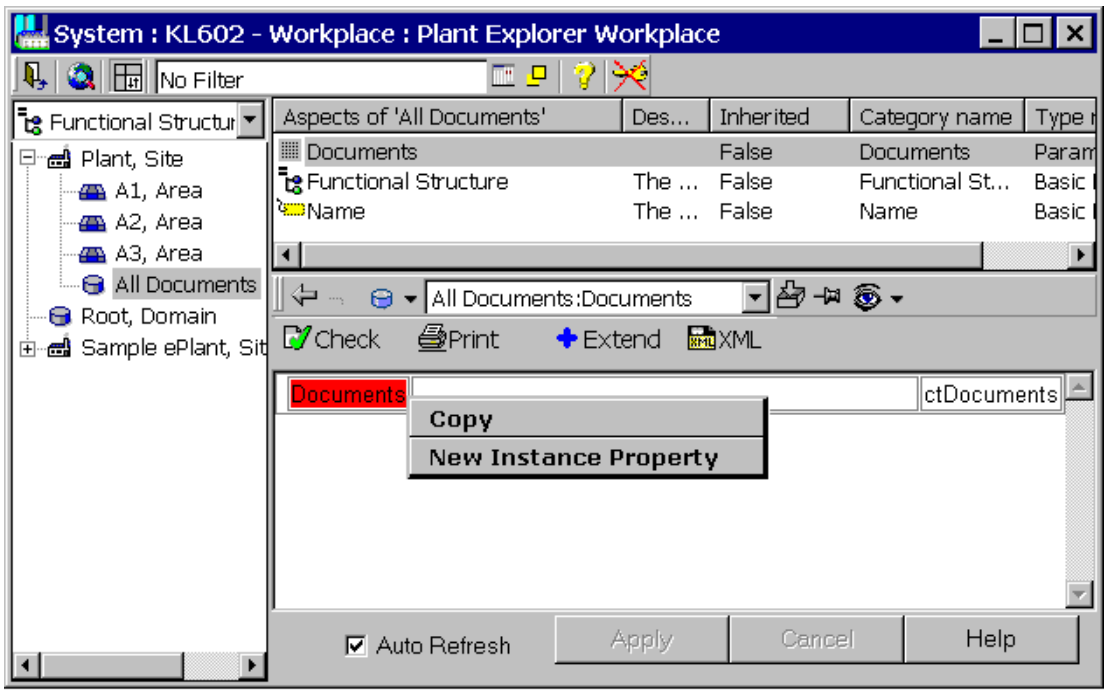


Figure 121. Add Instance Specific Property

The system will now display a dialog to specify the new property.

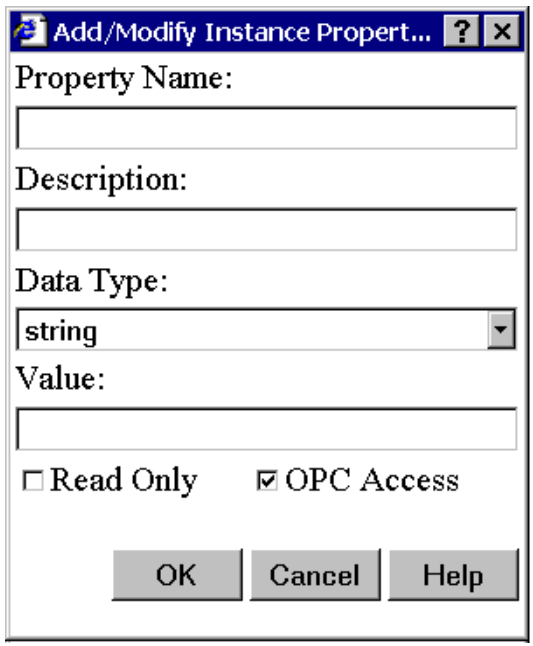


Figure 122. Dialog to Add / Modify Instance Specific Properties

The Help button in Add/Modify Instance Property dialog is not supported.

Now enter the properties and values for this instance. The following table describes the different fields of the dialog.

Table 21. Description for Add / Modify Instance Specific Properties

Field	Description
Property Name	Name of the new property. The name may contain blanks. Example: “Motor Speed”. Input is required.
Description	Description text for the property.
Data Type	Data type of the property. The following types are supported: string, integer, float, boolean. Input is required.
Value	Value of the property.
Read Only	If selected, other aspect system have only read access to the property.
OPC Access	If selected, the property can be accessed via OPC.

Press **OK** to close the dialog. The Parameter Manager will now display the updated list of properties for the aspect.

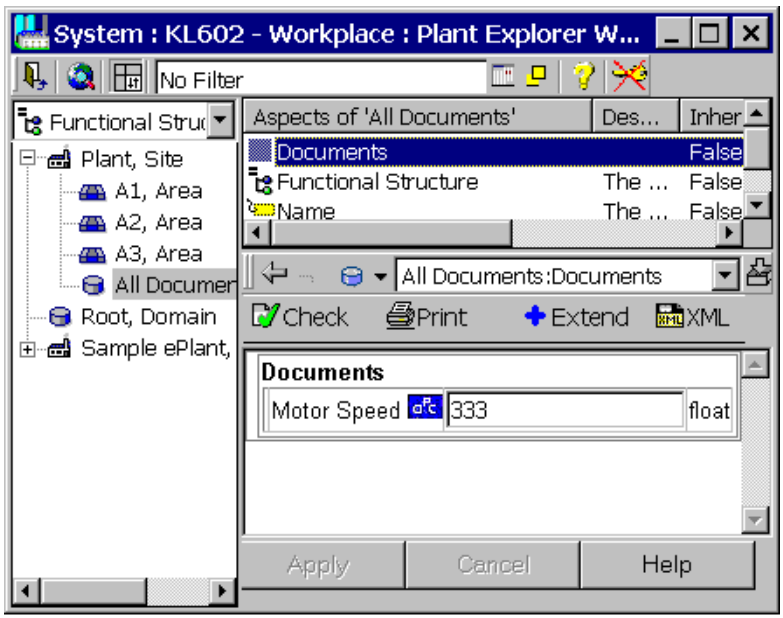


Figure 123. Aspect with Instance Specific Property

Delete a Parameter Aspect

Select the **Delete** item of the context menu on the parameter aspect, you want to delete, in the Plant Explorer's aspect window.

Copy a Parameter Aspect

To copy an aspect follow the steps:

- Select the **Copy** item of the context menu on the parameter aspect, you want to copy, in the Plant Explorer's aspect window
- Select the object, where you want to paste the parameter aspect, in the Plant Explorer's tree view

- Select the **Paste Special > Paste Aspect** item of the context menu on this object

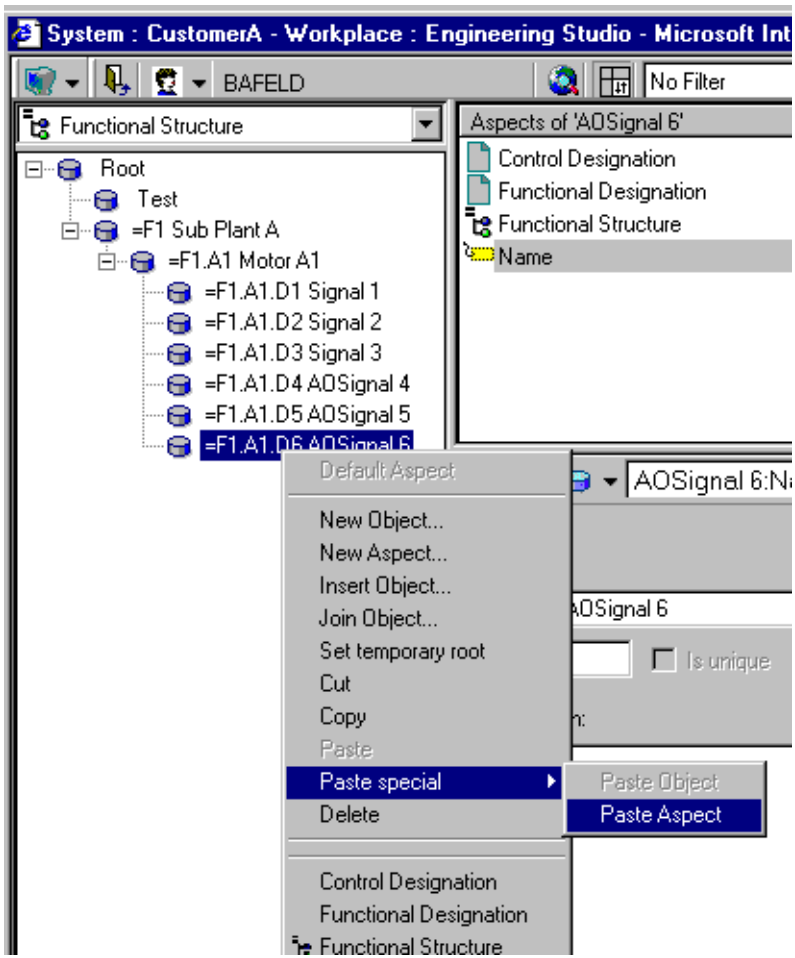


Figure 124. Paste a Parameter Aspect

The parameter aspect will be copied

Override a Parameter Aspect

If a parameter aspect was created from an object type and was inherited, the inherited flag is set to **True**, only one instance (the one of the object type) exists in the Parameter Manager and it is not possible to change attributes in the aspect preview window. See also Parameter Aspects in ObjectType Definition and Usage.

To change this behavior you can select **Override** from the context menu on the parameter aspect. The flag is set to **False** and an own new instance is created for this parameter aspect in the Parameter Manager.

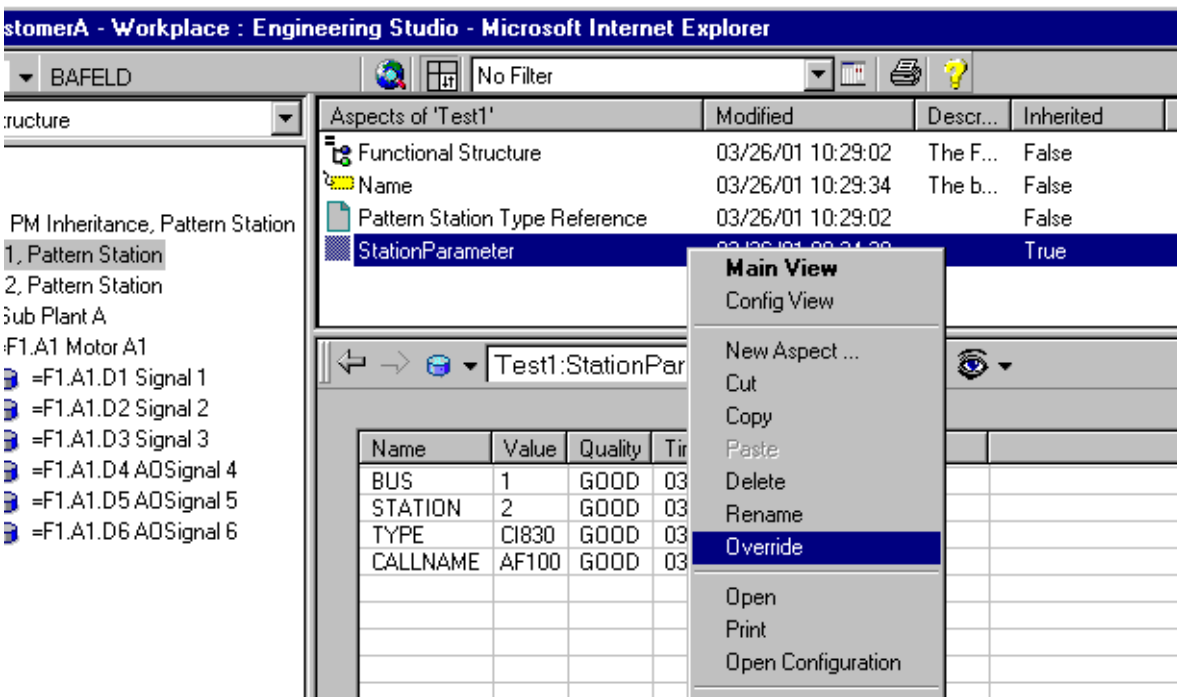


Figure 125. Override a Parameter Aspect

Property Editor (PED) Print

PED prepares its printout as HTML files managed by Internet Explorer (IE), therefore it is recommended to define the user defined page settings inside IE. Open IE through **File > Page Setup...** .
Figure 126 shows an example for header and footer configuration. For details about the control variables like &b, &P and etc. see IE Help, Page Setup.

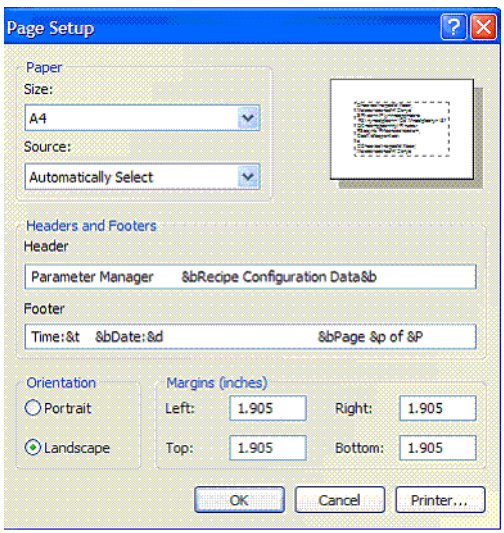
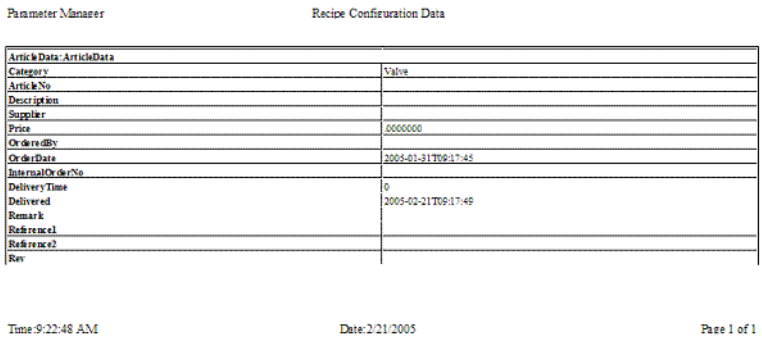


Figure 126. Internet Explorer Print Setup

The printout shown in based on the settings of Figure 126.



Parameter Manager Recipe Configuration Data

ArticleData:ArticleData	
Category	Valve
ArticleNo	
Description	
Supplier	
Price	0000000
OrderedBy	
OrderDate	2005-01-31T09:17:45
InternalOrderNo	
DeliveryTime	0
Delivered	2005-02-21T09:17:49
Remark	
Reference1	
Reference2	
Rev	

Time:9:22:48 AM Date:2/21/2005 Page 1 of 1

Figure 127. Property Editor Print Out

Open the Properties of a Parameter Aspect

To open a Parameter Aspect in Data Sheet view select the **Open Properties** item of the context menu on the parameter aspect, you want to view, in the Plant Explorer's aspect window.

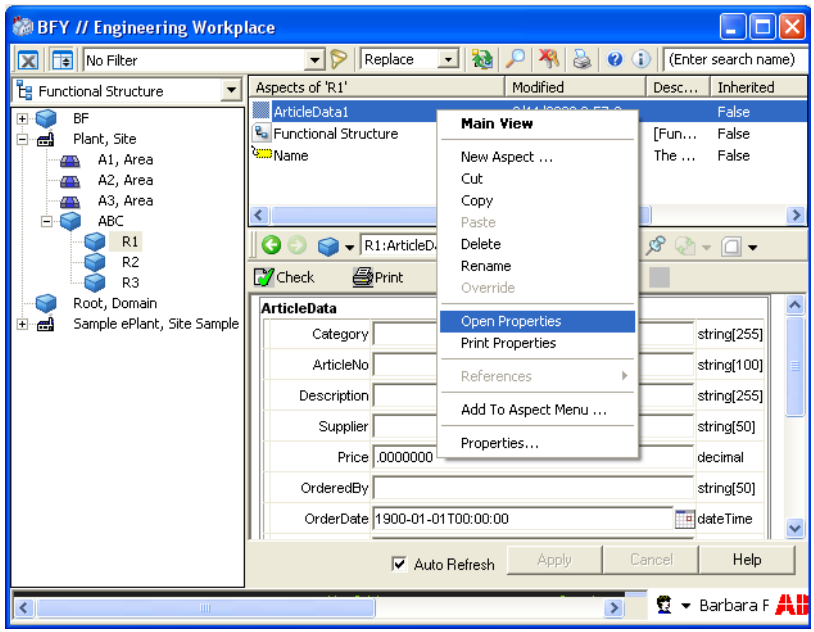


Figure 128. Open Properties of a Parameter Aspect

The Data Sheet view based on Bulk Data Manager of this aspect will be opened:

For working in the Data sheet View see Working with the Data Sheet View of Parameter Aspects

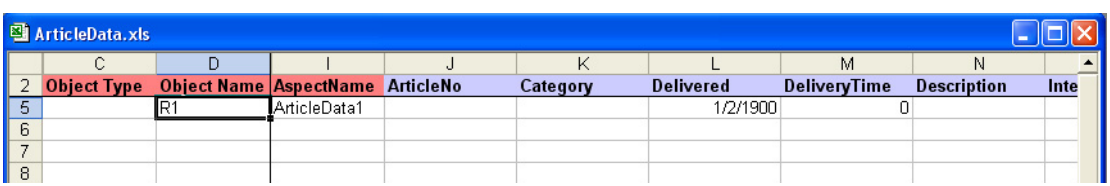


Figure 129. Data Sheet for one Parameter Aspect

Print the Properties of a Parameter Aspect

Select the **Print Properties** item of the context menu on the parameter aspect, you want to print, in the Plant Explorer's aspect window.

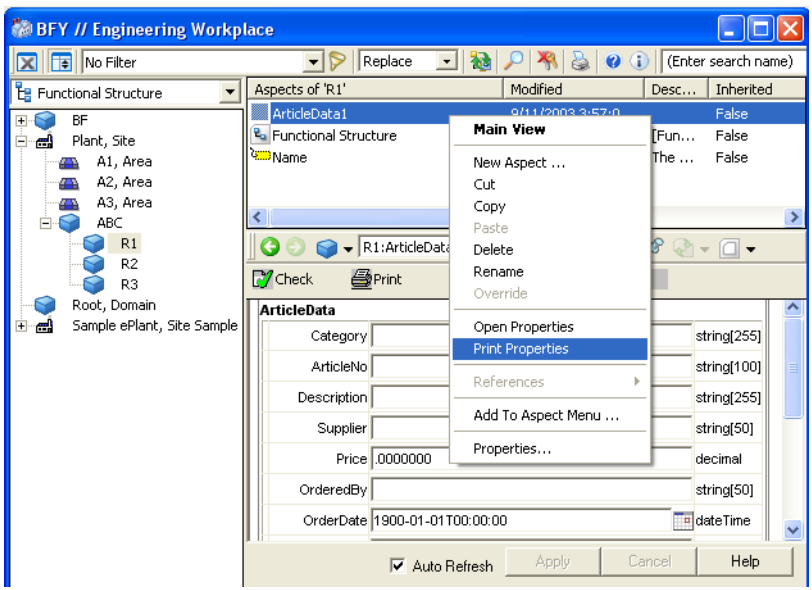


Figure 130. Print Parameter Aspects

All parameter aspects in the same category of the whole subtree will be printed.

Working with the Data Sheet View of Parameter Aspects

Within the Data Sheet view the user can modify data and store the results back into the Parameter Management System or the System 800xA platform respectively.

Configuring the Data Sheet

The Data Sheet is automatically configured to show the properties defined for the related category.

You can hold down the <Ctrl>-key during open to select only a subset of properties to show on the data sheet.

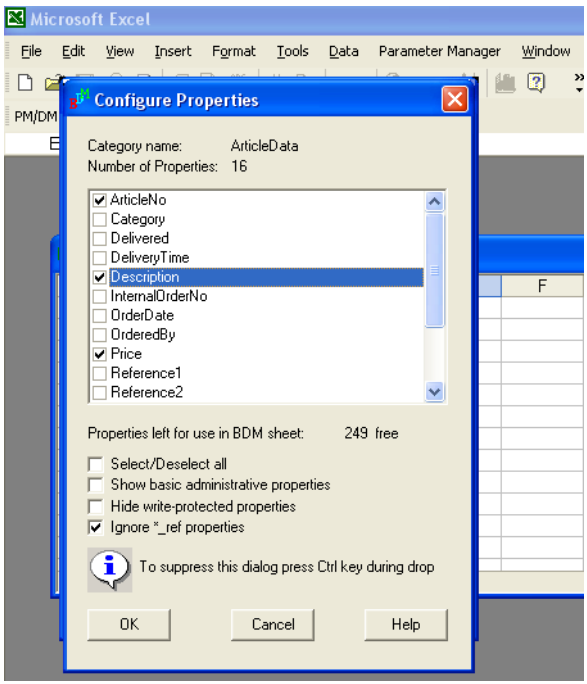


Figure 131. Configure Properties Dialog During Open

the result will be:

Figure 132. Data Sheet of a Parameter Aspect; Selected Properties

To see all Parameter Aspects in the subtree choose menu item

Parameter Manager > Show Subtree or the **Show Subtree** button in PM-Toolbar:

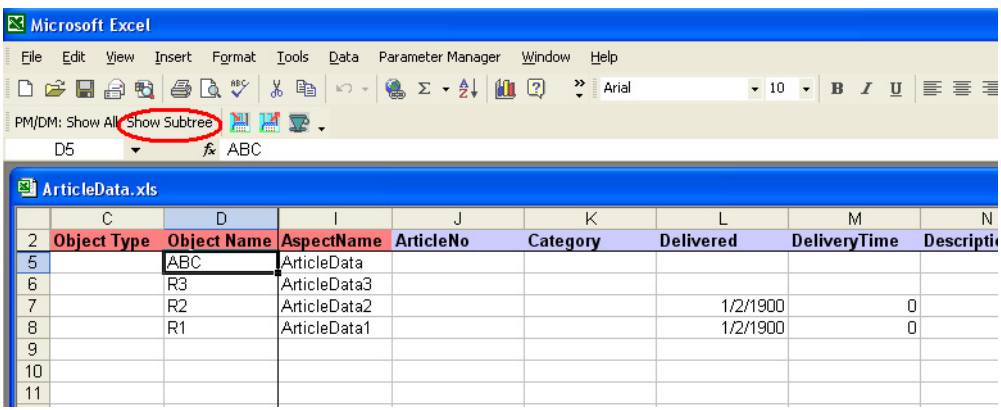
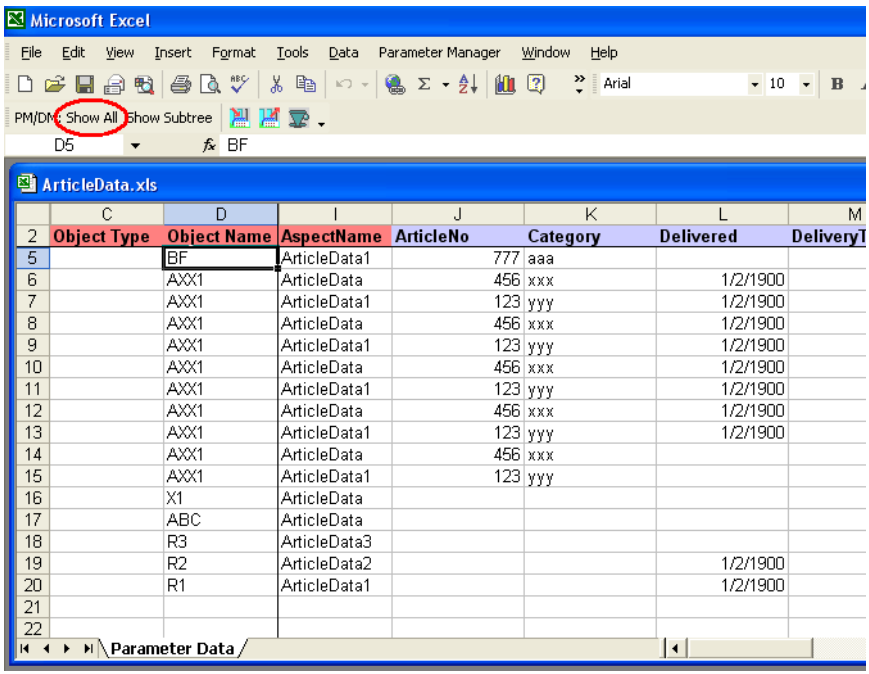


Figure 133. Data Sheet After 'Show Subtree'

To see all Parameter Aspects of this category in the whole structure choose menu item

Parameter Manager > Show All or the **Show All** button in PM-Toolbar:



	C	D	I	J	K	L	M
2	Object Type	Object Name	AspectName	ArticleNo	Category	Delivered	DeliveryT
5		BF	ArticleData1	777	aaa		
6		AXX1	ArticleData	456	xxx	1/2/1900	
7		AXX1	ArticleData1	123	yyy	1/2/1900	
8		AXX1	ArticleData	456	xxx	1/2/1900	
9		AXX1	ArticleData1	123	yyy	1/2/1900	
10		AXX1	ArticleData	456	xxx	1/2/1900	
11		AXX1	ArticleData1	123	yyy	1/2/1900	
12		AXX1	ArticleData	456	xxx	1/2/1900	
13		AXX1	ArticleData1	123	yyy	1/2/1900	
14		AXX1	ArticleData	456	xxx		
15		AXX1	ArticleData1	123	yyy		
16		X1	ArticleData				
17		ABC	ArticleData				
18		R3	ArticleData3				
19		R2	ArticleData2			1/2/1900	
20		R1	ArticleData1			1/2/1900	
21							
22							

Figure 134. Data Sheet After 'Show All'

The Menus of the Data Sheet

to work with the data sheet view use the **PM-Toolbar**-Buttons,

Figure 135. Toolbar of Parameter Manager

or the Menu **Parameter Manager**.

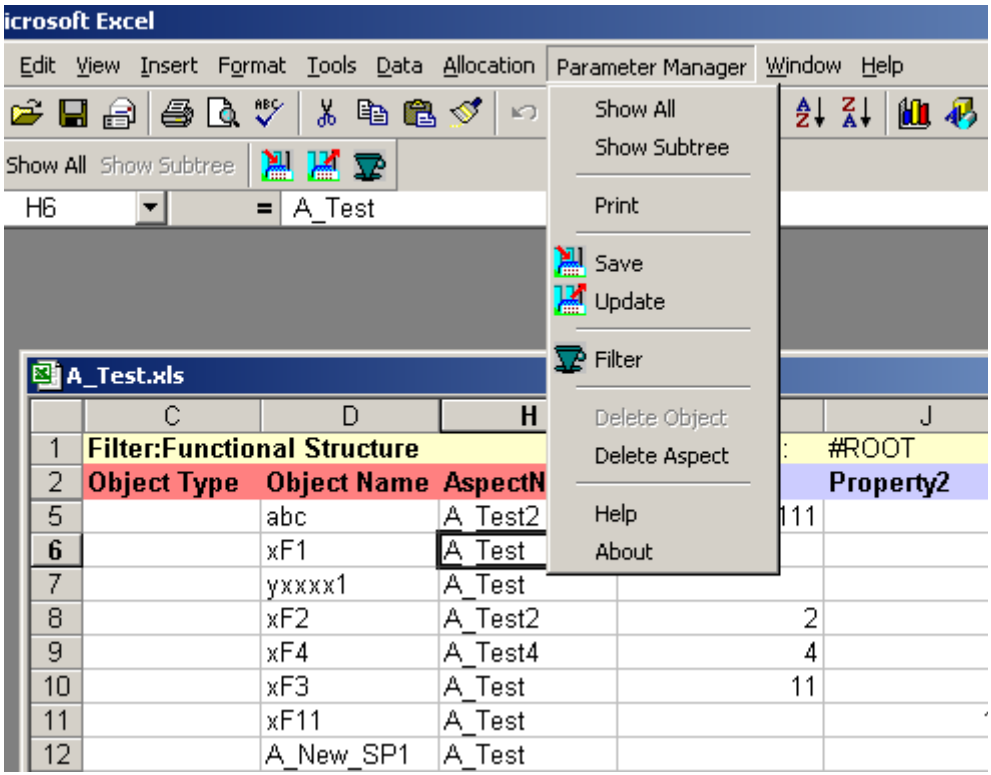


Figure 136. Menu 'Parameter Manager'

or the **Context Menu** (right-mouse-click after selection):

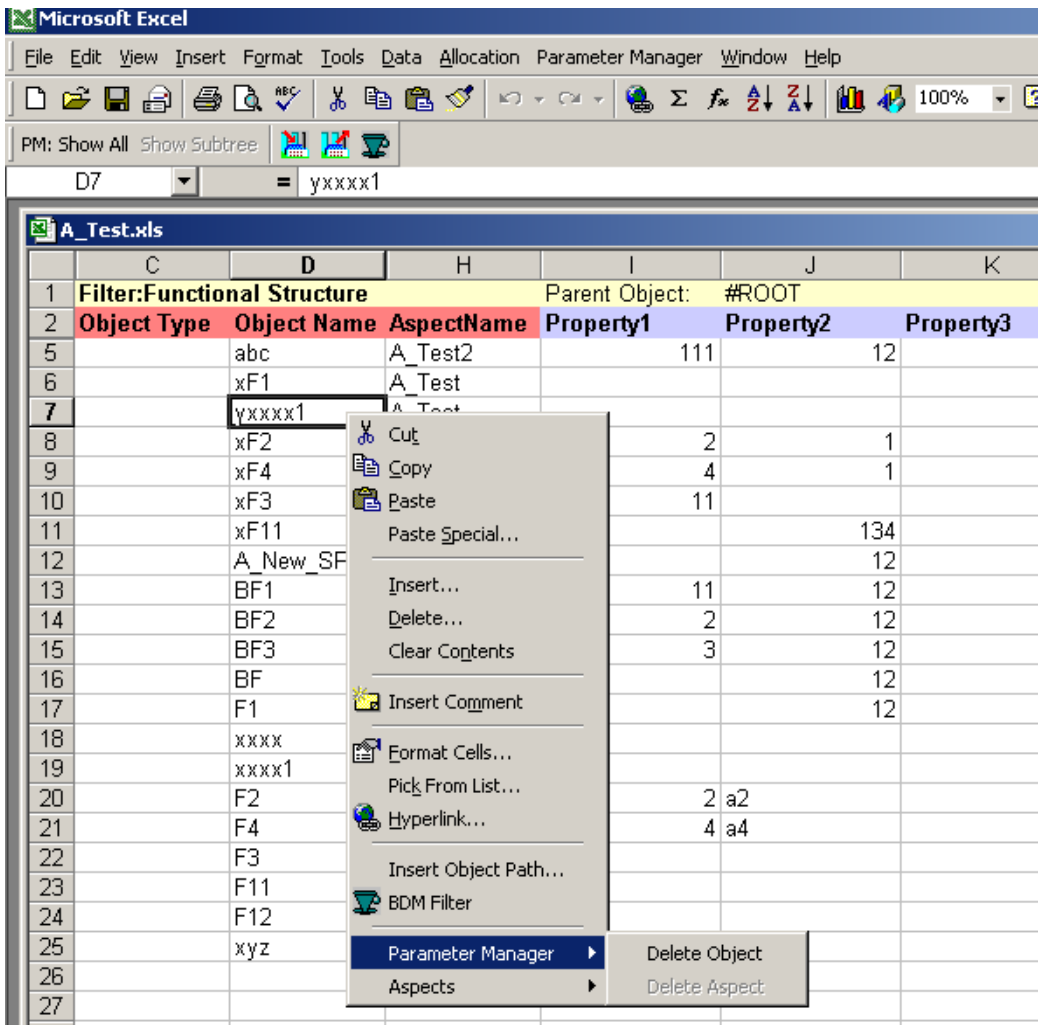


Figure 137. Context Menu

This is a short description of the buttons and menu items:

- **Show All** (Button or Menu item)

  To see all Parameter Aspects of this category in the selected structure (Parent is ROOT)

- **Show Subtree** (Button or Menu item)

  To see all Parameter Aspects of this category in the whole subtree

- **Print** (Menu item)

  Print the parameter aspects actually shown in the data sheet

- **Save** (Button or Menu item)

  Save the changes you have made to the actually shown parameter aspects back into the Parameter Management System or the System 800xA platform respectively.

- **Update** (Button or Menu item)

  Update the parameter aspects from Parameter Management System or the System 800xA platform respectively according to the Filter-settings.

- **Filter** (Button or Menu item)

  Show the BDM-Filter Dialog, here you can give filter-specifications for the aspect property-values

- **Delete Object** (Menu item or Context-Menu item)

  Delete the selected objects, this Menu item is only active if Object Names are selected (see Delete Objects or Aspects)

- **Delete Aspect (**Menu item or Context-Menu item)

  Delete the selected aspects, this Menu item is only active if Aspect Names are selected (see Delete Objects or Aspects)

- **Help** (Menu item)

  Show the Help of Parameter Manager

- **About** (Menu item)

  Show version information of Parameter Manager

Modify Property-Values

You can change values of properties and then save them back to the system. After save an update from system will be performed automatically.

To filter records, where you want to change the properties, the Bulk Data Manager Filter-Dialog is available, but you can also use the filter-mechanisms of MS-Excel.

If you change a property-value to the value ‘’ (blank), automatically the string “#NULL” will be inserted.

To avoid error messages do not save inherited Parameter aspect data. If you have a mix of non-inherited and inherited Parameter aspects in your data sheet remove the rows with the inherited aspects and save the rest or save the whole sheet and ignore the error sheet.

Inside the data sheet it is possible to **create new Instance-Properties** onto a Parameter aspect if the Aspect Category is an **Extendable Category** or a customized category.

To do this perform the following actions:

- Insert a new column as last column in the area of the aspect-properties.
- Give the name of the new instance property in the header line.
- Fill a value for the property for those objects, which shall have this instance property.
- Click **Save.**

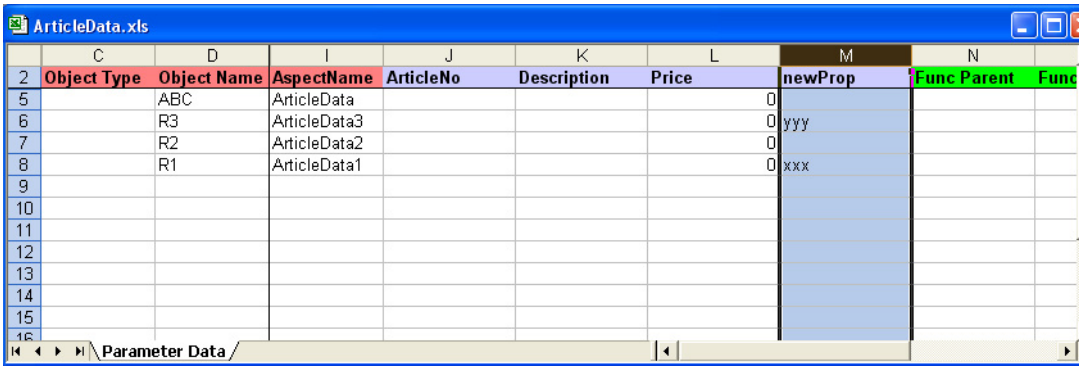


Figure 138. Create a new Instance Property

Figure 138 shows the property ‘newProp’, which is a newly inserted Instance property Column, which will be added to the object ‘R1’ and ‘R3’, but not to the objects ‘ABC’ and ‘R2’ after ‘**Save**’ was pressed.

Rename Objects or Aspects

To rename an existing object or aspect which is shown in the Data Sheet of an Aspect Category simply edit the column ‘ObjectName’ or ‘AspectName’.

Create Objects or Aspects

It is also possible to create new objects, when working in the Data Sheet of an Aspect Category. Insert a new row or edit the last row, where you must give at least the ‘Object Name’ of the object you want to be created.

The object is created under the parent given in the column ‘Func Parent’. If no Functional Parent is given, then ‘#ROOT’ is assumed. The created object always has an aspect of the given Aspect Category. The name of this aspect is by default the name of the given Aspect Category unless a name is given in the column ‘Aspect Name’.

Delete Objects or Aspects

To delete an existing object or aspect which is shown in the Data Sheet of an Aspect Category simply enter the cell of your selected object or aspect in the column ‘Object Name’ or ‘AspectName’ and right-mouse-click to get the **Context Menu** to select **Parameter Manager Delete Object** or **Delete Aspect** (see **Figure 139, Figure 141).** The data sheet will be extended by additional columns where the delete command will be included (see Figure 140, Figure 142). After pressing the tool bar button “Save” or menu item Save the delete command will be executed and the aspect object and its aspects or the Parameter aspect only will be removed.

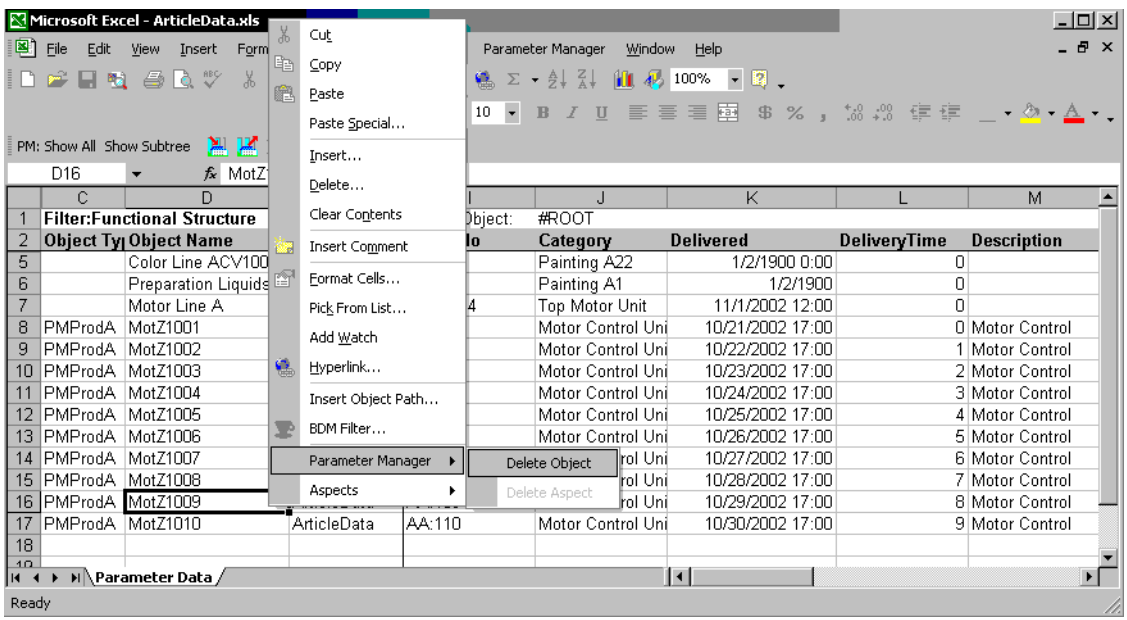


Figure 139. Delete Aspect Object

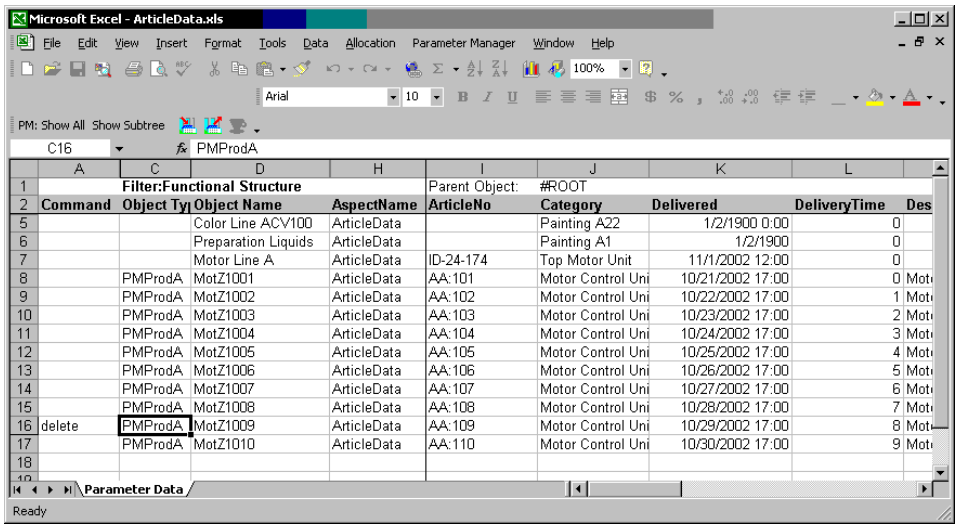


Figure 140. Delete Aspect Object Command inserted

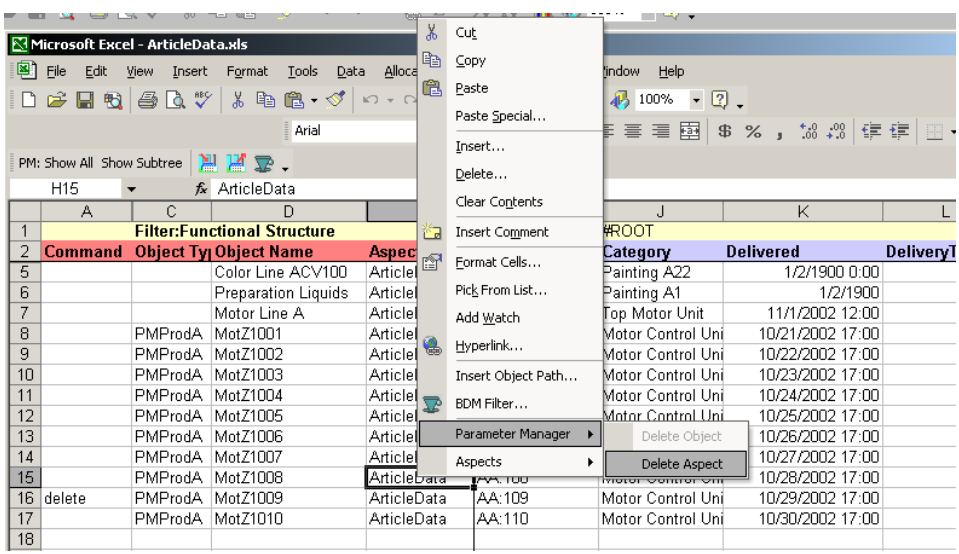
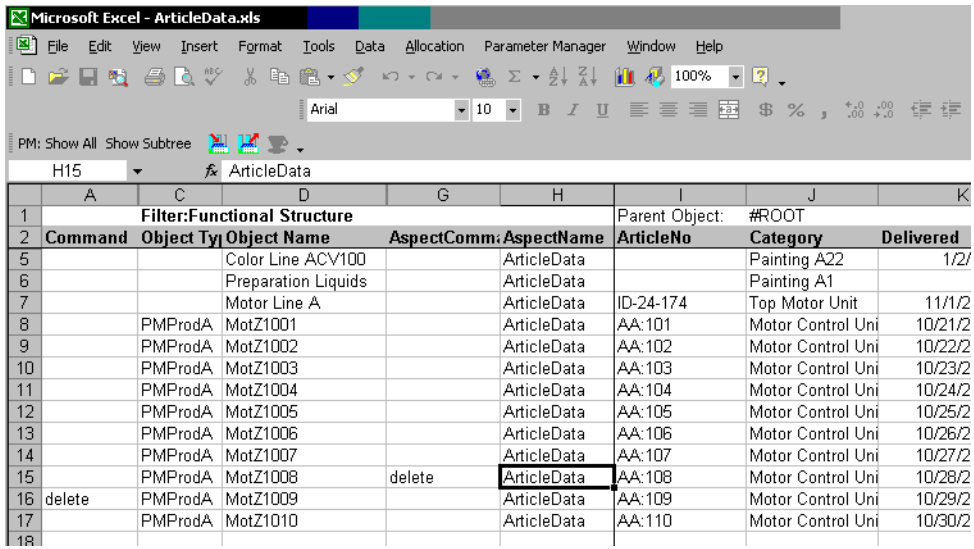


Figure 141. Delete Parameter Aspect



	A	C	D	G	H	I	J	K
1		Filter:Functional Structure				Parent Object:	#ROOT	
2	Command	Object Ty	Object Name	AspectComm	AspectName	ArticleNo	Category	Delivered
5			Color Line ACV100		ArticleData		Painting A22	1/2/
6			Preparation Liquids		ArticleData		Painting A1	
7			Motor Line A		ArticleData	ID-24-174	Top Motor Unit	11/1/2
8		PMProdA	MotZ1001		ArticleData	AA:101	Motor Control Uni	10/21/2
9		PMProdA	MotZ1002		ArticleData	AA:102	Motor Control Uni	10/22/2
10		PMProdA	MotZ1003		ArticleData	AA:103	Motor Control Uni	10/23/2
11		PMProdA	MotZ1004		ArticleData	AA:104	Motor Control Uni	10/24/2
12		PMProdA	MotZ1005		ArticleData	AA:105	Motor Control Uni	10/25/2
13		PMProdA	MotZ1006		ArticleData	AA:106	Motor Control Uni	10/26/2
14		PMProdA	MotZ1007		ArticleData	AA:107	Motor Control Uni	10/27/2
15		PMProdA	MotZ1008	delete	ArticleData	AA:108	Motor Control Uni	10/28/2
16	delete	PMProdA	MotZ1009		ArticleData	AA:109	Motor Control Uni	10/29/2
17		PMProdA	MotZ1010		ArticleData	AA:110	Motor Control Uni	10/30/2
18								

Figure 142. Delete Parameter Aspect command inserted

Parameter Aspects in ObjectType Definition and Usage

The reason to build and use object types is similar to why you would build typical solutions. It makes it possible for you to get dedicated, specialized and tested model or pattern objects to create Aspect Objects out of them in your project. Thus, Object Types decrease the engineering effort and increase the quality.

If you want to have parameter aspects on your Object Types, there are three modes to use parameter aspects. At creation of an Aspect Object out of any Object Type - which includes a parameter aspect:

- the parameter aspect can be **used as template** or
- the parameter aspect will be **copied** or
- the parameter aspect will be **inherited.**

The **used as template** mode means that at creation of the Aspect Object the parameter aspect will not be copied but if you create a parameter aspect sometime later, the one which was created on your Object Type will be used, i.e. copied.

The **copied** mode means that at creation of the Aspect Object the parameter aspect will be copied.

The **inherited** mode means that at creation of the Aspect Object the parameter aspect will not be copied but referenced.

For parameter aspects the standard behavior is the ‘copy’ and ‘use as template’ mode; if you have the necessity to inherit parameter aspects you have to configure the related aspect category.

To allow inheritance proceed in the following way:

4. In the Plant Explorer, select the Aspect System Structure.
5. Browse to the Parameter Manager, Aspect System, Parameter, Aspect Type and select the Parameter Aspect Category you want to allowed to be inherited, for example SignalParameter.
6. Select in the Aspect List Area the Aspect Category Definition aspect.
7. In the Preview Area choose Configuration and set Inheritance levels to **Unlimited** and click **Apply**.

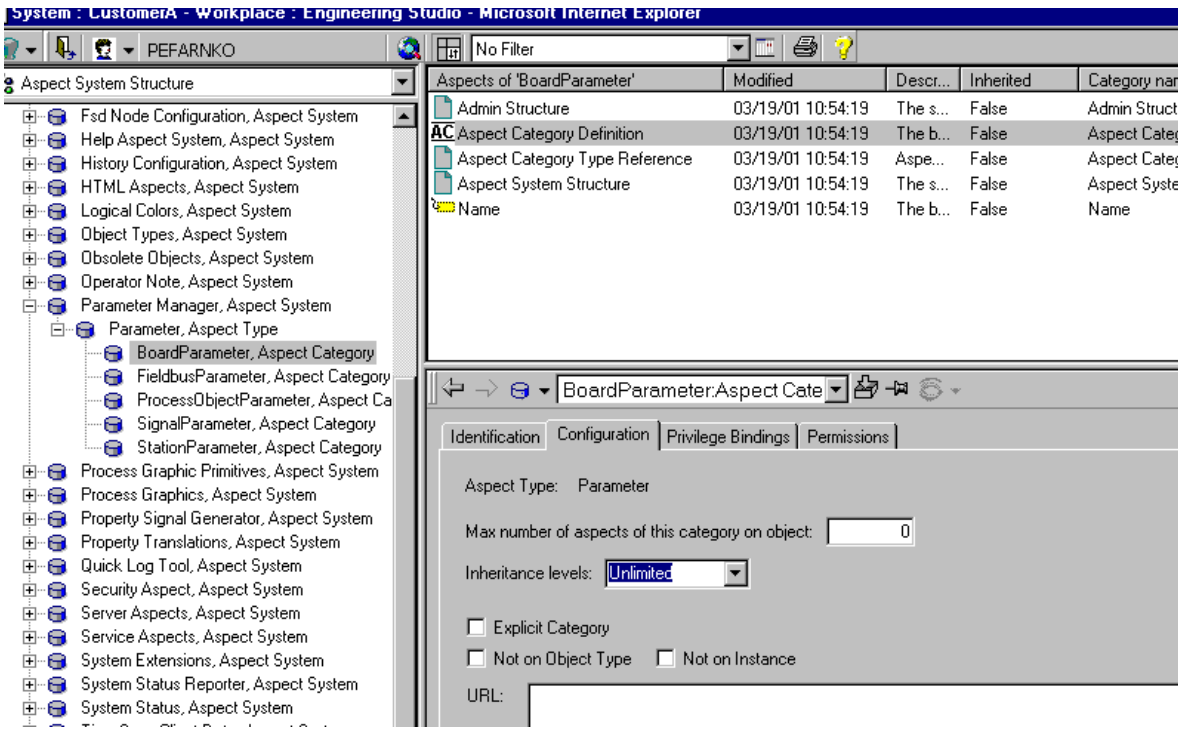


Figure 143. Set Inheritance Level to Unlimited

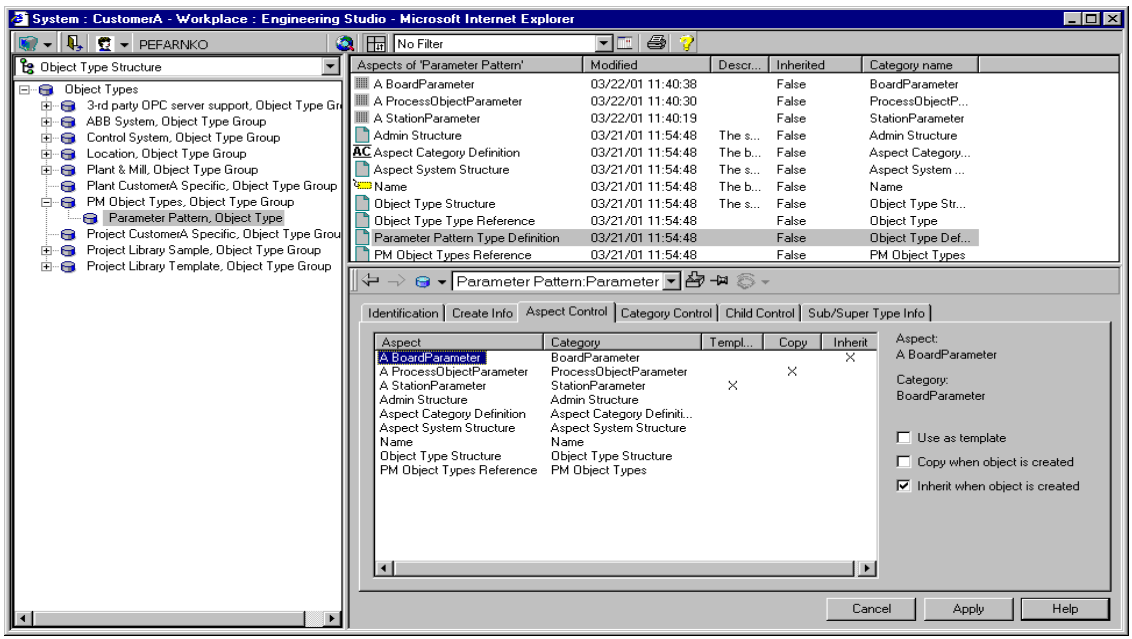


Figure 144. Configuration of Parameter Aspect Behavior

Figure 144 shows the configurations which you have to do at Object type definition to set the requested parameter aspect behavior at object creation. To simplify matters, the example shows the three modes on one Object type. The Object Type *Parameter Pattern* contains - beside others - three parameter aspects:

- one is called *A BoardParameter*, selected for inheritance
- the second is called *A ProcessObjectParameter*, selected to copy

the third is called *A SignalParameter*, selected to use as template.

See Figure 144

Audit Trail Events

Parameter Manager creates Audit Trail Events for modification operations onto Parameter aspects if the Audit Trail feature is enabled within the current 800xA system:

- Modification of Parameter properties will be listed in Alarm & Event aspect, column “Message Description” or NEWVALUE. The modified property will be shown like:
  PropName: <property name>, OldValue= <..>, NewValue=<...>.

Authentication

System 800xA supports Re-Authentication or Double Authentication, which can be configured for Parameter Manager categories:

- By default Authentication is not active. For example to activate Re-Authentication and Double Authentication for the Parameter category ArticleData, select Aspect System Structure.Parameter Manager (Aspect System).Parameter Manager (Aspect Type).ArticleData and check in aspect Aspect Category Definition the check boxes for Re-Authentication or Double Authentication.

E-Signature Properties

System 800xA supports E-Signature functions, First and Second Signature which can be activated for Parameter Manager categories.

- By default E-Signature is not active. For example to activate E-Signature for the ArticleData category, select Aspect System Structure.Parameter Manager (Aspect System).Parameter Manager (Aspect Type).ArticleData, and check in aspect Aspect Category Definition the check boxes for First Signature or Second Signature.

Section 7 Document Manager

This section describes Document Manager. It is meant for users to get familiar with the functions, commands and utilities of Document Manager. To get the most recent hints, recommendations and settings please read the actual Release Notes which are delivered along with the product.

When Environment support is enabled for a system, Document Manager can be used in Engineering Environment and in Production Environment. However, functionalities related to document aspect verbs such as set file, check-in, check-out etc are blocked in both the Production Environment and Engineering Environment.

Available menu items are different in the two environments to a certain extend.

If you intend to enable Environment Support - Configuration Wizard - "Environment Setup", ensure that all Plant Explorer(s) are closed and re-open it after Environment setting is executed.

Document Manager Functions

Document Manager is a component of the Engineering Workplace.
It integrates documents into Aspect Objects™ and allows you to navigate, retrieve, and maintain these documents.

The documents are typically dynamic Word documents, dynamic Excel workbooks or dynamic AutoCAD drawings making use of the Property Reference functionality (see Insertion of Property References into a Document).

Document Manager is composed of several software packages:

- Document Manager including an interface to the System 800xA platform to enable communication with other components of the Engineering Workplace.
- An extension offering additional functionality in Microsoft Word documents.
- An extension offering additional functionality in Microsoft Excel documents.
- An extension offering additional functionality in AutoCAD drawings.

Applications like Microsoft Word, Microsoft Excel, AutoCAD or FrameMaker™ used to handle the contents of the documents are optional applications and are not delivered with Document Manager.

The functionality of Document Manager provides you with excellent tools for the creation and administration of dynamic documents, for example, tag sheets in Excel featuring bidirectional property references and dynamic areas to access Aspect Object and aspect attributes.

Supported Documents and Document Tools

See the *System 800xA, Installation (3BSE034678*)* manual regarding the supported product versions of Microsoft Word, Microsoft Excel and AutoCAD.

Other Windows® applications available at the customer's site can easily be integrated into the Document Manager by the user. The level of integration (for example, supporting a Print command from the Engineering Workplace and the Document Manager) depends on the tool considered.

Some sample templates to be used as base for new document aspects are included and user defined templates can easily be added to Document Manager (see SectionAdding, Modifying and Deleting Templates).

What You Can Do with Document Manager

You can use Document Manager to handle documents belonging to Aspect Objects. Apart from creation, deletion and duplication of documents, it offers the possibility to work on documents directly by opening the respective tools and also store administrative data (also called document attributes or document properties) pertaining to these documents in the Document Manager's database.

These document properties contain information about the document itself like various document titles, the author's name, the status of the document (e.g. "under work") and so.

You have the option to insert administrative data pertaining to document aspects and also engineering data from other aspect systems (e.g. from the Parameter Manager) into documents. This dynamic data (also called **property references**) is automatically updated so that the documents always show up to date values.

For more information regarding insertion of references please refer to Insertion of Property References into a Document.

You can import a number of documents via the Import utility.

For more information regarding import of documents please refer to Section 3, Bulk Data Manager. Many of the actions pertaining to document aspects are triggered by selecting respective functions in the Plant Explorer.

System Creation

Create a new system.

When creating a new system, a **System Extension** dialog is available in the Configuration Wizard where you can check the system extensions or components which you want included in your system. To get the **Document Manager** functions in your system, click the check box ***DM & PM Application***

Category Definition and Usage

A predefined Document Aspect Category named *Document* is already available in Document Manager. Aspects instantiated out of this category automatically contain a predefined set of document properties.

The category definition for category *Document* can be modified to your needs, i.e. you can delete, modify or add new properties to the category. It is also possible to define new document categories for special needs, for example a category which contains (apart from internally used properties such as *FilePath* which cannot be modified or deleted) only one or two properties.

You can configure different kinds of Document Aspect Categories:

- **Table-like Categories**: all categories having only simple properties are called Table-like Categories.
- **Structured Categories**: all Parameter Categories having one or more subcategories as structured properties are called Structured Categories.
- **Extendable Categories**: categories created without any property but having the possibility to add instance specific properties are called Extendable Categories.

Document Category Configuration is done within the Plant Explorer. Document categories are displayed as objects in the *Aspect System Structure.* Create, copy, rename, and delete operations can be performed on these objects. Copying these objects from one system to another with the Import / Export tool is also possible.

Document Category objects are placed below the *Document Manager Documents* Type Object (see Figure 145).

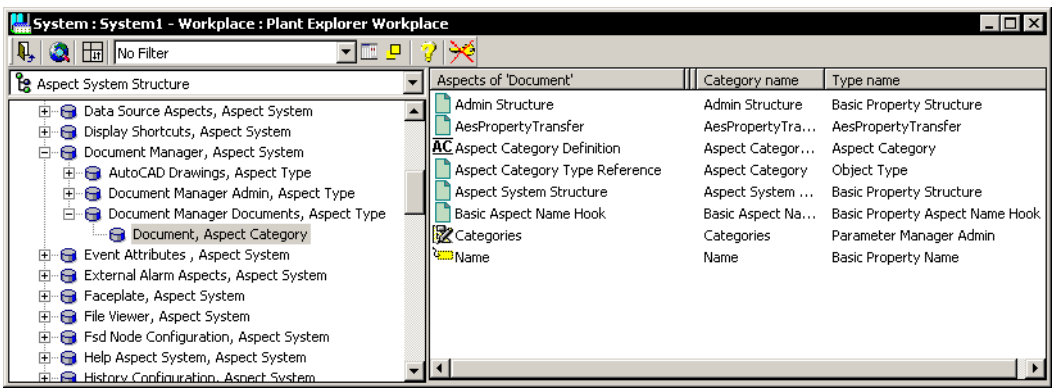


Figure 145. Document Category Definition in Aspect System Structure

For more information regarding category definition and modification please refer to Configuration of Parameter Aspect Categories.

Object Type Definition and Usage

The reason to build and to use object types is similar to why you would build typical solutions. It makes it possible for you to get dedicated, specialized and tested model- or pattern- objects to create true Aspect Objects out of them in your system. Thus, Object Types decrease the engineering effort and increase the quality.

If you want to have document aspects on your Object Types, there are three modes to use document aspects.

At creation of an Aspect Object out of any Object Type - which includes a document aspect

- the document aspect can be **used as template** or
- the document aspect is **copied** or
- the document aspect is **inherited.**

The **used as template** mode means, that at creation of the Aspect Object the document aspect is not copied but if you create a document aspect sometime later, the one which was created on your Object Type will be used, i.e. copied.

The **copied** mode means that at creation of the Aspect Object the document aspect is copied.

The **inherited** mode means that at creation of the Aspect Object the document aspect will not be copied but referenced.

Concepts of object types are described in *System 800xA, System Planning (3BSE041389*).* For details about how to create object types, refer to *System 800xA, Configuration (3BDS011222*).*

Figure 146 shows the configurations which you have to do at object type definition to set the requested document aspect behavior at object creation. To simplify matters, the example shows the three modes in one object type.

The object type *SpecialObjectType* contains - beside others - three document aspects:

- *DocAsTemplate*, selected to use as template,
- *DocToCopy*, selected to copy and
- *DocToInherit*, selected for inheritance.

Aspect and File Storage

Document aspects, i.e. the aspects themselves, document properties and files are stored redundantly on the server with the storage area hidden from the user. This enables to handle server failure as well as authority control regarding the access to document aspects.

In one server fails, it is no longer used but another server node is accessed which stores all relevant data redundantly. This means that all aspects and files have to be kept in sync on all servers which is handled by Document Manager internally.

If you want to edit a document, you have to check out the aspect first which results in the respective file being exported from the server to your local machine. After applying the necessary changes you can check in the document which transfers the

file back to the main server and all standby servers too. The file is now not accessible anymore till the next checkout.

Access rights to document aspects are checked whenever the user tries to perform an action on the aspect, e.g. check it out. If the required permissions are not given, an error message comes up and the action is not carried out.

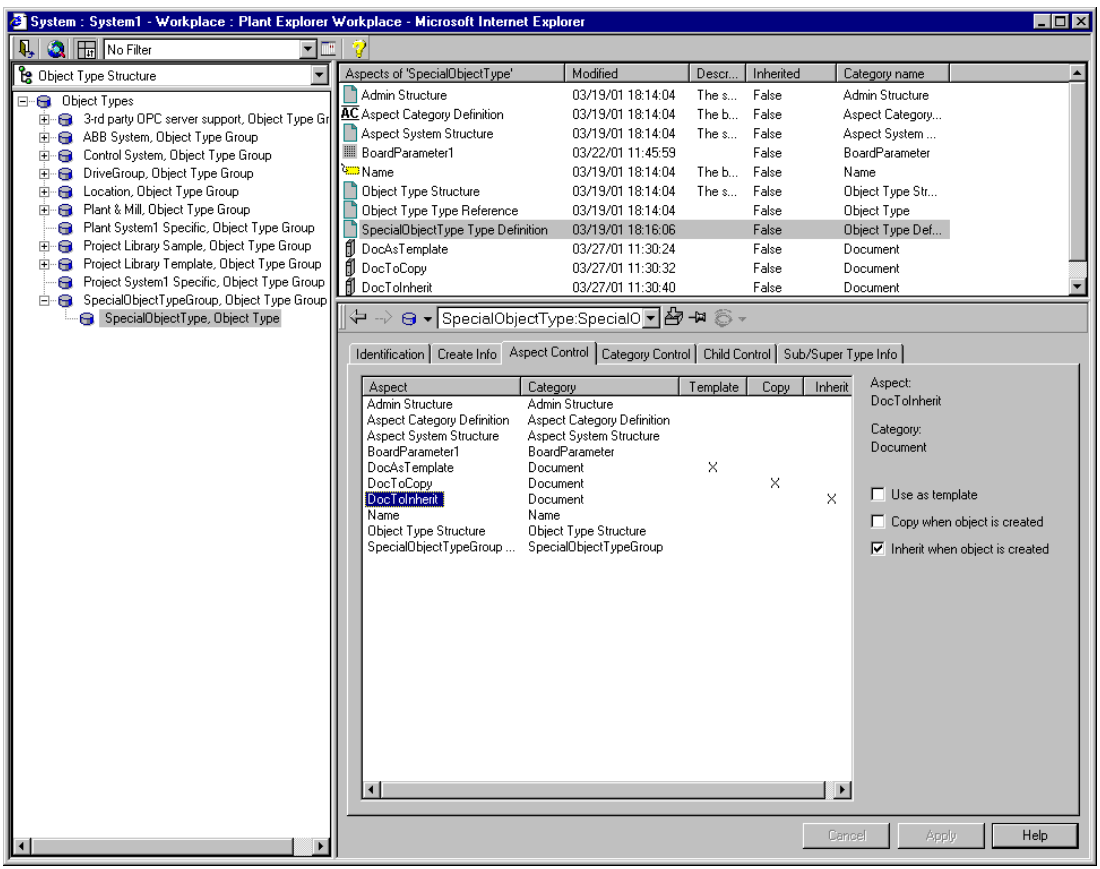


Figure 146. Example Object Type Definition, Containing Document Aspects

Preference Settings

Before working with Document Manager, ensure that Microsoft Office is installed - refer to *System 800xA, Installation (3BSE034678*).*

Folder Structures

Document Manager stores files in two areas in a PC:

- Product files are stored in <drive>:\Program Files\ABB Industrial IT\Engineer IT\Engineering Studio\Engineering Platform\DocumentParameterManager.
- Trace files are stored in MyDocuments\DocumentParameterManager\Tracefiles.

User Interface

Basically there are two user interfaces: the Property Editor and the Data Sheet.

Property Editor

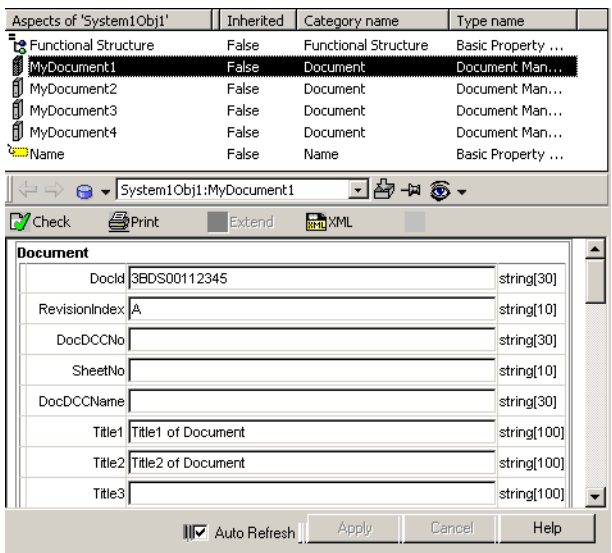


Figure 147. Property Editor for Single Document Aspect

The document aspect's *Main View* in the Plant Explorer's *Preview* area shows the Property Editor where property values for a single document aspect are presented and can be modified or printed.

The Property Editor is meant for viewing and editing aspect properties as well as category properties. The user interface provides support in terms of data type checks, limit checks, pick-lists, default values – provided this information is defined for the related Document Category. Data presented in the Property Editor can also be printed in the same layout.

Data Sheet

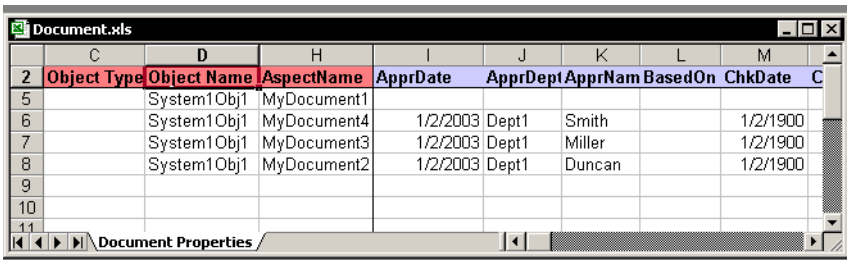


Figure 148. Data Sheet for Multiple Document Aspects

The Data Sheet view of the document aspect is opened when selecting menu item *Open Properties* and is based on the Bulk Data Manager. Here you can comfortably work on multiple instances of document aspects. The Data Sheet is automatically configured to show the properties defined for the related category

You are free to apply filters in order to retrieve for example all aspects of a certain category within the sub tree of the starting object. The Data Sheet (Bulk Data Manager) is automatically configured to show the properties defined for the related category. Within the Data Sheet view, the user can create additional objects having a Document Aspect, store them in the structures like the Functional Structure, delete objects, modify data and store the results back into the Document Management System or the System 800xA platform respectively. Data presented in the Data Sheet can also be printed

For detailed information regarding the Data Sheet please refer to *Working with the Data Sheet View of Parameter Aspects*, here we give only a short overview of the Data Sheet's main functionality.

Data Sheet Document Manager Menu

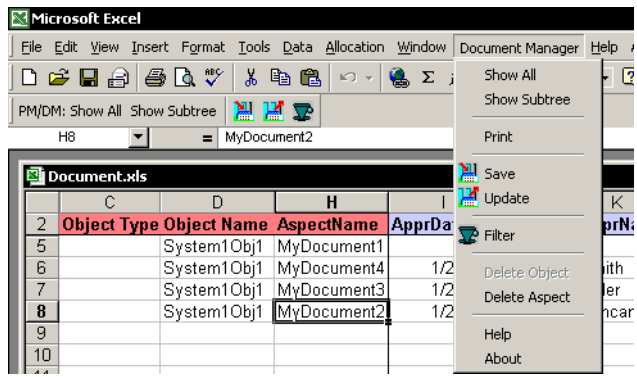


Figure 149. Data Sheet Document Manager Menu

- **Show All** (Button or Menu item)

  To see all document aspects of this category in the selected structure (Parent is ROOT)

- **Show Subtree** (Button or Menu item)

  To see all document aspects of this category in the whole subtree

- **Print** (Menu item)

  Print the document aspects actually shown in the data sheet

- **Save** (Button or Menu item)

  Save the changes you have made to the document aspects back into the Document Manager System or the System 800xA platform respectively.

- **Update** (Button or Menu item)

  Update the document aspects from Document Manager System or the System 800xA platform respectively according to the Filter-settings.

- **Filter** (Button or Menu item)

  Show the BDM-Filter Dialog, here you can give filter-specifications for the aspect property-values

- **Delete Object** (Menu item or Context-Menu item)

  Delete the selected objects, this Menu item is only active if Object Names are selected.

- **Delete Aspect (**Menu item or Context-Menu item)

  Delete the selected aspects, this Menu item is only active if Aspect Names are selected.

- **Help** (Menu item)

  Show the Help for Document Manager.

- **About** (Menu item)

  Show version information for Document Manager

Application Start-up

The Document Manager is an optional part of the Engineering Workplace. It is accessed through the Plant Explorer by opening an aspect representing a document. There are two basic views supported when opening a document aspect:

- Open document (through **View** or **Edit**)
  Starts the related application (for example, Microsoft Word or Microsoft Excel) and opens the document represented by the aspect.
- Open properties (through **Open Properties**)
  Opens the **Data Sheet** presenting various administrative attributes of the document like document identity, titles, etc.

Opening a document is performed in the Plant Explorer's ***Aspect List Area*** (see Figure 150).

Avoid using the Preview area for displaying Excel documents using the Document View. A further attempt to view or edit the aspect can result in an error message and a hanging Excel process. If it happened kill this process using Task Manager.

Working with Document Manager

Document Aspects

In the ***Aspect List Area*** of the Plant Explorer, displaying all aspects of an object (see Figure 150), you can perform several actions upon aspects. The following considerations focus on aspects representing documents. The functions that can be performed on documents within the Plant Explorer are:

- **Add** a new document,
- **Set a File** for a document aspect,
- **Delete** a document,
- **Copy** a document (only the document aspect or the aspect as part of an object),
- **Open** a document (the document itself or its properties),
- **Print** a document,
- **Override** an inherited document,
- **Checkout** the file belonging to a document (only if no Environment Support),
- **Checkin** a modified file or undo a checkout (only if no Environment Support,
- **Get** the last checked-in version of this file (only if no Environment Support),
- **Show the checkin-history** of a document (only if no Environment Support),
- **Reserve** a document (only if Environment support enabled),
- **Release** a document (only if Environment support enabled),
- **Bulk operation** on a set of documents.

How to execute these functions is described in the following chapters.

Aspects of 'MyDocuments'	Modified	Category name
myDocument1	8/6/2004 9:18:57...	Document
myDocument2	8/6/2004 9:19:14...	Document
myDocument3	8/6/2004 9:19:25...	Document
myDocument4	8/6/2004 9:19:38...	Document
myDocument5	8/6/2004 9:19:51...	Document
myDocument6	8/6/2004 9:18:45...	Document
Functional Structure	8/6/2004 9:17:36...	Functional Stru...
Name	8/6/2004 9:17:36...	Name

Figure 150. Plant Explorer Aspect List Area

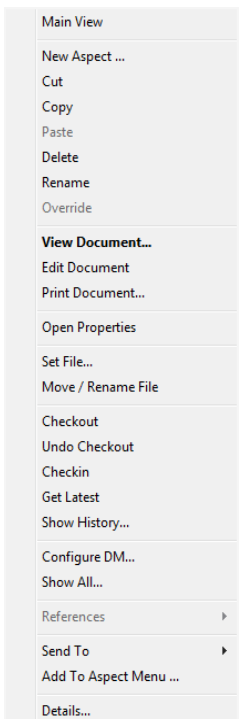


Figure 151. Document Manager Menu Items in Aspect Menu (no Environment Support)

Add a Document Aspect

A new document is created via the New Aspect dialog in Plant Explorer. The new document is visible in the *Aspect List Area* after creation.

Note that a new document always offers the possibility to insert dynamic data (references and document links), provided it is a Word or Excel or AutoCAD document and it is the copy of a template or an existing file - *referenced* non-dynamic documents are not made dynamic automatically.

For further explanations regarding *referenced documents* see below.

For further explanations regarding *dynamic documents* see Understanding Dynamic Data.

To add a document or a drawing, proceed as follows:

- In the ***Aspect List Area,*** press the right hand mouse button for the context sensitive menu to appear.
  Press **New Aspect** to open the **New Aspect** dialog (see Figure 152).
- Select the aspect category **Document** of *Document Manager/Document Manager Documents*, or the aspect category **AutoCAD_Drawing** of *Document Manager/AutoCAD Drawings*, enter a name for your new document and press **Create**. The new aspect is created and visible in the *Aspect List.*

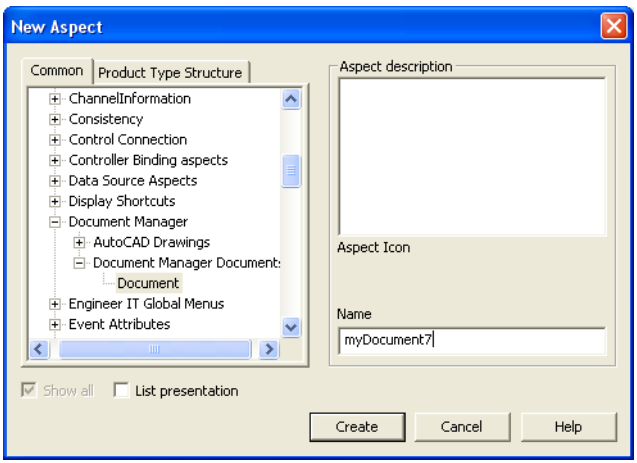


Figure 152. Add New Document Aspect in Plant Explorer

Set a File for the New Document

The new document aspect has no file yet. To set a file, proceed as follows:

- In the *Aspect List Area,* select the document aspect and press the right hand mouse button for the context sensitive menu to appear.
  Press **Set File** to open the selection dialog (see Figure 153).

Please note that the selection dialog also pops up if you want to open the document and have not yet set a file.

You have the following possibilities to select a base file for the new document:

- by copying a template,
- by copying an existing file,
- by moving an existing file,
- by referencing an existing file or
- by entering an URL.

The sections below describe these options in more detail.

Of course it is always possible to set a new file again for a document aspect where another file has already been set. In this case, you are asked if you want to save the previously set file or not.

When Environment support is enabled Set File does implicit Reserve and Release the document.

Set a File Based on a Template

Choosing this option results in the selected template to be copied. Please note that only a few templates are available after installation of **Document Manager**.

Templates are stored as aspects in the Plant Explorer's Library Structure and are loaded on system creation.

For further details and instructions on how to add your own templates please also refer to Understanding Document Manager Templates.

- Press button **Select File Template**,
- select the template of your choice (e.g. Word Blank Document) and press **Ok** (see Figure 153).

If you want to create a document aspect consisting of a whole folder, press **Select Folder Template** and proceed accordingly. In this case, you are asked in a second step to choose a default folder to be opened when opening the document aspect.

Please note that after setting the file you will **not** see it in the folder displayed in the Set File dialog - files belonging to document aspects are stored redundantly on the server and a local copy is created only when the file is checked out or opened for viewing!

In the **Set File for Document Aspect** dialog, click **Change** to modify the aspect name. A dialog appears with a default <object_name>_<aspect_name> for the new aspect name which can be overwritten.

In field **To File in My Documents** you can enter a new subfolder for the file to be stored in when being checked out.

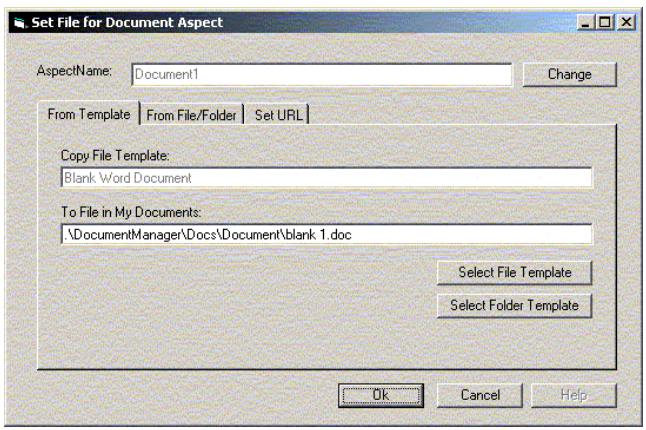


Figure 153. Set File Based on a Template

Set a File as Copy of an Existing File

Choosing this option results in the selected file to be copied.

- Select the **From File/Folder** tab in the **Set File** dialog and then press the **Copy File** button, resulting in a Browse dialog popping up in which you can select the file you want to copy.

Set a File by Moving an Existing File

Choosing this option results in the selected file to be moved.

- Select the **From File/Folder** tab in the **Set File** dialog and then press the **Move File** button, resulting in a Browse dialog popping up in which you can select the file you want to move.

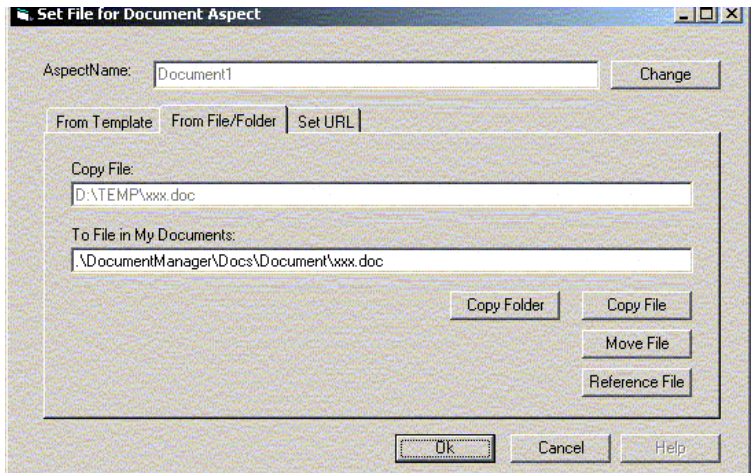


Figure 154. Set File as Copy of Existing File

Set a Folder as Copy of an Existing Folder

To copy a complete folder, select button **Copy Folder** on the **From File/Folder** tab of the **Set File** dialog.

Set a Reference to an Existing File

Choosing this option results in the selected file to be referenced only, no new file is created. Note that the file name cannot be changed.

- Select the **Reference File** button on the **From File/Folder** tab of the **Set File** dialog, resulting in a Browse dialog to pop up in which you can select the file you want to reference.

Please note that a referenced file is not stored on the server. Also, a referenced file can only be viewed but no be edited.

Use references only where really applicable, e.g. to refer to general department documents or the like.

Set an Internet Address

If you want to open an Internet dress directly from Plant Explorer via a document aspect, you can enter the URL in the **Enter URL** field of the **Set URL** tab (see Figure 155).

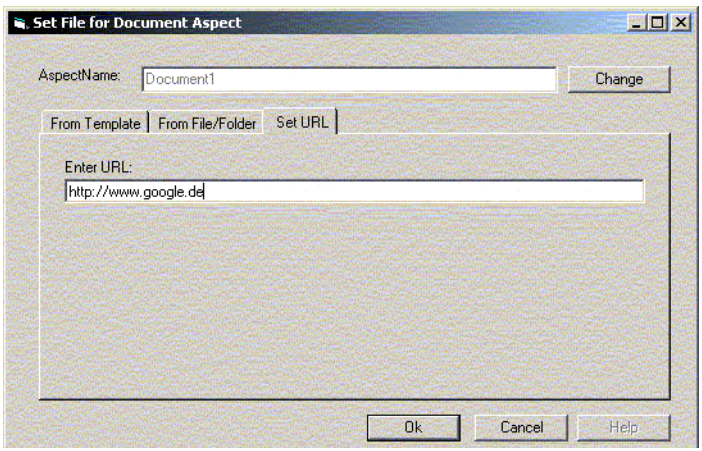


Figure 155. Set an Internet Address for a Document Aspect

The **Move/Rename File** option in the context menu of the Document aspect is disabled in 800xA 5.1 and later versions. On selecting this option, a **Disabled Feature** window appears, with the message “Rename and Move of File is not supported”.

Delete a Document Aspect

Select the document aspect you want to delete in the Plant Explorer’s ***Aspect List Area*** window and press the **Delete** menu item in the context sensitive menu.

The document aspect is deleted from the Plant Explorer as well as from Document Manager, i.e. administrative data in the Document Manager’s database and the document’s file on the server are deleted. In case of a folder aspect, the complete folder is deleted.

If the document was only a reference to a file anywhere on the file system, the referenced file is not deleted, only the document aspect's administrative data in the database.

Copy a Document Aspect

A document aspect can be copied either on its own or as part of an object.

In case of document aspects, the aspect's administrative data as well as the respective file is copied.

To copy a document aspect please follow these steps:

1. Select the aspect to be copied in the ***Aspect List Area*** of Plant Explorer.
2. Choose the menu entries **Copy** from the context sensitive menu or use the Drag and Drop technique.
3. Move to the object you want to place your aspect copy in and select **Paste** from the context sensitive menu.The copied aspect is now visible in the ***Aspect List Area*** of the target object and Document Manager has created a new document aspect instance with a new file that has been copied from the original.

Open a Document Aspect

There are basically two possibilities to open a document:

a. Open the document itself (for viewing or editing) or

b. open information about the document's administrative attributes, also called document administrative data or *document properties*.

Open a Document File for Editing

To open a document in its associated application, select the document aspect in the Plant Explorer's *Aspect List Area* and press the context sensitive menu item **Edit**.

The application, for example Word, is started and the document file is opened for editing.

If references inserted into a Word document should be updated before viewing it, the option **Update References in Word Documents before Editing** has to be set

(see Configuration of Dynamic Document Data Update). In case Auto check-in is enabled, save the edited file before closing it.

Note that the document is checked out before it can be edited. While it is checked-out by you, other users can only view the last checked-in version. If you want to make your changes to the document contents visible to others, you have to check-in the document.

For more information, please refer to Checkout Document.

Open a Document File for Viewing

To open a document in its associated application, select the document aspect in the Plant Explorer's *Aspect List Area* and press the context sensitive menu item **View** or simply double-click the aspect.The document is opened in read-only mode and cannot be edited.

If references inserted into a Word document should be updated before viewing it, the option **Update References in Word Documents before Viewing** has to be set (see Configuration of Dynamic Document Data Update).

Open a Document File for a Quick Preview

Select the document aspect in the Plant Explorer's *Aspect List Area* and in the *Preview Area* select menu item **Document View** (see Figure 156).This option is meant for a quick inspection of the document contents. The document is opened in read-only mode and cannot be edited.

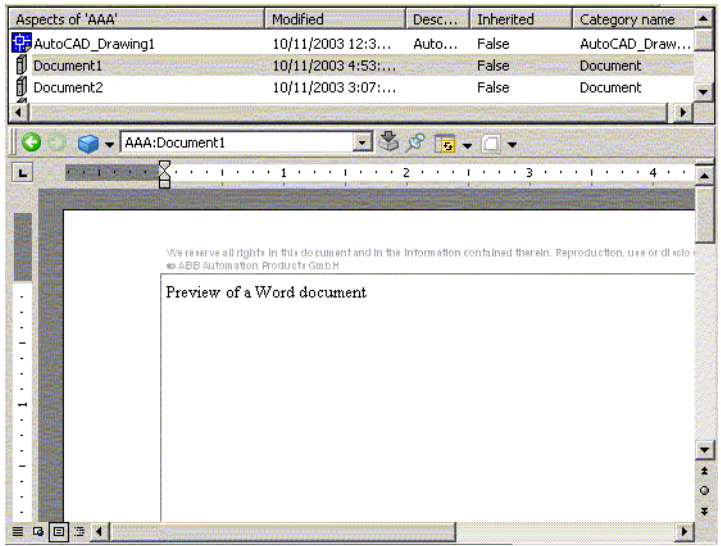


Figure 156. Open Document in Preview Window - Document View

Open a Document's Properties For Editing

A document's properties are visible in the Plant Explorer's *Preview* area as **Main View** and can be edited there (Figure 157)

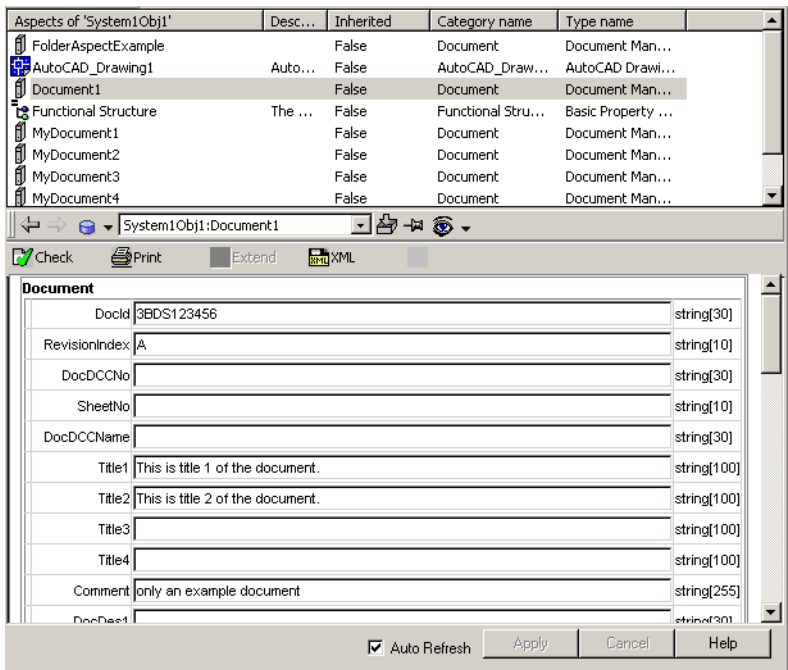


Figure 157. Open Properties in Preview Window - Main View

After modifying a property value, button **Apply** has to be pressed. To the right of each property, the property's data type and length is displayed for help.

If you want to modify DateTime properties, you can pop up a calendar by clicking on the respective symbol and select a date from there (Figure 158).

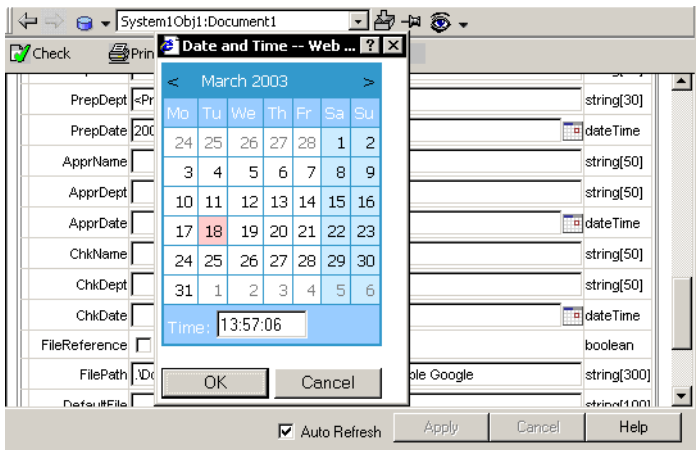


Figure 158. Set DateTime Property - Calendar

Open Multiple Documents' Properties for Editing

To open properties of multiple instances of document aspects, select context menu item **Open Properties** on a document aspect - the Data Sheet view comes up, showing properties of all document aspects of this object (Figure 159).

It is also possible to view and edit document aspect properties of a complete subtree (please see also User Interface).

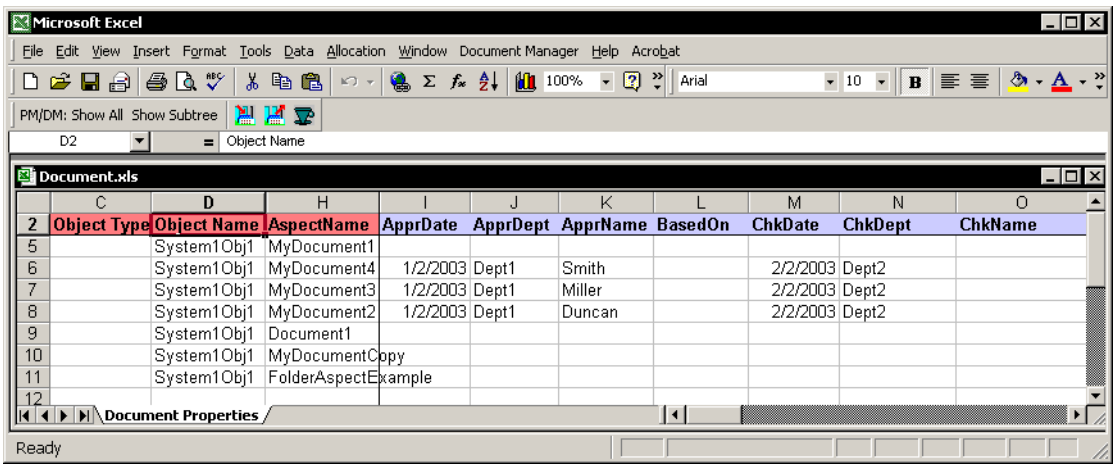


Figure 159. Open Multiple Properties in Data Sheet

You can also use Aspect Command **'Show All'** for a quick inspection of the actual state of documents concerning checked-in, checked-out, not yet set to a file or referencing a file. See Bulk Operations on a Set of Documents: 'Show All'.

The Data Sheet (opened with Open Properties) does not support list of values defined for Document properties. You have to use PED for using values out of such lists.

Print a Document Aspect

There are two possibilities for printing a document:

a. Print the document itself, i.e. the document contents, or

b. Print information about the document (administrative attributes about the document like document identity, titles, and so on), also called document properties.

For more detailed information regarding a. and b. see below.

Print Document Contents

Select the document in the Plant Explorer's ***Aspect List Area*** window and press the context menu item **Print**.

This will prompt the related application (for example Microsoft Word) to perform the Print command.

Depending on the tool associated with the document (via the file extension), the **Print** command may work or not. If the tool does not support to print a document from the Plant Explorer/Document Manager, it must be printed within the application. In this case open the document (see above) and perform the print locally in the tool

Print Document Properties

Select the document in the Plant Explorer's ***Aspect List Area*** and press the context menu item **Print Properties**.

Checkout Document

If you want to edit a document, it has to be checked out first. When selecting **Edit** for a document, this is done automatically, but the action can also be performed explicitly.

When a document is checked-out, it is fetched from it's storage location on the server and placed locally to the relative file path visible in the document aspect's property *FilePath*.

Example:

FilePath shows the relative path *.\DocumentManager\Docs\Document\XXX.doc,* so the local copy of the document is placed as *MyDocuments\DocumentParameterManager\DocumentManager\Docs\Document\XXX.doc* where *MyDocuments* refers to the physical folder *<drive>:\Documents and Settings\<user-name>\My Documents.*

A document can be checked out (and therefore be edited) only if it is not checked out by another user.

All modifications done to the document are stored only in the local copy. If user wants to make these changes visible to others, the document must be checked in.

When the document is checked out, the so-called checkout-properties (visible in the Property Editor) are depicted as in Figure 160 below.

Available only if Environment Support is not enabled.

When Environment support is enabled Checkout is handled by Reserve.

Undo Checkout

This action simply cancels the checkout. The modified local copy of the file is deleted and the respective document properties are reset.

Available only if Environment Support is not enabled.

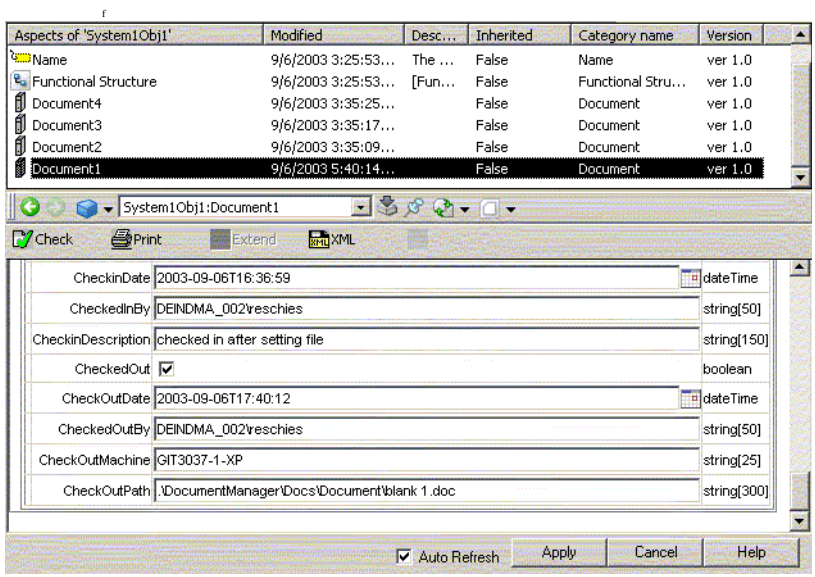


Figure 160. Checkin- and Checkout-Properties for a Document Aspect

Checkin Document

A checked-out and modified document has to be checked in to enable another user to see the modification or check it out for himself to further edit it.

The respective **Checkin** function stores the modified document in the document storage area of the server and removes the local copy of the file. The checkout properties as visible in Figure 160 are cleared and the checkin properties set with actual values.

During checkin you will be asked for a Checkin-Description. This description will be appended to a ‘history-string’ for this document to trace the checkin-history. See next item Show History.

After editing a Word- or Excel-Document you will be asked during close:

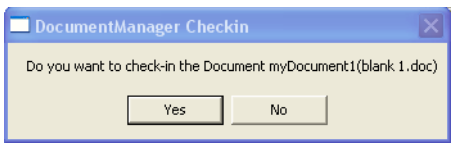


Figure 161. Automatic checkin after Exit of Word or Excel

Thus you will not forget to checkin a document after editing and make it ready for other users. This ‘automatic’ checkin can be switched off in DM-Configuration dialog see Configuration.

Only the user who has checked-out the document can check it in again!

Available only if Environment Support is not enabled.

When Environment support is enabled Checkin is handled by Release.

Show History

To show the history of a document, select the document aspect in the Plant Explorer’s *Aspect List Area* and press the context sensitive menu item **Show History**.

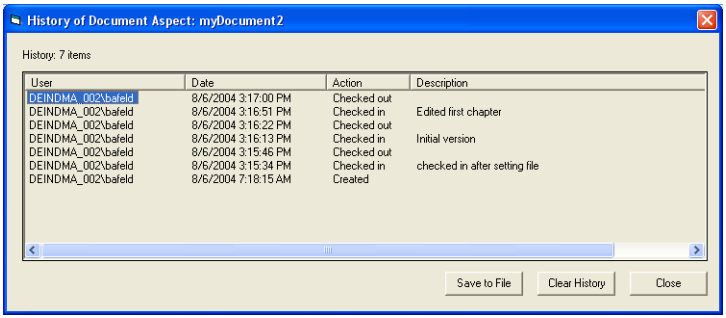


Figure 162. Show History Dialog

You have the possibility to save this history to a file or to clear all entries.The history-string is limited to 1000 characters, when this limit is reached, it will be cleared automatically.

Get Latest Checked-In Document

If for any reason you want to have a local copy of the latest checked-in version of a file, you can select **Get Latest** to do so.

Available only if Environment Support is not enabled.

Bulk Operations on a Set of Documents: ‘Show All’

Selecting the context sensitive menu item **Show All on any document aspect**, results in the following dialog.

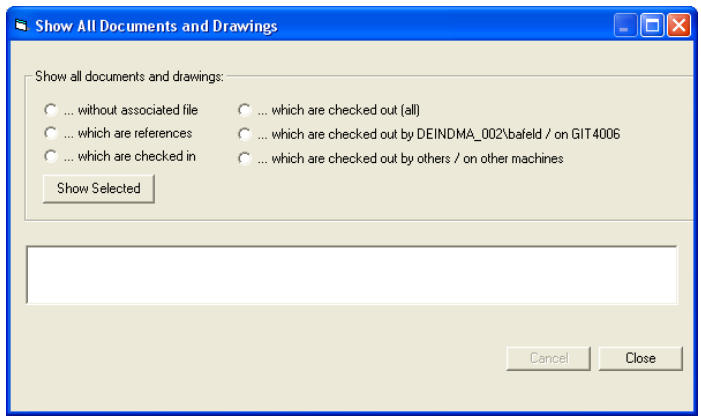


Figure 163. Show All Dialog

Now select which subset of documents you want to see and click ‘Show Selected’

Depending on the subset of documents you have chosen, you can checkin, checkout or set a file for a set of documents.

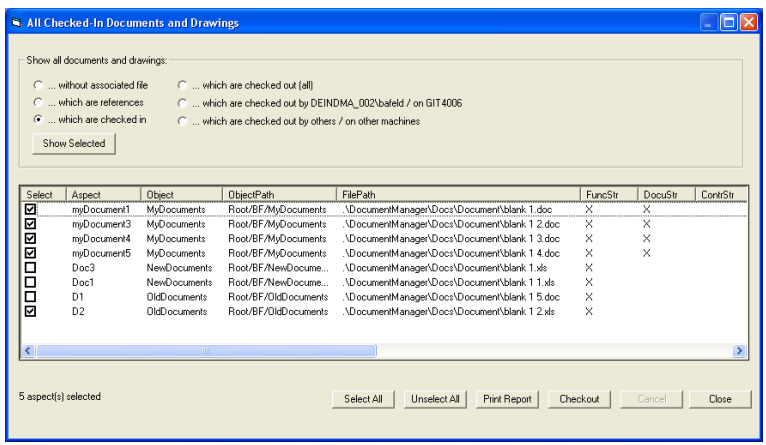


Figure 164. Show all documents, which are checked in, and checkout some of them

Override a Document Aspect

If a document aspect was created from an object type and was inherited, the *Inherited* flag is set to **true** and only one instance of this document (the one of the object type) exists in the **Document Manager.** It is also not possible to modify attribute values in *Main* view of the Plant Explorer's **Preview** area.

To change this behavior you can select **Override** from the context menu on the document aspect. The *Inherited* flag is set to **false** and a new instance is created for this document aspect in the **Document Manager**.

Export/Import Documents

Multiple documents can be

- imported with the **Bulk Data Manager** (see Section 3, Bulk Data Manager).
- exported and imported with the Plant Explorer's **Import / Export** function.

Import / Export with dependencies of instances of an object type which contains document aspect(s) of a user-defined category: Category, object type and object type instance are imported at random order, so it might happen that for example the category does not yet exist when the object type is to be imported.

To avoid problems, import the structures in the afw-file separately in the following order:

a) Aspect System Structure

b) Object Type Structure

c) <Structure where the object type instances reside>

Configuration

Right-click the **Document** aspect and select **Configure DM...** from the available context menu to display the **Configure Document Manager** dialog as in Figure 165. The following can be configured in the **Configure Document Manager** dialog:

- Write Tracefiles for all Actions.
- Update References in Word Documents before Editing

- Update References in Word, Excel, and AutoCAD Documents before Viewing
- Automatic Checkin after Exit of Word or Excel.

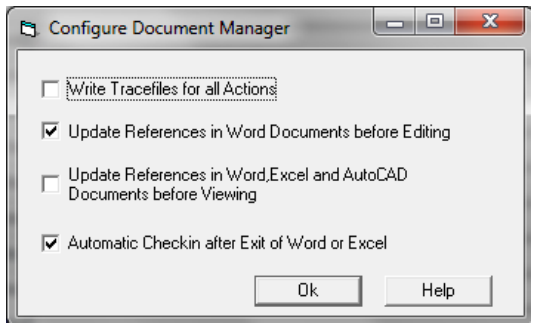


Figure 165. Configure Document Manager

Document Manager configurations are set within the scope of a client (PC specific). Other PCs might have different settings.

Configuration of Tracefile Writing

Item: Write Tracefiles for all Actions

Document Manager writes trace files for logging and error support to the user-specific folder MyDocuments\DocumentParameterManager\Tracefiles. Recorded actions are aspect specific operations, system creations or deletions and Word specific functions.

Switching off tracing improves the performance slightly but is not recommended as trace files are helpful for customer support in case of errors.

To switch off tracing, uncheck this item.

Configuration of Dynamic Document Data Update

Item: Update References in Word Documents Before Editing

The automatic update of references inserted in a Word document before opening this document for editing can be switched on or off.

The *advantage* of switching off the automatic update lies in faster execution of the **Edit** and **Print** actions for documents.

The *disadvantage* of switching off the automatic update is that some administrative attributes of a document might not be up to date.

To switch off automatic document update before editing a Word document, uncheck this item.

Item: Update References in Word, Excel, and AutoCAD Documents Before Viewing

The automatic update of references inserted in a Word, Excel, and AutoCAD documents before viewing these documents can be switched on or off.

Clear the check box, to switch off the automatic document update before viewing a Word, Excel, and AutoCAD documents.

Configuration of Automatic Checkin after Exit of Word or Excel

When a document was **opened for editing** by Word or Excel you will asked for checkin of this document after closing it. This automatic checkin of Word- or Excel- documents can be switched on or off.

Understanding Dynamic Data

What is Dynamic Data?

Basically, dynamic data consists of **property references** (sometimes also called only **references**) inserted anywhere in the document's text, header or footer and of **document links**.

References are connected to a data source and are automatically replaced with actual values.

Examples for dynamic data are:

- **Information visible in Plant Explorer,** inserted into the document via *references* (e.g. Object Name, Aspect Name or Reference Designations).
- **Information visible in Document Manager** and inserted into the document via *references* (like Title1, Author, Subject).
- **Information visible in any other aspect system** and inserted into the document via *references* (like Parameter aspect attribute values, e.g. the value of attribute BUS of a Board Parameter).

What is a Dynamic Document?

A dynamic document is a document supporting the insertion of Dynamic Data, i.e. *references.*

Which Documents Provide Dynamic Data?

- All Microsoft **Word** documents, provided **Document Manager** has been installed. Word documents offer the possibility to insert property references as well as document links.
- Microsoft **Excel** documents, provided **Bulk Data Manager Excel Add-in** has been installed (for further details regarding references in Excel documents, please refer to the Section 3, Bulk Data Manager).
- **AutoCAD** drawings, provided **AutoCAD** and **Document Manager** has been installed. **AutoCAD** drawings offer the possibility to insert property references.

Note that all Microsoft Word documents based on Document Manager templates or *copied* from files are dynamic documents - *referenced* files are dynamic only if the file itself is dynamic.

Note also that even if a Word template or a Word file to be copied is non-dynamic originally, it is made dynamic by Document Manager before opening it for modification.

How is Dynamic Data in Documents Updated?

Whenever a document is opened or printed, dynamic data inserted in this document by means of references is automatically updated from it’s original storage, provided

the update of dynamic data is activated (see Configuration). You can also trigger the update in Word documents via the menu item **Refresh References** and in AutoCAD drawings via **Refresh.**

Update of reference values in Microsoft Word and Excel and AutoCAD is bi-directional: it is possible to change values in the document and have the storage updated accordingly (for Excel please refer to the Section 3, Bulk Data Manager, for Word see Document Manager Word Functions, for AutoCAD see How to Update Property Values).

How is Dynamic Data Inserted into Documents?

Please refer to Insertion of Property References into a Document.

When to Use References?

Property references should be inserted into a document if you want to refer to aspect attribute values which might be modified in the course of the engineering process and should be updated automatically in the document.

What Happens in Case of Document Transfer?

If you copy a document to another environment, for example to your laptop for local processing, dynamic data is not effected, that is, the values written to the document during the last update still exist.

Of course, in this scenario there is no possibility to update the dynamic data in your document - the update procedure will display an error message which in this case can be ignored.

When you copy the file back to the system environment, update of dynamic data is automatically activated again. So the next time you open the document, the update will be performed correctly.

Insertion of Property References into a Document

The user interface to insert references looks slightly different in Word-, Excel-, and AutoCAD documents.

For extensions in Word documents see Working with Document Manager - Word Integration.

For extensions in Excel documents see Tutorial - Creation of Generic Dynamic Documents.

For extensions in AutoCAD documents see Working with Document Manager - AutoCAD Integration.

An overview of the common parts is described in Appendix B, Working With References.

Understanding Document Locking

When a file is opened for editing or printing, it is locked by you performing the command. This means, another user trying to open or print the same file will get a message from the respective application. Depending on the tool in use, the message looks different - please refer to the respective instruction manual for more information.

Figure 166 shows the message displayed by Microsoft Word when a document is locked by another user.

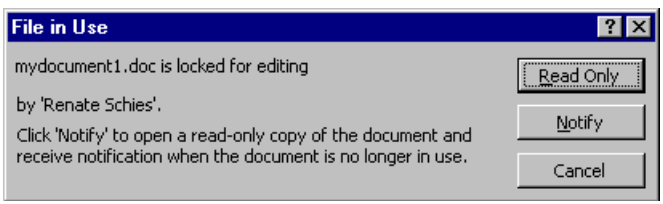


Figure 166. Word Message if Document is Locked

When Environment support is enabled corresponding Reservation for exclusive modify access is handled on aspect level.

Understanding Document Manager Templates

A template is a document with a certain layout, used as a base for other documents. Whenever you create a new document, you can select a template to base the document on. This selection is done via the selection dialog popping up after

choosing **Set File** on a document aspect *(see* Figure 153*)*.

The file that contains the template is associated with a tool based on the file extension, for example, the file extension “doc” is associated with Microsoft Word. Based on the file extension of the template, you can control which tool is to be used to work on the related document. To associate a tool with a document extension use menu item **Tools > FolderOptions > FileTypes** of the Explorer.

Document Manager is shipped with only a few Word, Excel and Autocad templates, but you can add any number of templates of your own.

Storage and Naming of Templates

Templates are stored as normal document aspects in the Library Structure of the system underneath the object *Document Manager Templates* and are further divided into *File Templates* and *Folder Templates*. These templates are available to all system users and are stored redundantly as any other document too.

When setting the file for a document aspect, the available templates are shown in a dialog from which the user can choose (see Figure 167).

Please note that only templates of a category belonging to the same type as the respective aspect can be selected as base for this aspect, e.g. for an aspect of type ‘Document Manager Documents’ and category ‘Document’ you can only choose from template aspects of the same category or any other category belonging to type ‘Document Manager Documents’.

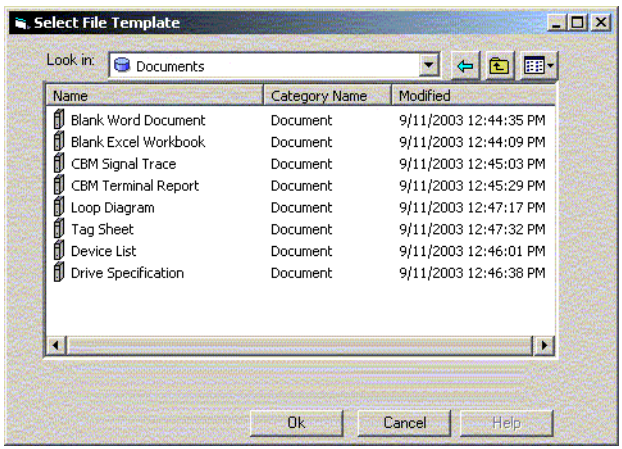


Figure 167. Template Selection Dialog

Adding, Modifying and Deleting Templates

To add a new template, create a new document with the application of your choice, e.g. Word. Style the document to your liking, e.g. by adding frames, headers and footers or by including predefined chapters. In headers and footers, as well as in normal text, you can insert property references to show dynamic data (for Word, Excel and Autocad only). Save and close the document.

In the Library structure, add a new document aspect with a meaningful name to the respective object, e.g. ‘Documents’. Select menu item **Set File** and in the resulting dialog choose **From File/Folder** and **Copy File** to copy your document. Your document is now ready to be used as a template.

To delete a template simply delete the respective aspect in the Library Structure.

To modify a template you proceed just as with any normal document aspect, i.e. the aspect has to be checked out before you can edit the file and checked in to be made available to other users.

To exchange templates between systems simply copy the respective template aspects to the destination system or use the Import / Export functionality.

Existing Predefined Templates

In addition to the blank Word- and Excel-Templates you can use some special templates such as:

- Loop Diagram - while using this template, set Document aspect name as **LoopDiagram**
- Tag Sheet
- Device List
- Drive Specification

The corresponding documentation for these templates is available in the desktop folder '**Engineering Templates**' within the '**_Documentation**' folder.

Refer to Appendix E, Word Report Templates to learn about the Word Report Templates.

Working with Document Manager - Word Integration

Insertion of Property References into a Document

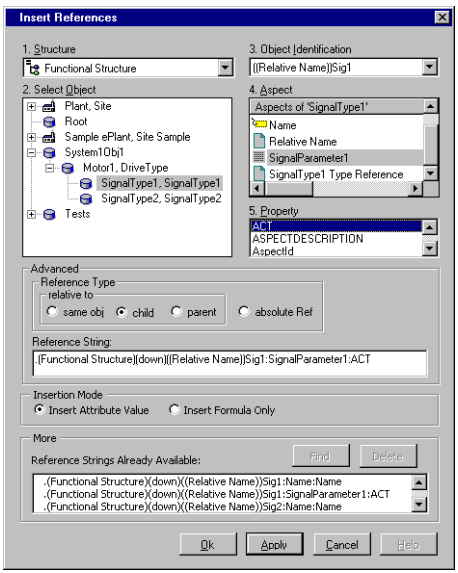


Figure 168. Word User Interface Insert References Dialog

Available Reference Strings (Insert, Find, Delete)

References already inserted into the document are visible as reference strings in the list box headed **Reference Strings Already Available** (see Figure 168). By clicking on a available string, this string is copied to the *Reference String* field and can then be edited by hand or immediately inserted by pressing **Apply**.

The **Find** button is meant to be used to find out the usage of one of these references in the document. Press the button which then offers the possibility to Find First/ Find Next occurrence of this string.

To be able to delete one of these references throughout the document, select the reference string in list box and press **Delete** button. The reference string itself and all occurrences of this string, that is, all inserted values are deleted.

Insert Formula Only

Normally, when inserting an aspect attribute by formulating a reference string, the value of the aspect attribute is retrieved immediately and inserted at cursor point (mode ***Insert Attribute Value***). If the value cannot be retrieved, for example, because the aspect described does not exist, nothing is inserted and an error message appears. To provide the possibility to prepare documents in advance, it is possible to insert property references in a so-called ***Insert Formula Only*** mode by checking the ***Insert Formula Only*** option (see Figure 168).

Even if no actual value could be retrieved, the reference string is inserted into the document to be re-evaluated at a later stage when the value is possibly available. As default behavior the ***Insert Attribute Value*** mode is checked.

Update Source Value of References

Inserted property references are bi-directional, i.e. the source value in the data source can be updated by updating the property reference inserted in a document.

In Excel, this is done by simply updating the value visible in a cell.

In Word. the reference (or multiple references) have to be selected.After selecting menu item ***Update Source Value(s),*** **a** dialog appears in which the new value can be entered.

Document Manager Word Functions

To activate special Document Manager functions in Word, select menu **Document Manager**. The identical menu items will be presented in the context menu by clicking the right mouse button (see Figure 169).

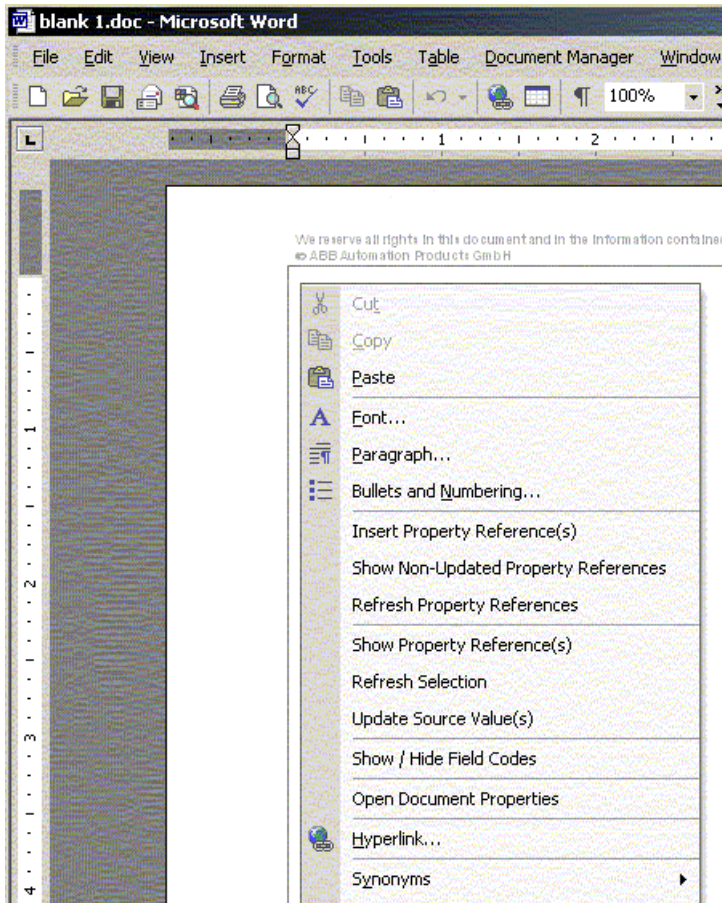


Figure 169. Word Document Manager Context Menu

Functions presented in the menu items list are:

- **Insert Property Reference(s)**
  For insertion of property references, showing values of aspect attributes, see Insertion of Property References into a Document.
- **Refresh Property References**
  When opening a document aspect from Plant Explorer, the document's inserted

property references are updated automatically during the open action (provided the respective option is set - see Configuration). This refresh action can also be executed when working with the document itself by selecting item **Refresh Attributes/Document Links**.

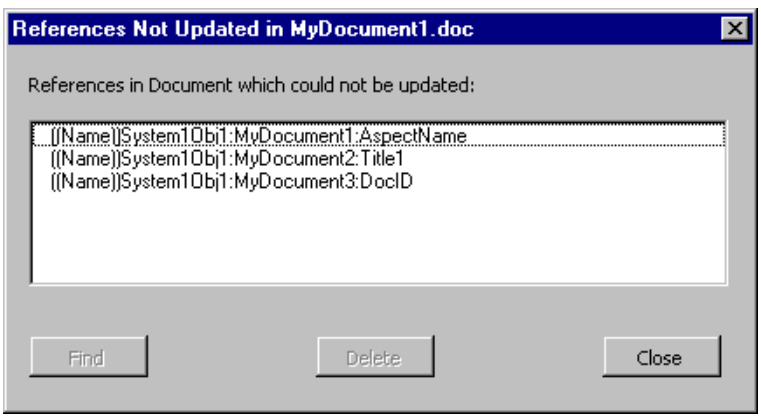


Figure 170. Example for Non-Updated References

- **Show Non-Updated Property References**
  When opening a document containing dynamic data in the form of inserted property references, this data is updated automatically by retrieving values. As inserted property references might refer to aspects no longer existent, it is not possible to update their value. Select item **Show Non-Updated References** to check for un-updated respectively un-retrievable references (for an example, see figure above).
  You can also select one of the non-updated references and by pressing either button **Find** or button **Delete** search for any instances of this reference in the document or delete all instances.
- **Show Property Reference(s)**
- When inserting a reference in a Word document, this is done by inserting a Word *field*. Visible is only the actual value of the inserted property reference, but not the field code itself. This function offers the possibility to select more

than one field and see the field codes behind the values, see Figure 171

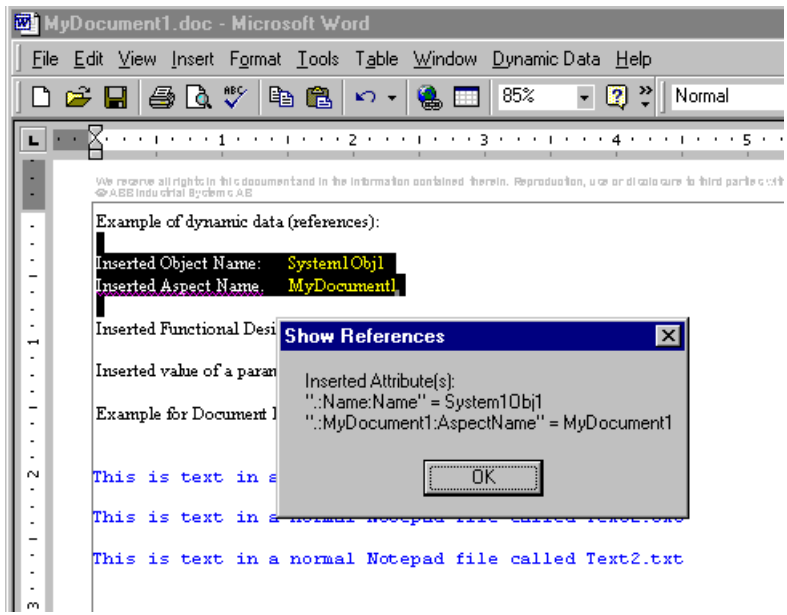


Figure 171. Show Property References Dialog

- **Update Source Value(s)**

  This dialog enables you to update the source value of an inserted reference. After selecting one or more references, the dialog is displayed for each one and a new value can be entered (see Figure 172).

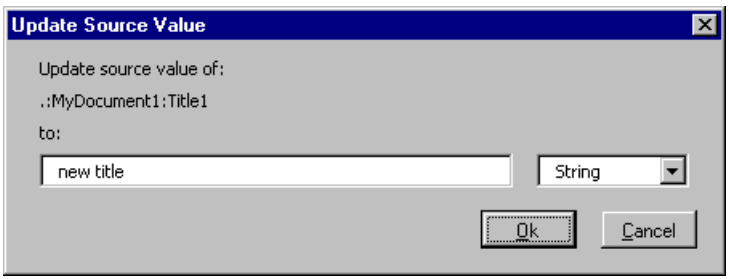


Figure 172. Update Source Value Dialog

- **Show/Hide Field Codes**
  Property references.and document links are inserted into a document by using the *field object* available in Word. This field object is normally not visible. To show all inserted fields in a document, select item **Show/Hide Filed Codes** and vice versa (toggle function - example see example below).
- **Open Document Properties**

  To open the Document Manager Data Sheet, see Open Multiple Documents' Properties for Editing.
- **Aspects**
  To open an Aspect List window, provided an inserted reference is properly selected. The Aspect List window then shows all aspects of the object belonging to the inserted reference and also offers the possibility to open them.

Document Versioning

Though Document Manager itself does not offer the possibility to freeze a document in a version, you can take advantage of the built-in Word option to create versions of a document.

This option is activated by selecting menu item **File > Versions** and results in the dialog depicted in Figure 173.

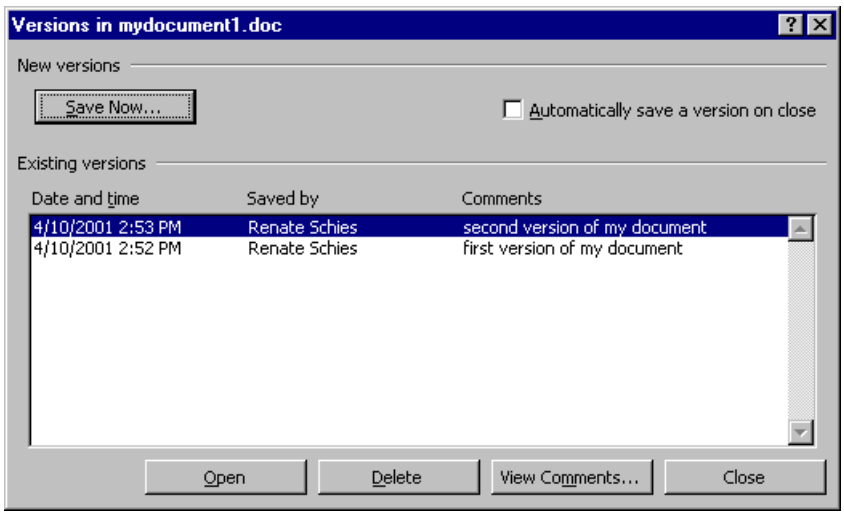


Figure 173. Word Versions Dialog

By pressing the **Save Now** button you can create a new version of the document which you can later view again by selecting it and pressing **Open**.

Please note that all versions of a document are kept in one file.

Please also note that references inserted into the document keep their original value in document versions. This provides an excellent possibility to create a kind of ‘release version’ of a document.

Working with Document Manager - AutoCAD Integration

About AutoCAD Integration

Ensure that the AutoCAD tool is opened prior to any AutoCAD integration operations.

Document Manager - AutoCAD Integration extends the **Document Manager** to deal with dynamic drawing documents of file format DWG, based on AutoCAD.

Supported Versions are AutoCAD 2004, AutoCAD 2005 and AutoCAD 2006.

Viewing support is supplied for format DWF and DXF. As with all documents managed by the Document Manager all functions to administrate documents are also available on drawings. (Document Aspects)

Drawings are supplied with aspect type **AutoCAD Drawings** and the default category **AutoCAD_Drawing**.

Existing DWG-files can be imported and made dynamic as well as new drawings can be created and drawn from scratch either using pure AutoCAD or an AutoCAD based CAE tool (How to Specify the AutoCAD Startup Behavior).

As most AutoCAD based Applications deal with blocks and block attributes it is possible to create property references on this type of drawing objects. However some drawings have relevant information assigned to TEXT or MTEXT (multi line text) objects. Thus also text can be made dynamic by use of property references. (Understanding Dynamic Data and Property Reference Dialog)

When opening or printing drawings it might be preferable to automatically refresh property references or start subscription. This and the behavior in cases of unresolved property references can be specified per machine. (How to Specify Property Reference Behavior)

Printing/Plotting in AutoCAD is a complex issue. For each drawing format the way of printing (rotation, positioning, pen assignment etc.) and the physical printer may differ. Furthermore it is sometimes necessary to produce a file based documentation e.g. in PDF for which a PDF based printer driver is necessary. All this kind of configuration can be predefined per drawing format. (How to Specify Plotting Behavior)

Once property references are specified it is desirable to have the possibility to navigate to other aspects or aspect objects which are related to the aspects and aspect objects addressed within the drawing. Using the standard navigation views of the Engineering Workplace this can be done starting from any drawing object having a property reference. (See How to Navigate.)

In cases where a drawing is supplied only in DWF or DXF format and in cases where AutoCAD is not installed on the current machine it is possible to view, open and print drawings using any (third party) viewers. Some of the suggested viewers are mentioned below along with the url's to downloads.

- DWGTrueView (http://www.autodesk.com/dwgtrueview)
- DWFViewer (http://www.autodesk.com/dwfviewer)
- DWGViewer (http://www.cimmetry.com/dwgviewer.html)
- FreeDWGViewer (http://www.infograph.com/products/dwgviewer/DWGdownload1.asp)

In cases where a drawing is only available on the internet in DWF format it is possible to access such a drawing using a URL provided the Autodesk Express Viewer plug-in for the Internet Explorer is installed.

If you intend to use AutoCAD in combination with **Document Manager** to deal with dynamic drawing documents it is recommendable to startup AutoCAD in front of operation, and if you leave drawings not to Exit but rather to Close for a smoother working rhythm.

User Interface

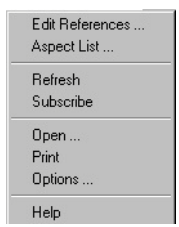


Figure 174. AutoCAD Integration - Document Manager Menu

- Edit References
  Starts the Property References dialog.
  For details see Property Reference Dialog.
- Aspect List
  Starts the Aspect List dialog.

For details see Aspect List Dialog.

- Refresh
  All property references will be resolved and the respective values within the drawing will be updated.
  For details see How to Refresh Property References.
- Subscribe
  All property references will be resolved and the respective values within the drawing will be updated. Additionally subscription for the property values which are referenced will be started.
  For details see How to Subscribe Property References.
- Open
  Starts the Open Drawing dialog.
  For details see Open Drawing Dialog.
- Print
  Starts printing the current drawing.
  For details see How to Print a Drawing.
- Options
  Start the Options dialog.
  For details see Options Dialog.
- **Help**
  Shows the Document Manager - AutoCAD Integration help.

Options Dialog

Using the Options Dialog, configurations can be made for the AutoCAD integration, per client computer. Subject of configuration is the startup behavior of AutoCAD, plotter configurations and property reference behavior.

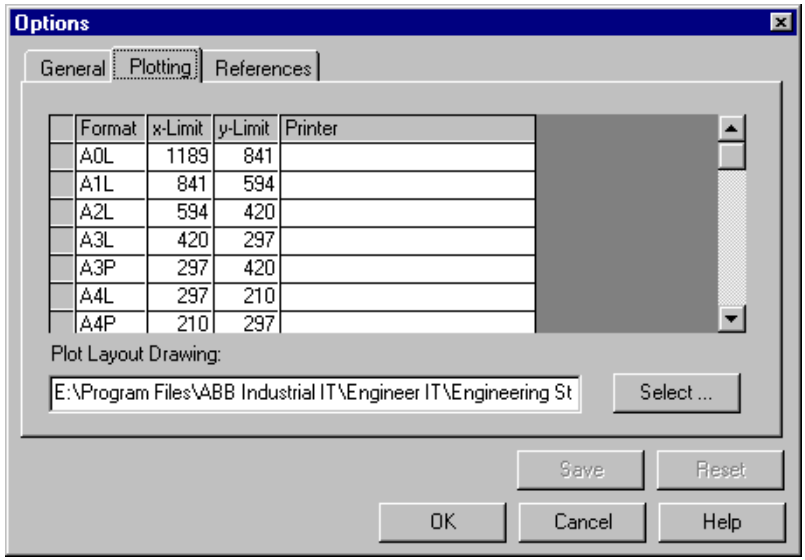


Figure 175. AutoCAD Integration - Options Dialog

- **Saving Options**
  When changing a setting within the Options dialog the Save button on the bottom of the dialog will be enabled. As long this button or the OK button will not be pressed the changes are not permanently stored.
- **Resetting Options**
  When changing a setting within the Options dialog the Reset button on the bottom of the dialog will be enabled. If this button or the Cancel button will be pressed the changes will not be stored.

  Moreover, when pressing the Reset button the dialog keeps open and all settings that were changed since the last save or reset will be set to the value before the change.

How to Specify the AutoCAD Startup Behavior

The General Options Tab

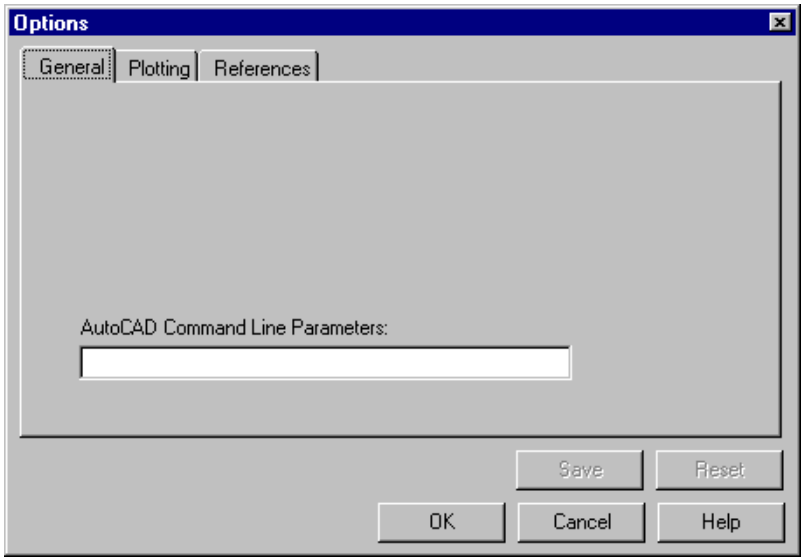


Figure 176. AutoCAD Integration - Options Dialog - General Tab

- AutoCAD Command Line Parameters
  With Document Manager it is possible to not only integrate pure AutoCAD but also applications which are developed on top of AutoCAD.
  Most of this applications provide an own profile and/or a different hardware configuration folder then the default. For that purpose it is possible to start AutoCAD with some command line parameters in order to specify e.g. a different profile that should be used. For other possible AutoCAD command line parameters see the AutoCAD online documentation.

To find out which command line parameters should be set for a specific AutoCAD based Application look for a shortcut on the desktop or the Start menu. At the Shortcut tab of the Properties dialog of that shortcut you will find how AutoCAD should be called.

If a profile is specified with the command line this will be additionally used to identify the correct running AutoCAD instance when opening or printing drawings. When working with an AutoCAD based application (probably based on some ARX-

extension) not any running AutoCAD instance can be used but only this which is running with the correct profile.

How to Specify Plotting Behavior

The Plotting Options Tab

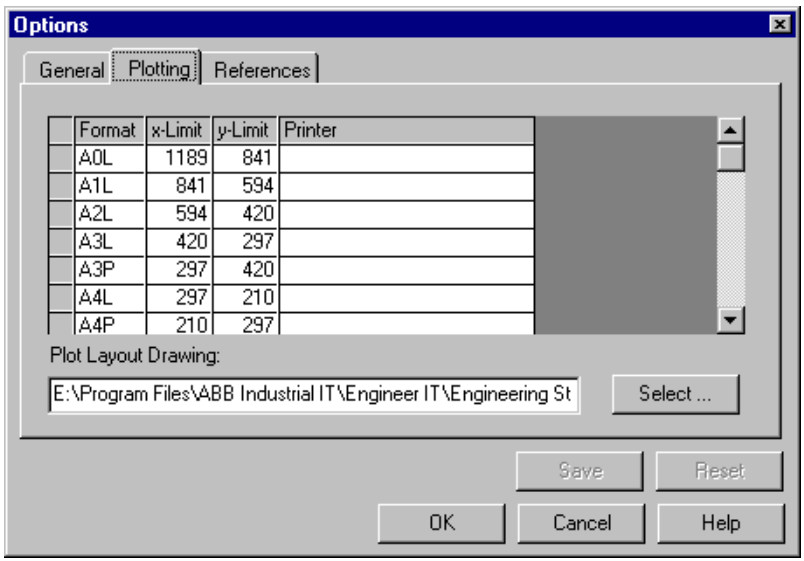


Figure 177. AutoCAD Integration - Options Dialog - Plotting Tab

- Format List
  The format list is a grid containing settings about the configured drawing formats organized by their limits. Each format is specified by a name (column Format), by the y- and y-limits (columns X-Limit and Y-Limit) and by a printer name (column Printer).
  The form name can be any name that best describes the format to be specified. There are no limitations to the name except that the names cannot be empty and must be unique.
  The x- and y-limits describe the formats x- and y-extensions. This can be any valid AutoCAD limits.
  The printer column can be either empty or can be the name of any installed

printer. If the name is empty the systems default printer will be used. In the name is not empty the printer specified by the name will be used.

On how to manipulate formats see

- How to Add a New Format Definition,
- How to Change an Existing Format Definition and
- How to Remove a Format Definition.

- Plot Layout Drawing
  The plot layout drawing is a separate drawing located as described in the Plot Layout Drawing text field.
  You may use the delivered plot layout drawing (esacad.dwg) or create your own or eventually create more then one plot layout drawing for different purposes. By changing the path in the text box it is possible to work with a different file.
  The plot layout drawing consists of a set of plot layouts which can be done in AutoCAD by selecting menu item **File > Page Setup ...**. Plot layout will be used by the AutoCAD integration to specify the plot behavior of a format specified in the format list. The names of the format list and the names of the plot layout must be the same.
  When calling print from either the Engineering Workplace, the Document Manager application or from the Document Manager menu inside AutoCAD the print function will detect first the limits and tries to find an adequate format from the list. When found the format, the plot layout drawing will be used to get the plotter configuration and the specified printer in the format list will be used to locate the printer.
  On how to manipulate plotting layouts see
  - How to Create or Change a Plot Layout and

How to Add a New Format Definition

To add a new format definition an entry must be made within the last row of the format list. The row is marked with an asterix ('*').

At least a format name must be specified when saving the changes. As long the changes should not be saved all columns can be empty.

How to Change an Existing Format Definition

To change an existing format the respective cell must be selected. When double clicking on the cell an edit box appears that allows to make changes.

Within the limits cells it is only allowed to type in valid numbers.

How to Remove a Format Definition

Removing a format definition can be done by just empty all cells in the row. After the next refresh of the list the row disappears.

How to Specify Property Reference Behavior

The References Options Tab

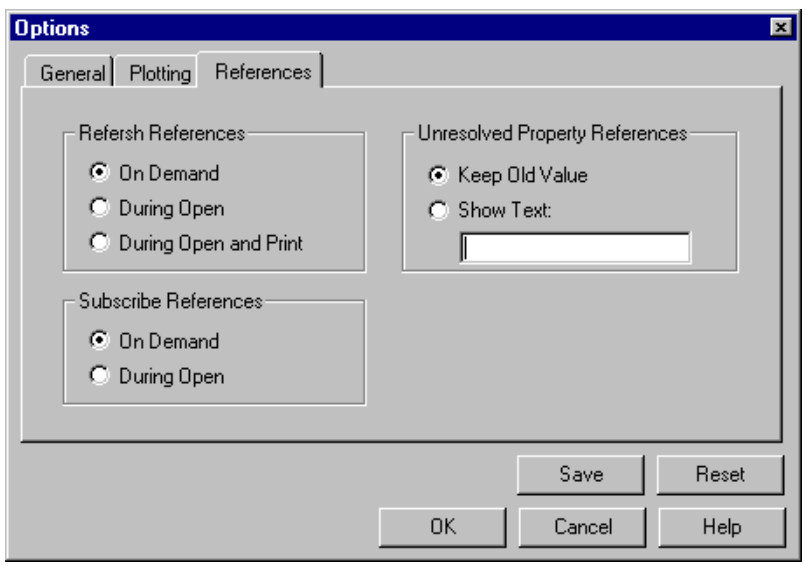


Figure 178. AutoCAD Integration - Options Dialog - References Tab

- Refresh References
  When having specified property references there are different situations when the associated values can be refreshed.

  - On Demand
    Specifying this option the property references will not be automatically refreshed.

    You need to refresh as described in How to Refresh Property References.

    However if Subscribe References During Open is enabled this will also affect the refresh behavior.
  - During Open
    Specifying this option the property references will be automatically refreshed during open. This options has no effect if the drawing is read only.
  - During Open and Print
    Specifying this option the property references will be automatically refreshed during open and print. This options has no effect if the drawing is read only.

- Subscribe References
  When having specified property references there are different situations when the associated values can be subscribed. When turning subscription on, first all property references will be refreshed.
  - On Demand
    Specifying this option, subscription of the values referenced by property references will not be automatically turned on.

    You need to turn on subscription as described in How to Subscribe Property References.
  - During Open
    Specifying this option, subscription of the values referenced by property references will be automatically turned on during open.
- Unresolved Property References
  It may happen that some of the property references cannot be resolved during refresh. Tis might be because of a syntax error in the references, an invalid property or because of the fat that a referenced aspect does not yet or does not anymore exist. It is possible to configure the behavior in this situations.

– Keep old Value
When turning on this option, for each value that cannot be refreshed due to an unresolved property reference, the value within the drawing will not be updated. The value will as it was before starting the refresh action.

– Show Text
When turning on this option, for each value that cannot be refreshed due to an unresolved property reference, the value within the drawing will be replaced by a text that can be specified in the text box. (The default is ##).

Property Reference Dialog

The **Property References** Dialog, is a modeless dialog that can be used for

- viewing existing property references,
- creating new property references,
- changing property references,
- updating property values which are referred by property references and
- finding unresolved property references.

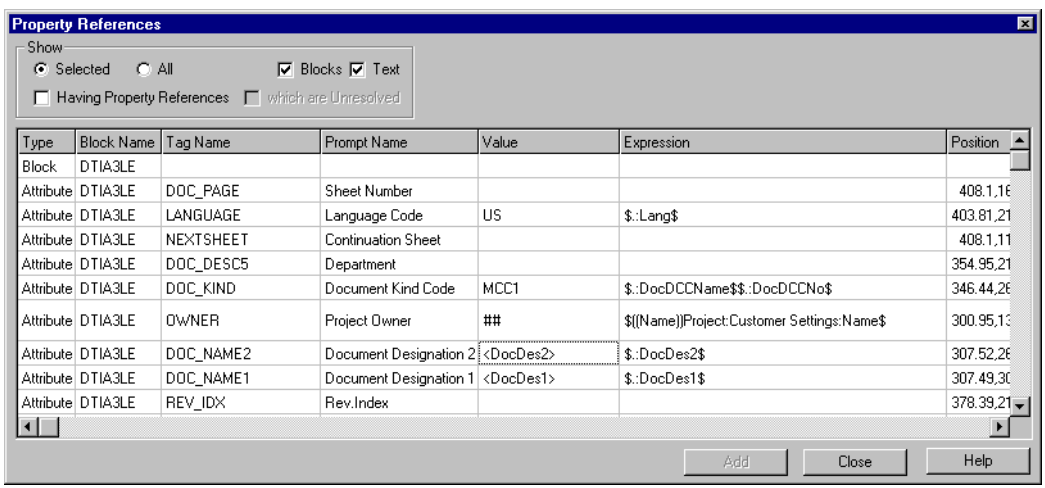


Figure 179. AutoCAD Integration - Property References Dialog - Left Columns

The **Property Reference** dialog has the columns

- Type
  This column describes the drawing object. Possible values are Block for AutoCAD block references, Attribute for block attributes, Text for TEXT objects and Multiline Text for MTEXT objects.
  This columns cannot be edited.
- Block Name
  This column contains the block name in cases where the object type is Block or Attribute. For Text and Multiline Text the column is empty.
  This columns cannot be edited.
- Tag Name
  This column contains the tag name (i.e. the internal attribute name) of an block attribute in cases where the object type is Attribute. For Block, Text and Multiline Text the column is empty.
  This columns cannot be edited.
- Prompt Name
  This column contains the prompt name (i.e. the name used in dialogs for editing attributes) of an block attribute in cases where the object type is Attribute. For Block, Text and Multiline Text the column is empty.
  This columns cannot be edited.
- Value
  This column contains the value of the respective drawing object. In case of object type Block this value does not have a representation in AutoCAD. In the other case this values are associated to the corresponding AutoCAD values.
  This column can be edited as long there is no Property reference specified that cannot be resolved. On how to edit this column see How to Update Property Values.
- Expression
  This column contains the definition of one ore more property references and static text. If the expression will be evaluated the result will be shown within the Value column. In cases of errors when resolving property references the Message column will contain an error message describing the reason why the reference cannot be resolved.

This columns can be edited for all mentioned types.
In case of a Multiline Text the text box is a multiline text box.

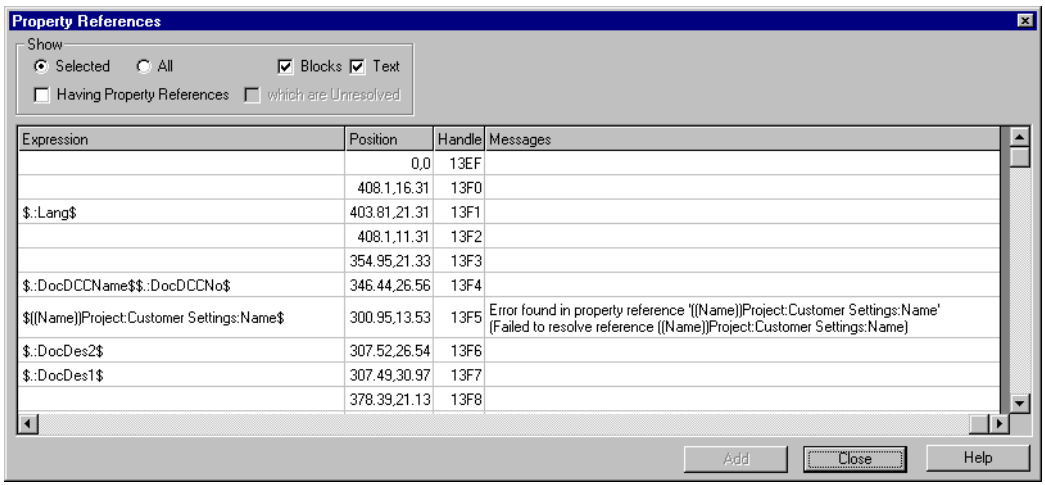


Figure 180. AutoCAD Integration - Property References Dialog - Right Columns

- Position
  This column shows the x-y-coordinates of the object.
  This columns cannot be edited.
- **Handle**
  This column shows the handle of the object.
  This columns cannot be edited.
- Message
  This column shows one or more error messages in cases where the expression contains error or a property reference could not be resolved for some other reason.
  This columns can be edited. The main purpose of this is to be able to cut and paste the message for error reporting.

How to Select Drawing Objects

When working with the **Property References** dialog there are several possibilities to select the drawing objects that should be shown within the dialog.

First of all it is possible to select objects before opening the dialog or when the dialog is displayed. Furthermore its is possible to select a single object or all object that are enclosed within a rectangle. Finally all drawing object can be selected by selecting the respective option within the dialog. When selecting a block object all of its attributes will be shown as well.

- **Selecting Single Objects**
  When selecting a single object and add another object by pressing the Ctrl-Key only the last selected object will be taken.
- **Selecting Rectangles**
  Selecting via a rectangle is done by pressing the left mouse button and moving the cursor downward right (respectively upward left) and then pressing again the left mouse button. This selects all object which are fully enclose in this rectangle (respectively all objects that are included in the rectangle and also those objects that are intersected by the rectangle).
- **Selection Before Dialog is Displayed**
  One of the methods described above can be used to select the objects which shall be shown within the dialog. When starting the dialog the selected object will be shown.
- **Selection When Dialog is Displayed**
  One of the methods described above can be used to select the objects which shall be shown within the dialog. The content of the dialog changes immediately when finishing the selection.

How to use Filters

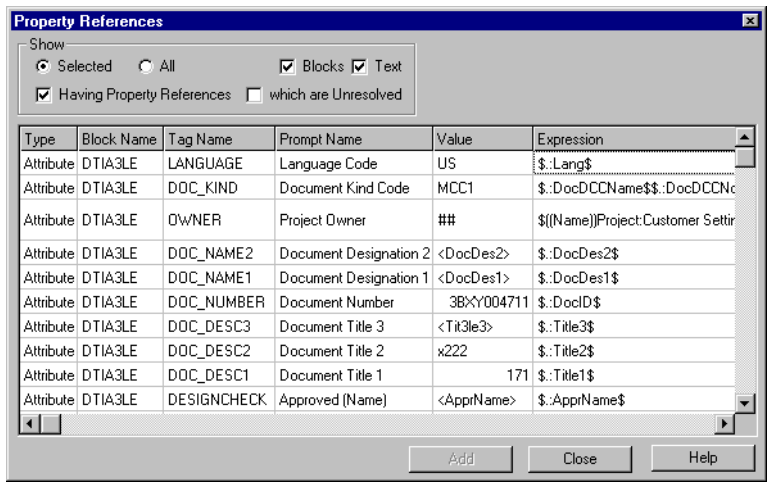


Figure 181. AutoCAD Integration - Property References Dialog - Filter

Several filters can be used to restrict the set of drawing objects within the dialog.

- **Show All vs. Show Selection**
  As described in How to Select Drawing Objects it is possible to select some drawing objects or to show all drawing objects within the drawing. This can be done by use of the radio buttons **Selected** or **All** in the **Show** section.
- **Show Blocks and/or Show Text**
  When the objects within the dialog should be additionally restricted to objects of type **Block** and **Attribute** respectively to drawing object of type **Text** or **Multiline Text** this can be done by checking the check boxes **Block** and **Text** in the **Show** section.
  Unchecking both check boxes will show an empty list.
- **Show Objects Having Property References**
  When checking the check box **Having Property References** only those rows will be shown that show object having a defined expression. In this case it might happen that there are attributes shown but the owning block is not shown.

After checked this check box another check box will be enabled whose name is **which are Unresolved**.

- **Show Objects Having Property References which are Unresolved**
  When checking the check box **which are Unresolved** in the **Show** section only those rows will be shown that represent objects having a defined expression that is unresolved.

Understanding Expressions

Expressions which can be specified within the **Expression** column of the **Property References** dialog can be used to specify values by using property references and VBScript sub expressions.

The syntax is given as follows:

EXP = VBS_TERM$_{opt}$ ... **VBS_TERM**$_{opt}$ .... **REF_TERM**$_{opt}$ **REF_TERM**$_{opt}$ ....

An expression **EXP** can be any sequence consisting of property reference terms and VB Script terms.

REF_TERM = TEXT$_{opt}$ **REF_EXP**$_{opt}$ **TEXT**$_{opt}$ .... **REF_EXP**$_{opt}$ **TEXT**$_{opt}$

A reference term **REF_TERM** can be any sequence consisting of text and reference expressions

VBS_TERM = <*vbs*> **VBS_EXP**$_{opt}$ </*vbs*>

A VBScript term is build up by a starting <vbs> tag and a closing </vbs> tag. Between this tags any valid VBScript expression can be written.

TEXT = any text

REF_EXP = *$* **PROP_REF** *$*

PROP_REF = any valid property reference

VBS_EXP = any VBScript expression which can make additionally use of the object **dm**

See the following examples for a principle categorization of expressions.

Examples

- Example 1 (Static text)
  The example consists of some static text only.
  > *Some Static Text*
- **Example 2 (Using property reference)**
  The example consists of a property reference (enclosed by '$' characters) that addresses the Functional Reference Designation of the object where the current aspect belongs to.
  > *$.:Functional Structure:ARD_DEFAULT$*
- **Example 3 (Using static text with property reference)**
  The example consists of a property reference (enclosed by '$' characters) that addresses the Functional Designation of the object denoted by PCS which is proceeded by a '-' character. This is a mixed form combining text and property references.
  > *-$.(Functional Structure)(down)PCS:Functional Designation:Name$*
- **Example 4 (Using more than one property reference)**
  The example consists of two property references (enclosed by '$' characters) that address two distinct properties (defining the document kind classification code) of the current drawing aspect.
  $.:DocDCCName$$.:DocDCCNo$
- **Example 5 (Using VB Script expression)**
  The example consists of a property references (enclosed by '$' characters) that addresses the preparation date of the drawing aspect. As this property contains the date and the time of the creation and the date is neither IEC nor DIN conform it is necessary to change the date format appropriate. This can be done using a VB Script expression.
  Use
  > *<vbs>dm.FormatDate("$.:PrepDate$", "MM/DD/YY") </vbs>*
  for IEC conform dates and use
  > *<vbs>dm.FormatDate("$.:PrepDate$", "DD.MM.YY") </vbs>*
  for DIN conform dates.
- **Example 6 (Using Term in VB Script expression)**
  <vbs>"Date:" & dm.FormatDate("$.:PrepDate$","YY-MM-DD")</vbs>

As only VBScript expressions can be evaluated, only functions, operators and constants can be used.

Go to **http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vtoriVBScript.asp** for a detailed description on how to write VBScript expressions.Within VBScript expressions all function for the object with name **dm** can be used. Currently the following functions are available:

- Function **FormatDate**(*expression*, *format*)

  The **FormatDate** function has the following parameters
  expression:Any valid string expression.
  format: A valid name or user defined-format

  The function takes the string **expression** and formats it according the format specified in the **format** parameter.

  See the example above for how to use the **FormatDate** function of the object **dm**.
  For a detailed description on what formats are valid go to **http://msdn.microsoft.com/library/en-us/vbenlr98/html/vafctFormat.asp?frame=true** and see the VisualBasic **Format** function.

- Function **Matches**(text,pattern)

  The **Matches** function has the following parameters
  text: Any string.
  pattern: Any string which can additionally contain wildcards

  Wildcards
  ? - Any single character.
  ***** - Zero or more characters.
  # - Any single digit (0–9).
  [*charlist*] - Any single character in *charlist*.
  [**!***charlist*] - Any single character not in *charlist*.

  The **Matches** function takes the **text** parameter and checks wether it is like the **pattern** parameter.

For more details go to **http://msdn.microsoft.com/library/en-us/vbenlr98/html/vaoprlike.asp?frame=true** to get a description of Visual Basic like operator.

- Function **IfThenElse**(conditiona,if-case,else-case)

  The **IfThenElse** function has the following parameters
  condition: Any valid boolean expression.
  if-case: Any valid expression of either type
  else-case: Any valid expression of either type

  The function **IfThenElse** evaluates the condition and return the **if-case** expression if the condition is true and returns the **else-case** expression if the condition is false.

  For more details go to **http://msdn.microsoft.com/library/en-us/vbenlr98/html/vafctIIF.asp?frame=true** to get a description of Visual Basic **If** function.

How to Add and Change Property References

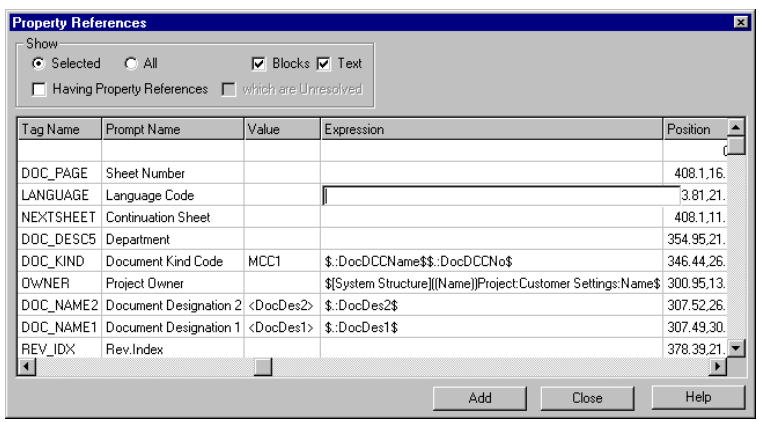


Figure 182. AutoCAD Integration - Add and Change Property References

Adding property references can be done for each row in the dialog.

When double clicking on the cell in the Expression column the cell changes to a text box (which is a multiline text box in case of a Multiline Text object). At the same time the **Add** button on the bottom of the dialog will be enabled.

In principle there are two methods to specify an expression. The expression can be either edited manually and by using copy/paste operations or it can be edited by use of the **Insert Property Reference** dialog. The **Insert Property Reference** dialog will be opened using the **Add** button. How to work with the **Insert Property Reference** dialog is described in Insertion of Property References into a Document.

Expressions make use of property references in that way that a property reference can occur within an expression as a part that is enclosed in **'$'** characters. There are several types of expressions starting from pure static text to complex expressions using VB Script.

For more details on expressions see Understanding Expressions.

Insert Property References Dialog

The dialog Insert Property References can be used to build up a property reference just by navigating throw a structure and selecting an aspects property.

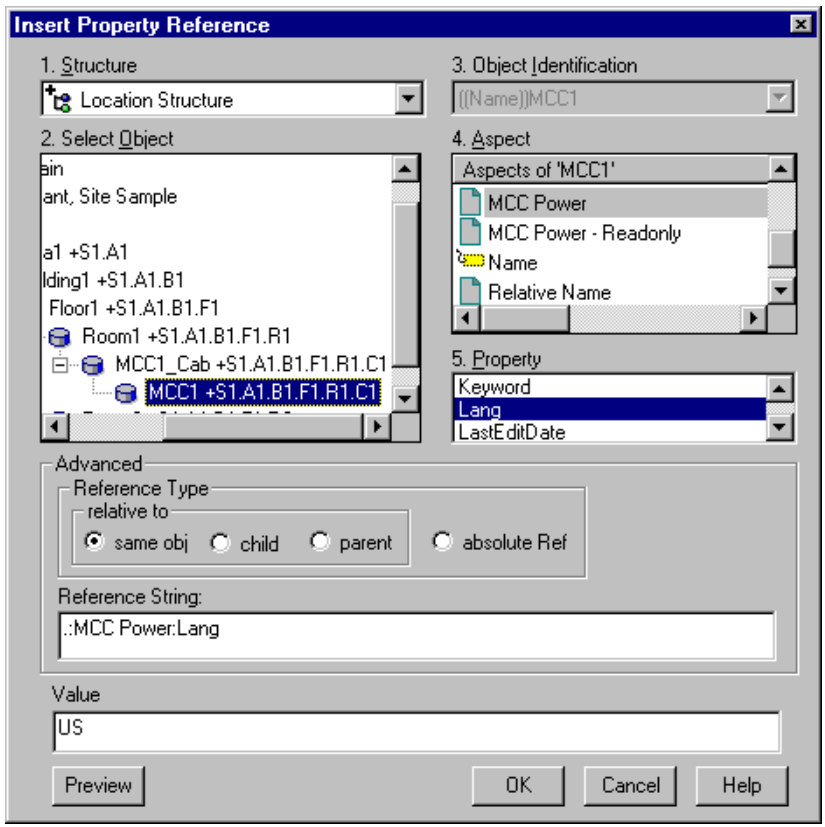


Figure 183. AutoCAD Integration - Insert Property Reference dialog

The details of the dialog are described in Insertion of Property References into a Document. However there are some additions which will be described below.

- **Previewing the Selected Property Reference**
  When having selected a valid property reference with the dialog it is possible to preview the value of the referenced property by pressing at the Preview button. In the case the reference cannot be resolved the preview text field is empty.
- **Accepting the Selected Property Reference**
  When pressing the **OK** button the reference will be inserted at the current

cursor position within the active text box of the grid. In addition the property reference selected by the dialog will be enclosed by '$' characters.

- **Ignoring the Selected Property Reference**
  When pressing the **Cancel** button the reference will not be inserted within the active text box of the grid.

How to Update Property Values

If for a drawing object a property reference is defined it is possible to change the value within AutoCAD and update the value of the referenced property. This can be done within the **Property Reference** dialog.

The editing of values is different depending on the expression defined includes one or more property references.

If one property reference is included and no other static text the value will be edited within the value cell of the grid.

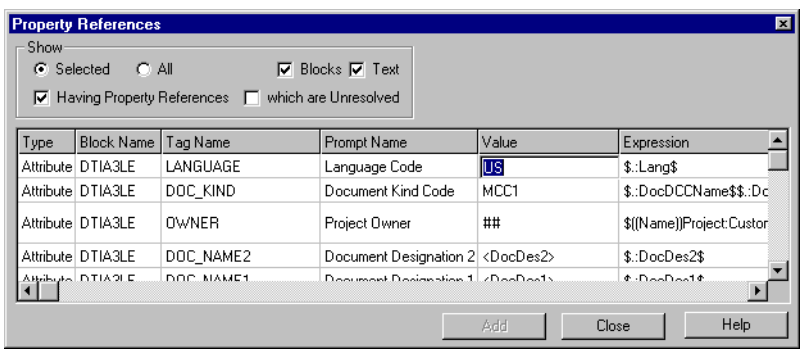


Figure 184. AutoCAD Integration - Update Value

The example in Figure 184 shows how to edit the value if the expressions is e.g.

$.:Lang$

If one property reference is defined within the expression together with some static text or two property references are include in the expression then the value within the grid cannot be edited directly. Instead of that a small sub-grid appears containing

a row for each defined property reference included in the expression. Within this sub-grid it is possible to edit the values associated with the respective property references.

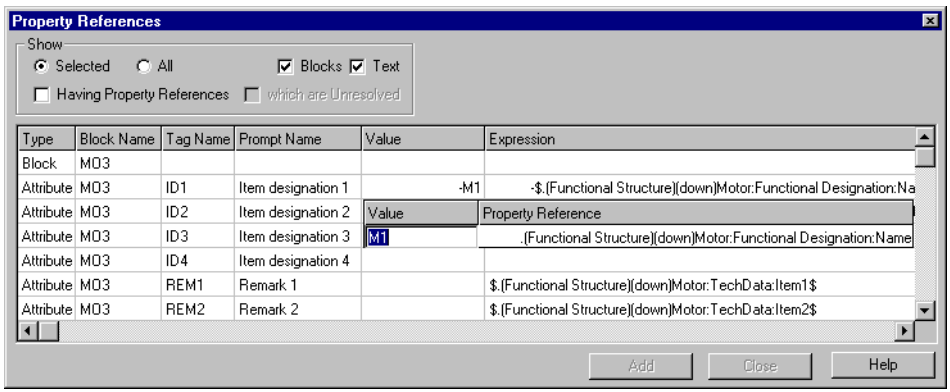


Figure 185. AutoCAD Integration - Update in mixed expression

The example in Figure 185 shows how to edit the value if the expressions is e.g.

-$.(Functional Structure)(down)PCS:Functional Designation:Name$

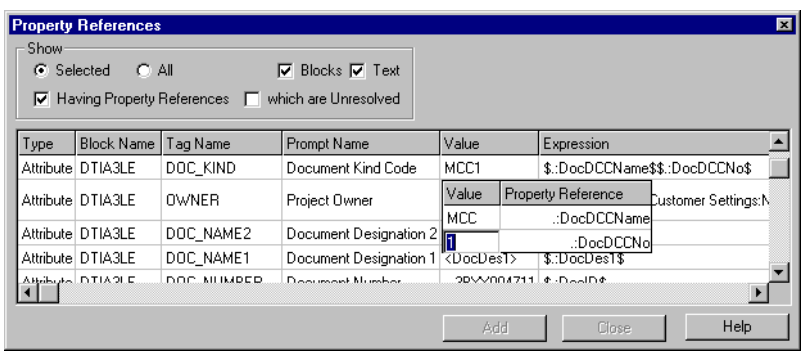


Figure 186. AutoCAD Integration - Update in multi-reference expression

The example in Figure 186 shows how to edit the value if the expressions is e.g.

$.:DocDCCName$$.:DocDCCNo$

thus contains multiple property references.

When leaving the value cell the value will be updated to the property that is referenced.

If the expression contains unresolved property references the value cannot be edited.

Aspect List Dialog

The **Aspect List** dialog will be started when selecting the menu item **Document Manager > Aspect List**.

The **Aspect List** dialog shows all aspects of one object. The objects that can be shown are those objects having an aspect that is referenced via a property reference within the current drawing. Property references will be selected by clicking on the drawing objects. This can be all drawing objects of type Block, Attribute, Text and Multiline Text.

The dialog is non-modal, so it is possible to keep the dialog open while changing the selection. When changed the selection the aspect list may or may not change dependent of if because of the change a different aspect of a different object was selected.

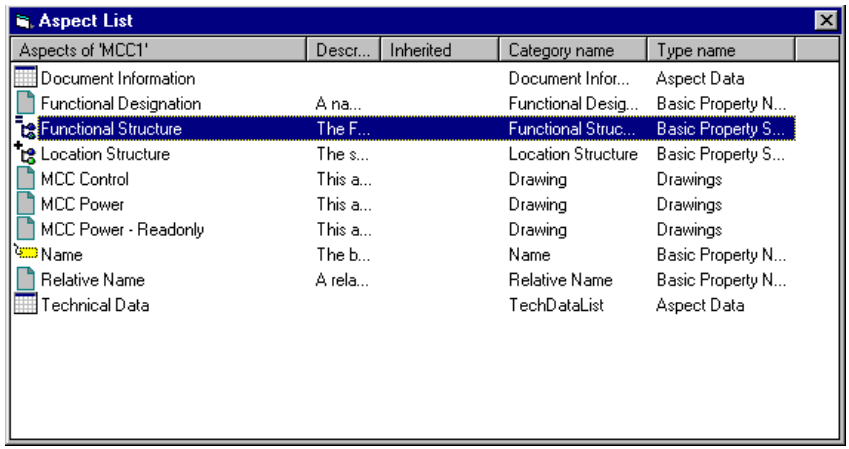


Figure 187. AutoCAD Integration - Aspect List Dialog

- **Starting the Aspect List Dialog**
  The dialog will be started by selecting the menu item **Document Manger > Aspect List**. Then a prompt on the command-line shows *Select an Entity* (An entity is a drawing object).
- **Selecting Drawing Objects**
  If the **Aspect List** dialog is already displayed each change of a selection may cause a change in the aspect list.

  If there is no property reference assigned to the selected drawing object the content of the dialog does not change.

  If there is a property reference assigned to the selected drawing object it might be possible that the selected aspect changes as the new property reference refers to a different aspect of the same object. However if the property reference refers to an aspect of an other object the aspect list changes.

Open Drawing Dialog

The **Open** dialog will be started when selecting the menu item **Document Manager > Open**.

With the **Open** dialog it is possible to open drawings by selecting them by navigating through the structures visible in the **Engineering Workplace**. The only aspects that can be seen are aspects of aspect type **Drawings**.

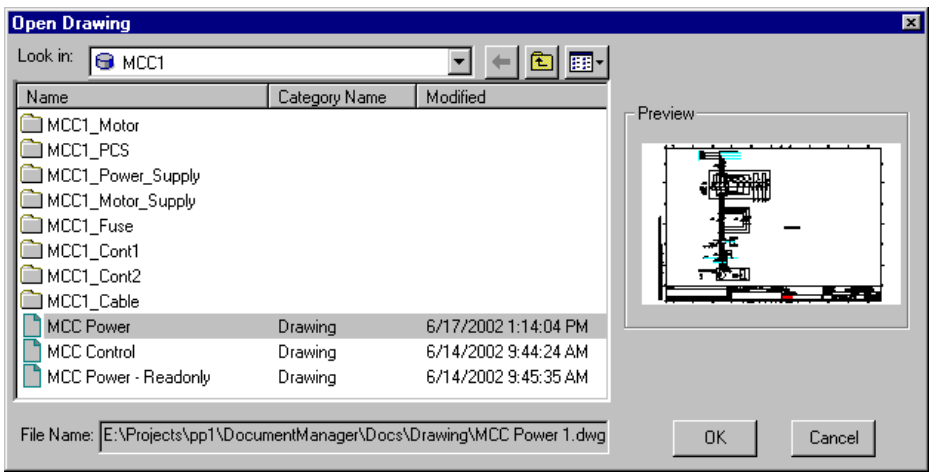


Figure 188. Document Manager - AutoCAD Integration - Open Dialog

- **Navigation**
  Navigation is similar to navigation in the standard open dialogs of Windows. The structures can be selected at the top of the dialog as selecting drives in the standard open dialog. Aspect Objects play the role of folders and aspects of aspect type Drawings play the role of files.
- **File Formats**
  Document Manager - AutoCAD Integration supports the file formats dwg, dxf, and dwf. Within the Document Managers **Open** dialog inside AutoCAD only drawings having the dwg format can be opened and previewed. In this case a thumbnail view shows the content of the drawing.
- **Selection**
  When working with AutoCAD's multi document interface (MDI) it is possible to select more then one file within one Aspect Object and open them simultaneously.

When working with AutoCAD's single document interface (SDI) only one drawing can be selected at a time.

The single document interface in AutoCAD is better known as **Single-drawing compatibility mode** and can be set within the **Options** dialog on tab **System**. The **Options** dialog can be opened by the context menu when selecting the **Options** menu item.

When selecting multiple aspects the thumbnail preview and the file path of the first aspect will be shown. If some of the selected aspects do not have a drawing in dwg file format but in dwf or dxf file format the first aspect with a dwg drawing file will be shown. Furthermore the aspects having drawing files in dwf and dxf file format will not be opened.

- **Read-only Files**
  In cases where the underlying file is read-only this will be shown underneath the thumbnail view. The drawing will then be opened in read-only mode.

How to Refresh Property References

Refreshing the values of the property references within a drawing can be done either on demand by selecting the menu item **Document Manager > Refresh** in AutoCAD or by specifying to refresh during open or during open and print in the options. How to change the options is described in Options Dialog.

Anyway, if the drawing is read-only refreshing has no effect to the drawing. The menu item **Document Manager > Refresh** is then disabled.

When starting subscription (described in How to Subscribe Property References) all property references will be initially refreshed as well.

Whenever an error occurs during refreshing property references they will be shown with in the error list dialog **Document Manager - AutoCAD Integration - Error List** which pops up in case of errors.

Dependent on the settings for the error behavior the old value or a replacement text (e.g. '##') will be shown if a property reference cannot be resolved. See Options Dialog on how to specify this settings.

How to Subscribe Property References

Subscribing values using property references has the effect that whenever a property value will be change by some application, the changed value will be immediately updated to the drawing if the drawing is currently open and active. If a drawing is currently open, subscription is enabled but the drawing is not the active drawing the changed values will be updated when the drawing becomes active. Even if all drawings will be closed without explicitly activating the drawings they will be refreshed before closing them.

Subscribing the values of the property references within a drawing can be done either on demand by selecting the menu item **Document Manager > Subscribe** in AutoCAD or by specifying to subscribe during open in the options. How to change the options is described in Options Dialog.

Anyway, if the drawing is read-only subscribing has no effect to the drawing. The menu item **Document Manager > Subscribe** is then disabled.

When starting subscription all property references will be initially refreshed as well.

Whenever an error occurs during initially refreshing or subscribing property references they will be shown with in the error list dialog **Document Manager - AutoCAD Integration - Error List** which pops up in case of errors.

Dependent on the settings for the error behavior the old value or a replacement text (e.g. '##') will be shown if a property reference cannot be resolved. See Options Dialog on how to specify this settings.

How to Open a Drawing

Drawings can be opened within

1. **Engineering Workplace**:
   See Open a Document Aspect on how to open an drawing aspect within the Engineering Workplace.
2. **AutoCAD**
   See Open Drawing Dialog on how to open an drawing aspect within AutoCAD.

When opening a drawing with the AutoCAD standard Open dialog using menu item **File > Open** a drawing file will be opened but without having a connection to an

aspect. In this case the menu **Document Manager** will not appear within the menu bar.

How to Print a Drawing

Drawings can be printed within

1. **Engineering Workplace**:
   See Print a Document Aspect on how to open a drawing aspect within the Engineering Workplace.
2. **AutoCAD**
   Use menu item **Document Manager > Print** to print the current drawing. On how to configure the print behavior in this case see Options Dialog.
   When printing with AutoCAD's plot function printer and plot layout configuration must be done manually.

When start printing from **Engineering Workplace** or from **Document Manger Application** the drawing file opened within AutoCAD will be closed after printing has finished if it was not already opened before.

How to Create or Change a Plot Layout

Open the plot layout drawing which is referenced on the **Plotting** tab of the **Options** dialog. Within the plot layout drawing a set of page setups are predefined, one for each format that can be seen in the **Options** dialog.

There is a default text within the plot layout drawing (See Figure 189) which should not be deleted because there is otherwise no way to specify the **Plot Area** to be **Extends** within this drawing.

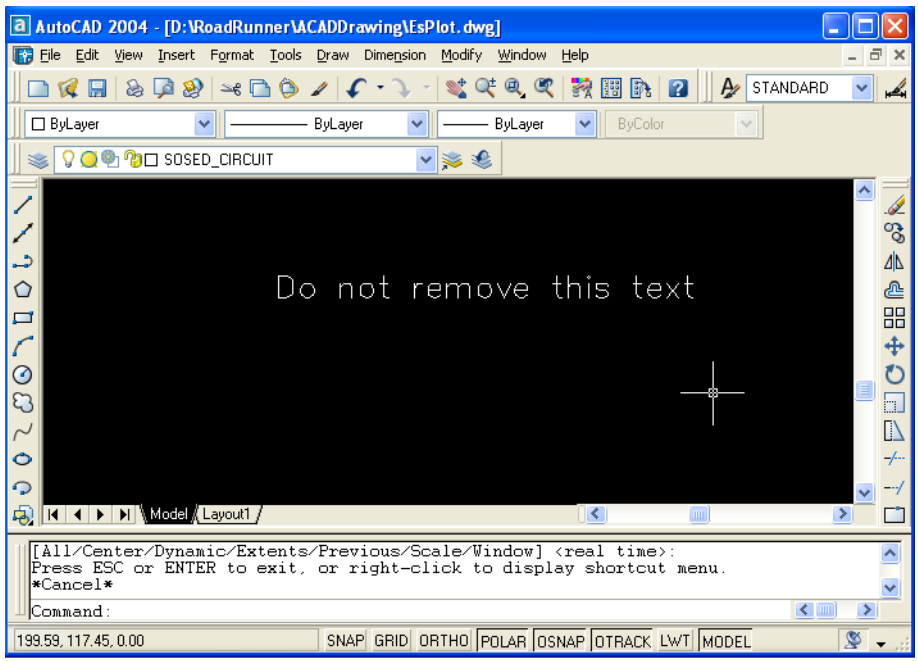


Figure 189. AutoCAD Integration - Plot Layout Drawing

When selecting menu item **File > Page Setup** the **Page Setup - Model** dialog appears.

At the **Plot Device** tab (Figure 190) no specific plotter should be specified. The plotter will be assigned within the Options dialog of the Document Manager. Specifying a plotter here may cause errors if the plot layout drawing will be used on machines where the printer is not installed and where a different one should be used. So, on this tab always **None** should be selected for the plotter.

Plot style tables can be used as desired.

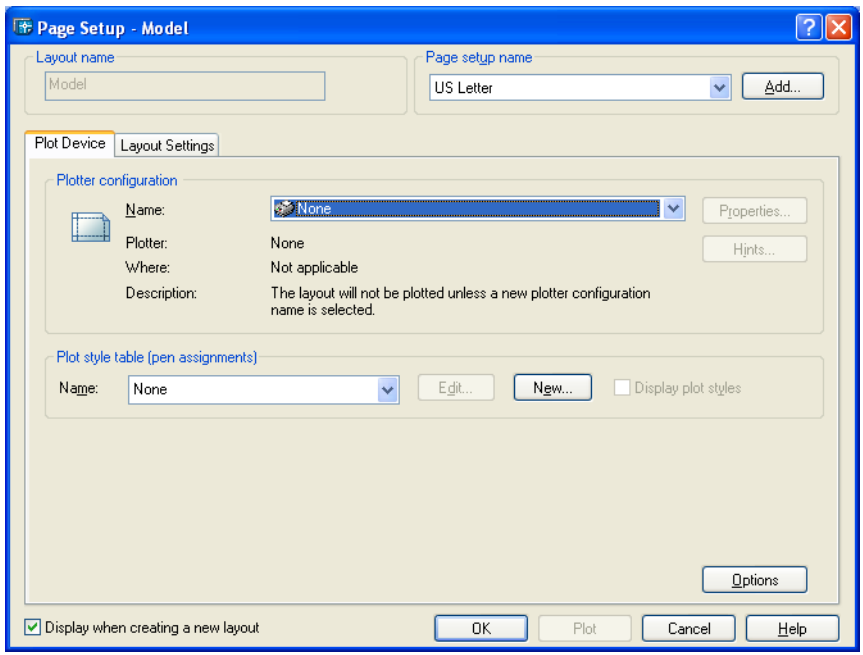


Figure 190. AutoCAD - Page Setup dialog - Plot Device

At the **Layout Settings** tab (see Figure 192) it is possible now to specify the paper size, plotting area, orientation, scale and offset. The settings can be adapted to the requirements of each format.

In both cases, when a new page setup will be created or an existing page setup should be changed the **Add** button must be pressed to proceed. Within the **User Defined Page Setups** (see Figure 191) dialog the new format name respectively the existing format name (in case of changes) must be typed in at the **New page setup name** edit box.

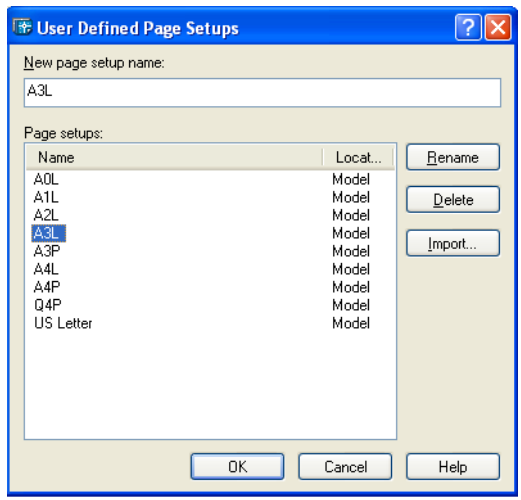


Figure 191. AutoCAD - User Defined Page Setup dialog

When pressing at the OK Button the new setup will be stored. In case of changes a message box comes up with a message like

Page setup “A3L“ already exists - Redefine it?

Accepting this message box with **OK** will make the changes persistent.

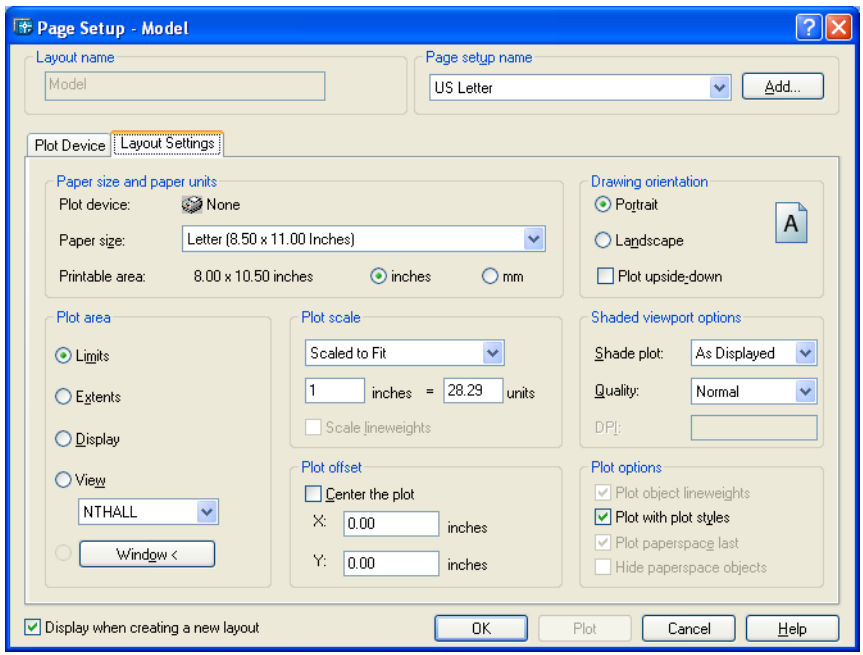


Figure 192. AutoCAD - Page Setup dialog - Layout Settings

How to Navigate

Using the **Aspect List** dialog it is possible to navigate to other Aspect Object starting from an open AutoCAD drawing and open any view or execute any action that is accessible from the aspect or object verb menu.

See Aspect List Dialog on how to start the **Aspect List** dialog and on how to select the Aspect Object shown within the **Aspect List** dialog.

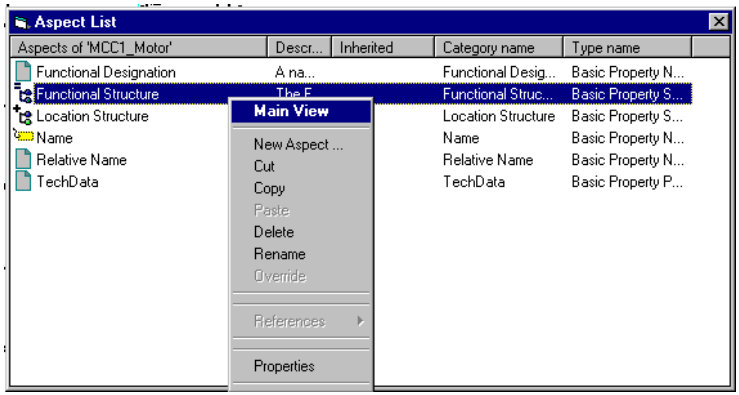


Figure 193. AutoCAD Integration - Aspect List with aspect verbs menu

Navigation is primarily done using the **Main** view of any Structure aspect (e.g. **Functional Structure** or **Location Structure**).

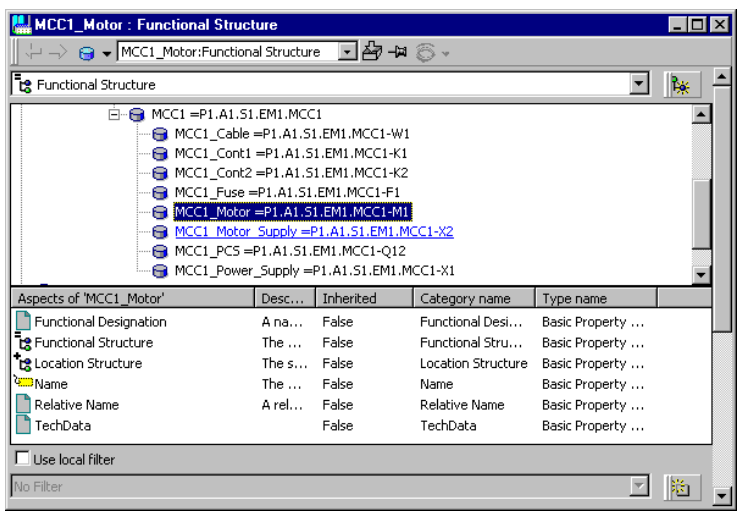


Figure 194. AutoCAD Integration - Structure view for navigation

Audit Trail Events

Document Manager creates Audit Trail Events for modification operations onto Document aspects if the Audit Trail feature is enabled within the current 800xA system:

- Modification of Document properties will be listed in Alarm & Event aspect, column “Message Description” or NEWVALUE. The modified property will be shown like:
  PropName: <property name>, OldValue= <..>, NewValue=<...>.

Authentication

System 800xA supports Re-Authentication or Double Authentication, which can be configured for Document Manager, Document and AutoCAD Drawings categories:

- At initial these functionality is not active. For example to activate Re-Authentication and Double Authentication for the Document category, enter the Aspect System Structure, “Document Manager, Aspect System”, “Document Manager Documents”, “Document” and check in aspect “Aspect Category Definition” the check boxes for Re-Authentication or Double Authentication.

E-Signature Properties

System 800xA supports E-Signature functions, First and Second Signature which can be activated for Document Manager categories.

- At initial these functionality is not active. For example to activate E-Signature for the Document category, enter the Aspect System Structure, “Document Manager, Aspect System”, “Document Manager Documents”, “Document” and check in aspect “Aspect Category Definition” the check boxes for First Signature or Second Signature.

Tutorial - Creation of Generic Dynamic Documents

The term *Generic Document* means a document which contains inserted references and is used as a template

- either in an object type definition or
- in a library system or
- as a template.

In each of these cases, the generic document is instantiated as a 'real' document, e.g. when an object of the type definition is instantiated or when a document from a library system is instantiated in a working system or when a real document instance is created from the template.

The problem now is to insert references into the generic document in such a way that they are resolved correctly in the instantiated document. These references can point to an attribute value of an aspect in another object or to an attribute value of the same aspect, i.e. to *this* document aspect.

Example 1

Consider the following scenario:

1. There is an object type definition in Plant Explorer's **Object Type Structure** where you build a complex generic structure of objects containing all sorts of aspects, including documents.
2. Objects in another structure, e.g. the **Functional Structure,** can be instantiated from this object type definition.
3. The documents of the document aspects in the object type definition are technical descriptions of a kind and contain references to show values of properties/attributes belonging to aspects in the object type definition.

Of course, you want the values of the references to be updated correctly not only in the documents of the object type definition but also in the documents of the instantiated object - what you want to create is a *generic dynamic document* or template.

- The *problem* is that instanced objects normally have a different name than the original objects in the type definition, so if there is a reference to, lets say object *X* and of this object the aspect *Y* and of this aspect the attribute *Z*, the reference cannot be resolved in the instantiated object as there the object is not named *X* but maybe *A*.

- The *solution* is to take advantage of the different name categories available in Plant Explorer. The objects in the library project should have *two* Basic Properties aspects:

  a. An aspect *Name*, (in our example this has the value *X*) which is always created by default when creating a new object and

  b. an additional aspect of category *Basic Property/Basic Property Name/ Relative* Name, whose *Name* attribute is set to maybe *xxx*.

In the document, we do *not* reference the object via the attribute *Name* of the *Name* aspect because the value of this attribute changes from *X* to *A*, but via the attribute *Name* of *Relative Name* - here the value of the *Name* attribute remains the same ('xxx') and so, the inserted reference in the copied document of the instantiated object can be resolved and updated with the actual value, though the objects visible in Plant Explorer have completely different names.

It works correctly only if the *Name* attributes of the *Relative Name*-aspects contain unique names, at least in regard to the substructure in which the aspects are located.

For generic dynamic documents, only the relative format of the reference string can be used, not the absolute format!

In addition,

- the *Object Identification* has to be set to '((Relative Name)) <Name>',
- *Insert Formula Only* has to be selected,
- reference type *Child* has to be selected and
- in *Reference String,* the structure name has to be modified to the structure in which the instantiated object is to reside later.

Refer to Figure 195 which shows an example of how to insert a reference in a document of the object type definition.

The aspect of type *SignalParameter* of which we want to reference the attribute *ACT* has the name *SignalType1* and the relative name *Sig1*. Note that *Object Identification* is set to '((Relative Name)) Sig1'.

In *Reference String,* the structure name is given as 'Functional Structure'.

Reference type *child* and Insertion mode *Insert Formula Only* are selected.

For a detailed explanation of the two formats please refer to Types of References.

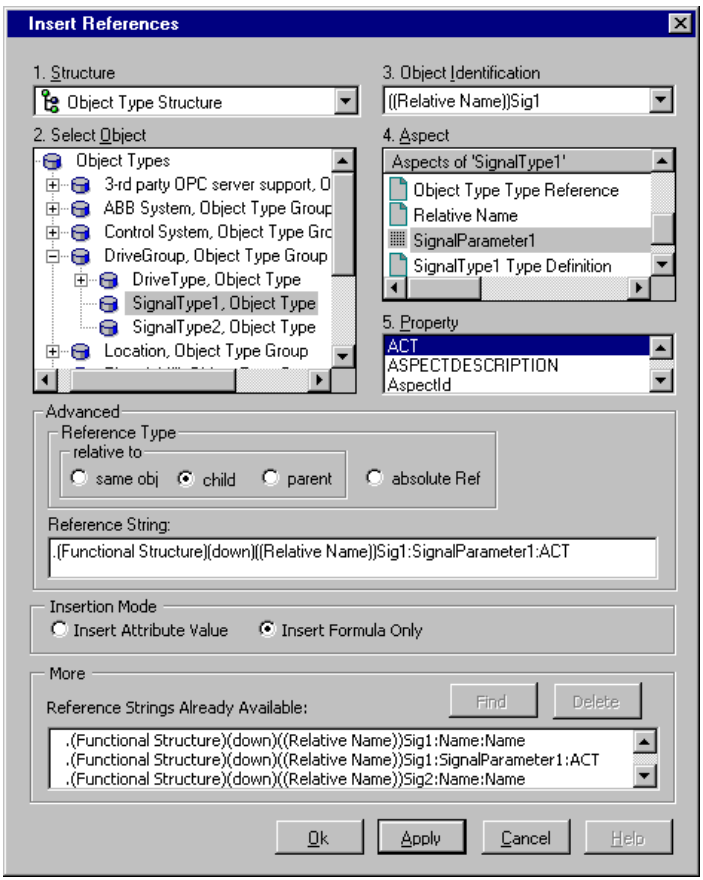


Figure 195. Insert Reference in an Object Type Definition Document

Example 2

Consider the following example:

1. You want to prepare your own Word Template in the Library Structure where Document Manager templates are located. This Word Template should present

in the footer the DocID property to show the “Document number”. The “DocId” is one of the document’s own properties.

To resolve correctly the inserted reference, the property reference string must not contain the aspect’s name but be aspect-relative.

To insert the reference into the footer of the “Documents_FD Template” document aspect, select that aspect and the required “DocId” property but then edit the resulting reference string:

It should ends up in the format

.:<property_name>, e.g.

.:DocId.

Refer to Figure 196.

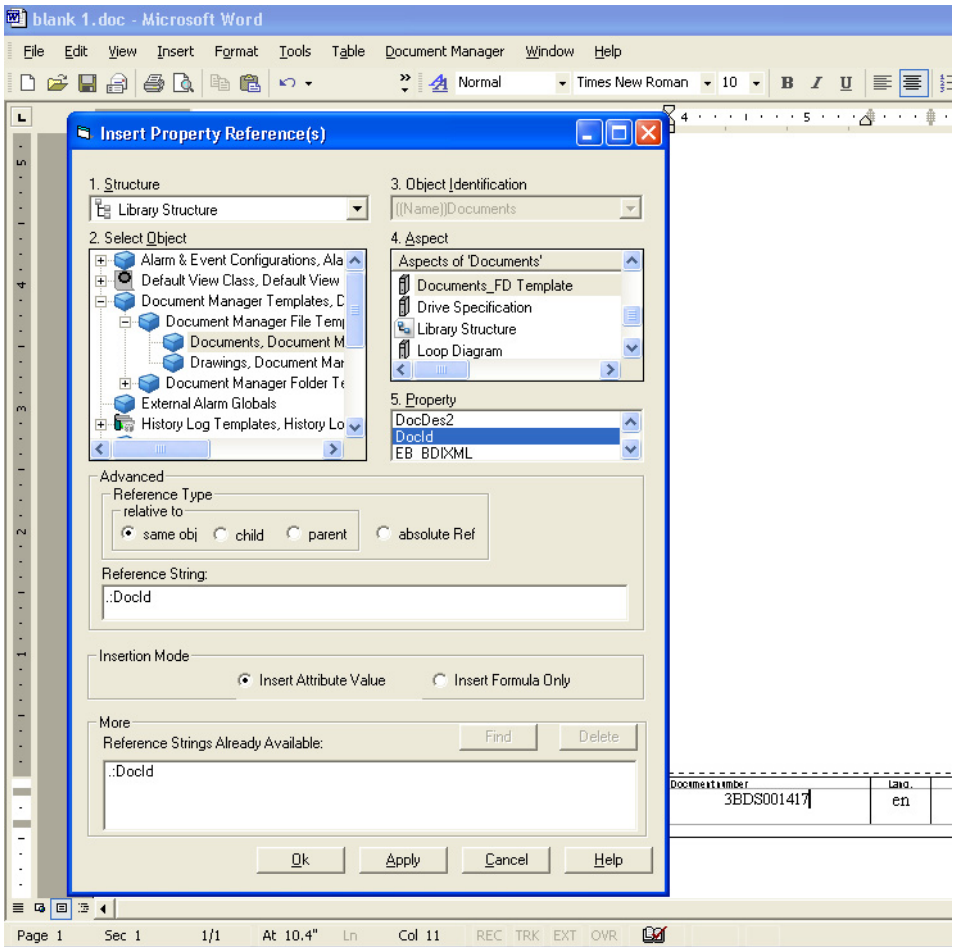


Figure 196. Insert an aspect-relative Reference into a Document Template

Error Messages

If any errors occur, error messages are displayed. Errors can either be displayed in the Plant Explorer's **Message** window or in a application's own error handling window.

Tracefile Writing

Many actions performed on documents are traced in files DmPm***.log** stored in the user-specific folder *MyDocuments\DocumentParameterManager\trace*. These tracefiles are not so much meant for the user but for analysis by support specialists after errors occurred (see also next section). The following logfiles are written:

- DmPmASO.log (for actions started from the Plant Explorer).
- DmPmApp.log (for functions executed in Word documents).
- DmPmExt.log (fro system creation/deletion and backup/restore).
- DmXLSRepGen.log (for Word Report Excel AddIn activities)
- DmRepGen.log (for Word Report data transfer functions)

Reporting Problems

Please report any problem by filling in a Product Maintenance Report (PMR). For additional information please provide a copy of the respective tracefile as this will greatly facilitate analysis.

What to do at Program Halts

See Reporting Problems.

Section 8 Reuse Assistant

Reuse is seen as the most relevant success factor during engineering. Instead of re-inventing the complete problem solution, pre-configured standard solutions are to be reused to keep the variety of *Solutions* down. This not only ensures faster and safer engineering, with less testing effort, but also helps in operation and maintenance due to a minimum of diagnosis effort in the case of an error.

Reuse of *Solutions*, e.g. a valve control, a pump, a reactor etc. can only be achieved, if the solutions are known. Above a certain basic level, more elaborate solutions are seen as combinations of the basic level solutions. The nature of the application domain makes this relationship arbitrarily complex and recursive.

Within the *System 800xA platform*, designed *Solutions* can be kept either in form of object types or simply by a set of *Aspect Objects*.

Due to the high number of possible combinations, not each combination of *Aspect Objects* of all *Aspect Object Types* can efficiently be presented in *System 800xA platform*. Even if that were the case, finding the matching *Solutions* would be a challenging task. To simplify the reuse the Reuse Assistant guides the engineer through the selection and applies a solution.

The Reuse Assistant is based on the *System 800xA platform* built by ABB. It is an option to the *Engineering Workplace* product.

It provides you with the Reuse Assistant Architect and the Reuse Assistant Builder and some small Reuse Assistant examples.

Reuse Assistant Architect

The **Reuse Assistant Architect** feature enables you to design a Reuse Instruction by defining substitution variables and a tree-like structure of questions with a set of possible answers and operation to be performed for each answer.

Reuse Assistant Builder

In a subsequent step you can use the feature **Reuse Assistant Builder** to apply the Reuse Instruction to instances. In a wizard-like view you will be asked to answer the questions and enter the values for the substitution variables. Once this is completed you can create the solution by executing the Reuse Instruction. Depending on the answers given and the operations defined, Aspect Objects get created or modified accordingly.

Reuse Assistant Example

The feature **Reuse Assistant Example** provides you with four small examples for designs of Reuse Instructions.

After having installed the examples using the Complete or the Custom option of Reuse Assistant Setup you can import corresponding .afw files, stored on <installdrive>:\Program Files\ABB Industrial IT\Engineer IT\Engineering Studio\Reuse Assistant\Import, manually with the Import / Export tool of the 800xA workplace.

Reuse Assistant Workflow

The classic way to ensure efficient engineering is to design reusable solutions as Object Types and reuse the Object Types and build instances. Without the Reuse Assistant for each variant of a solution an Object Type has to be created. Structuring the Object Types gets challenging and searching for a particular variant of a solution gets complicated. The Reuse Assistant is driving the reuse-concept further by drastically limiting the number of Object Types per solution and guiding the user through the selection by questions and answers.

You can use the *Reuse Assistant Architect Mode* to design reusable *Solutions* by defining a *Reuse Instruction*. Once designed you should document and test the *Solution* before you approve it for distribution. In a subsequent step applying a *Reuse Instruction* to a certain problem can then be done easily.

Architect - Getting Started

This subsection describes how to use Reuse Assistant in architect mode.

Reuse Assistant in architect mode is used to construct new, and edit existing Reuse Instructions. The result of a session is a new (or modified) aspect object type that includes the Reuse Instruction as one of its aspects. This aspect object type can be shipped to the engineer at site who wants to use the Reuse Instruction. Please see Architect - Shipping the Reuse Instruction for details.

When a Reuse Instruction gets executed (by Reuse Assistant in Build Mode), the operations that have been defined are carried out. An operation can, for example, create an aspect object. You can find a complete list of all available operations in the Appendix F, Reuse Assistant Architect - Reference.

In some cases, the available operation may not be sufficient to implement your Reuse Instruction. In this case, you can extend the function of the operations by adding own script code to the operation. Reuse Assistant supports the scripting language “VB Script”. This language itself is not described in this book, but you can view the Microsoft documentation on the WEB (www.microsoft.com/scripting). Reuse Assistant specific aspects of using “VB Script” are described in the Architect - Script References.

Analysis

Before you start your design work, you need to make an analysis of the problem you want to solve with the Reuse Instruction. The analysis can be split into the following steps:

1. Identify the different variants of the solution that your new Reuse Instruction should produce.
2. Identify the type of the aspect objects that you need to construct the solution.
3. Identify the decisions that the user of your instruction has to make in order to select the correct variant. This should lead you to a set of questions you want to ask the user and a set of possible answer your user can choose from.
4. Define what kind of actions (like creating a new aspect object) your Reuse Instruction should perform on the system of your user if she/he selected one of the different variants.

5. Put the “Questions” that you want to ask the user into an order that is logical your user.

Now you are well prepared to design the new Reuse Instruction. The Reuse Assistant in architect mode has a graphical user interface that makes is easy for you to implement your design.

User Interface

The Reuse Assistant in architect mode uses the Plant Explorer of the Engineering Workplace as its user interface. The objects that you are dealing with, like "Questions" or "Answers", are modelled as Aspect Objects.

Working Environment

The Engineering Workplace is suitable to design new Reuse Instruction.

In the Plant Explorer select the Reuse Design Structure. It is recommended to select the aspect filter **Reuse Instruction Architect**. This will filter away all aspects that are not relevant when you are working with Reuse Instructions (see Figure 197). Now the Plant Explorer is ready to be used to design Reuse Instructions.

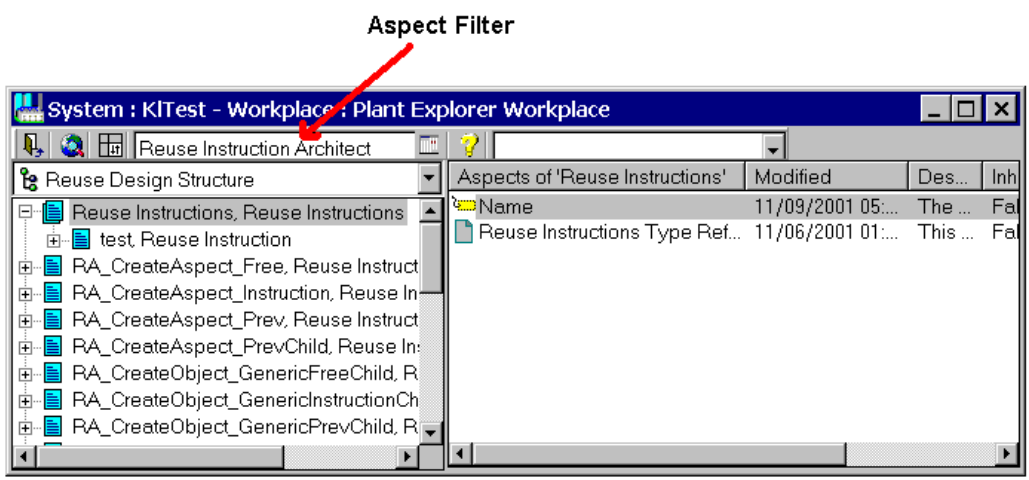


Figure 197. Aspect Filter

Architect Mode

You can use the Reuse Assistant in architect mode to design new and to edit existing Reuse Instructions. To create a new Reuse Instruction, select the root object in the Reuse Design Structure, open the New Object dialog and create a new aspect object of the object type Reuse Instruction.

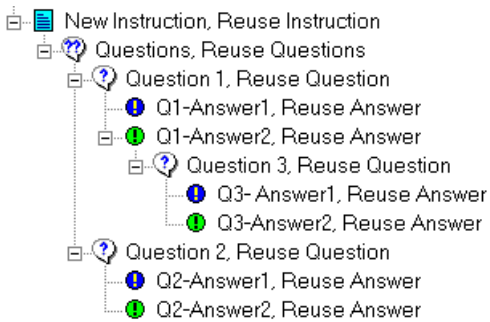


Figure 198. Simple Instruction

Once you have created a new Reuse Instruction object you can define the sequence of questions and answers (result of your analysis) using aspect objects.

Working with Questions and Answers

The Reuse Assistant in Build Mode (just called “Builder” for the rest of the document) presents the user a sequence of questions and answers. You define this sequence by creating a hierarchy of aspect objects. Place this hierarchy under the Questions aspect object of your new Reuse Instruction.

So, when the user executes Reuse Instruction as shown in Figure 198, she/he will see the following picture:

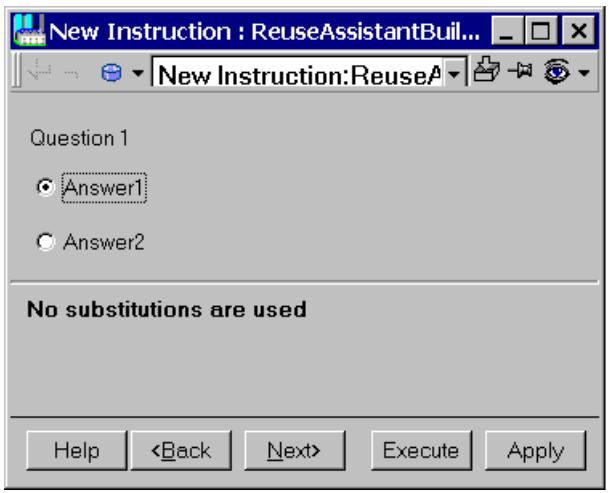


Figure 199. Simple Question

So, what can we learn from this two pictures?

- The text of the question that the Builder presents to the user is the name of the Reuse Question aspect object
- The possible answers that the users can choose from are the Reuse Answer aspect object that children of the Reuse Question aspect object.
- The text of the answers is the name of the Reuse Answer aspect object.
- The order of the answers is the order of the Reuse Answer aspect object underneath the Reuse Question aspect object.

The following figure summarizes this facts.

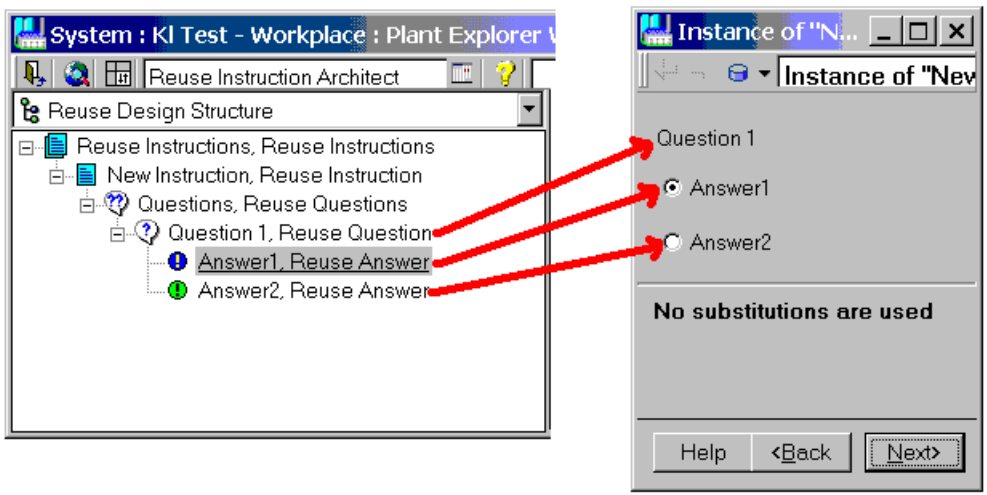


Figure 200. Relation Between Architect and Builder Objects

In a real instruction, you may want to ask the user more than just one question. You can insert new questions:

- Underneath the Reuse Questions object.
- Underneath an Reuse Answer object.

A question that is located under the Reuse Questions object is always presented to the user. A question that is located under an Reuse Answer object is only presented to the user if the answer that is the parent to the question is selected. Let's look at an example.

Assume the following object tree:

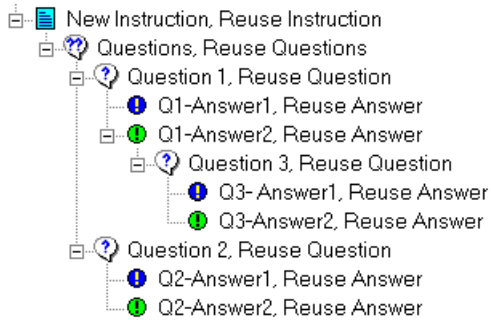


Figure 201. Nested Questions

When your user runs through this Reuse Instruction in the Builder, she/he gets the following sequence of screens:

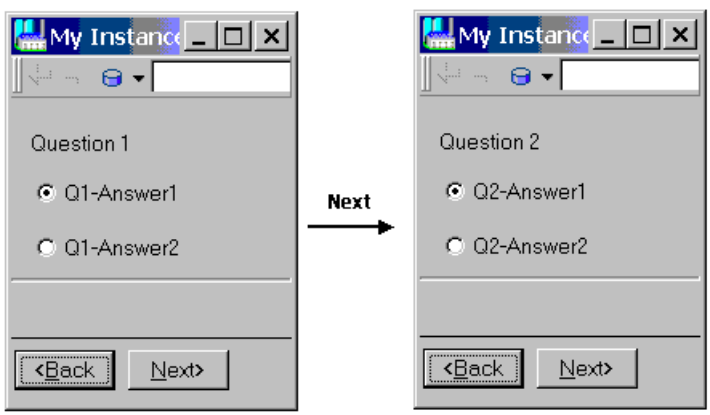


Figure 202. Nested Questions in Builder

If the answer **Q1-Answer2** instead of **Q1-Answer1** is selected, this results in getting the following sequence of screens:

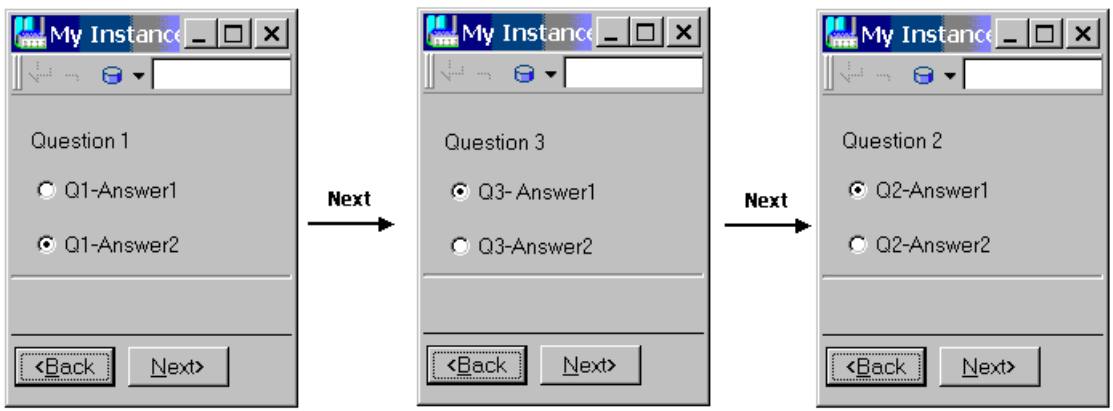


Figure 203. Nested Questions in Builder - 2nd Version

Here comes a short summary of the rules regarding Reuse Questions and Reuse Answers.

- A Reuse Instruction object has one child object of the type Reuse Questions.
- The Reuse Questions object can contain any number object objects of the type Reuse Question.
- A Reuse Question object can contain any number of Reuse Answer objects.
- A Reuse Answer object can contain any number of Reuse Question objects.

The order of the Reuse Question objects in the Reuse Design Structure defines the order in which the questions are presented to your user when she/he executes the instruction in the Builder. The order of the Reuse Answer objects in the Reuse Design Structure defines the order in which they are displayed by the Builder. You can change the order of the Reuse Question objects and the Reuse Answer object via the context menu of this objects.

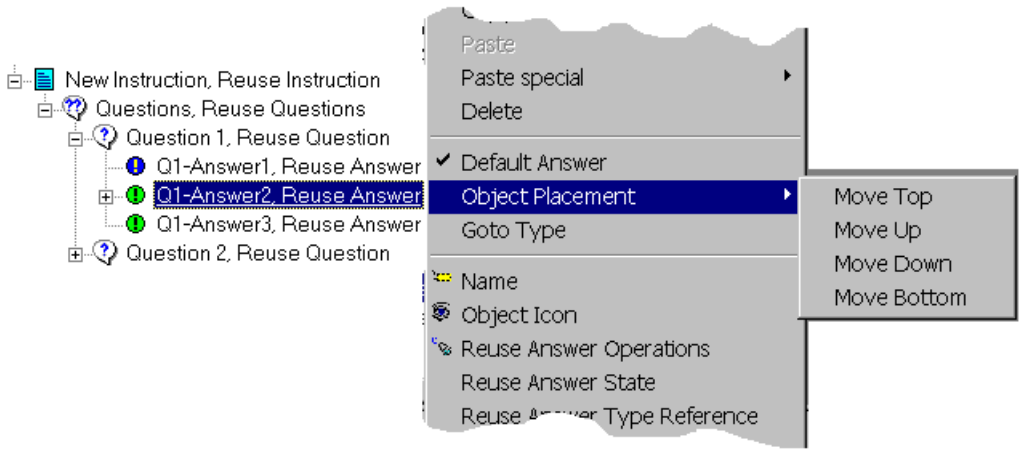


Figure 204. Object Placement

One of the possible answers of a Reuse Question should be defined as the default answer. The default answer should represent the “most likely” selection. You can define a Reuse Answer object as being the default answer by setting the setting the check mark on the **Default Answer** menu entry in the context menu of the Reuse Answer. The Default Answer is indicated by a blue icon in the Plant Explorer.

Working with Reuse Instruction Generator

Once you have defined the sequence of questions and answer, you can compile and test this sequence. Your Reuse Instruction will not yet do anything (the operations are still missing).

The Reuse Instruction Generator is an aspect of the Reuse Instruction object. Its task is to:

- check, if the Reuse Instruction is consistent.
- scan the complete sub-structure of the Reuse Instruction in the Reuse Design Structure and store the result including the operations (see Working with

Operations) and the help texts (see Architect - Working with Help Texts) in a single aspect that can be attached to an object or an object type.

The Reuse Instruction Generator shows the following user interface:

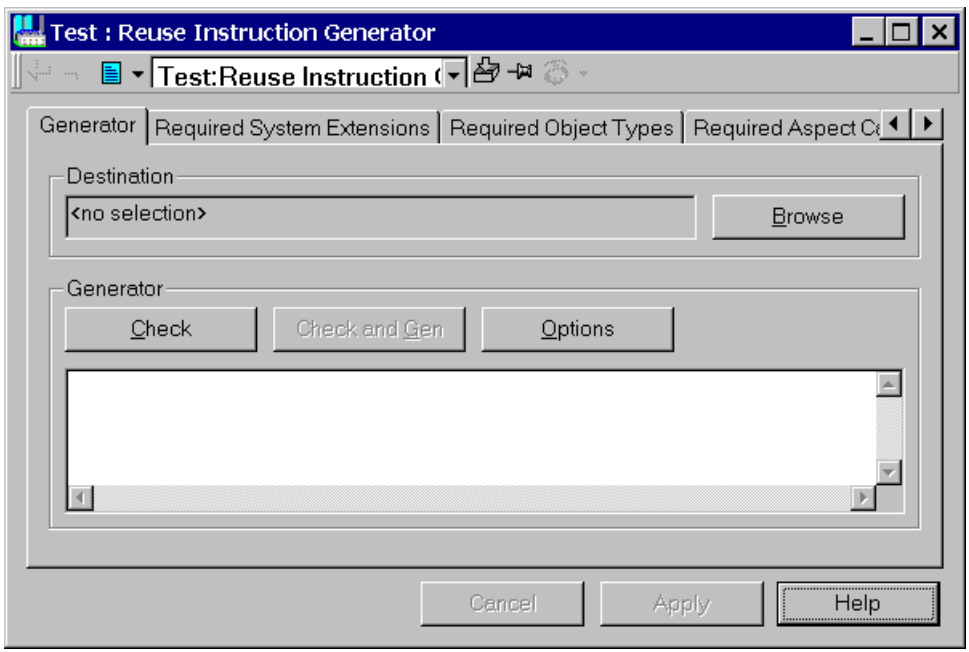


Figure 205. Reuse Instruction Generator

The most important button in this dialog is the **Check and Gen**. If you press this button, the Instruction Generator will start to scan the Reuse Design Structure and to product the output data as described in the beginning of this subsection.

You need to define the destination where the Instruction Generator should stored the result. The destination must be an aspect of the category Reuse Instruction attached to an object type. Use the **Browse** button to select an aspect. This will display the Browse Aspect dialog.

Browse Aspect Dialog

In the Browse Aspect dialog you specify the aspect where the Instruction Generator stores the compiled version of the Reuse Instruction.

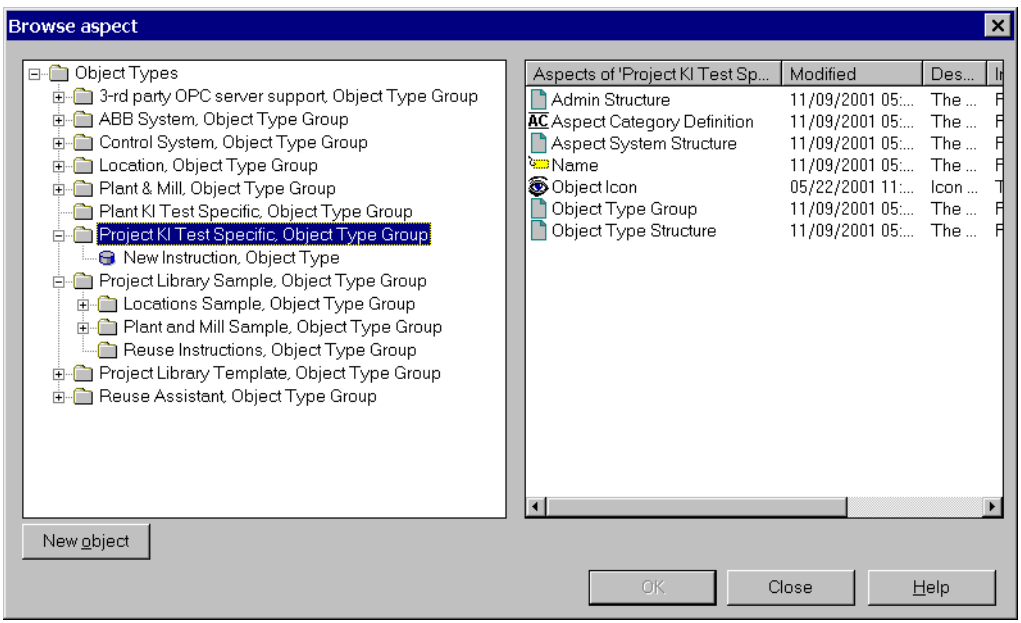


Figure 206. Browse Aspect

The Browse Aspect dialog shows the Object Type Structure on the left side of the dialog and the aspects of the selected object type on the right side. You can either select an aspect of the category Reuse Instruction from an existing object type, or create a new aspect object type with the **New Object** button. If you create a new object type, the system will automatically create an aspect of the category Reuse Instruction with it.

The Instruction Generator stores the path to the selected aspect.

Checks

The Instruction Generator performs a number of checks on the Reuse Instruction. If a check fails, an error message is written into the output window of the Instruction Generator.

Table 22. Instruction Generator Checks

Message	Type	Rule
More than one answer of this question are declared as 'Default Answer'.	Error	Only one of the Reuse Answer object belonging to a question can be the default answer.
A question without an answer is not allowed!	Error	A Reuse Questions must be followed by at least two Reuse Answers.
A question has to be followed by at least two answers!	Error	A Reuse Questions must be followed by at least two Reuse Answers.
This question has no default answer.	Warning	One of the Reuse Answer object of a Reuse Question should be marked as the default answer.
At this node exist at least two answers with name “Name Answer”.	Error	It is not allowed to have two Reuse Answers with the same name at the same Reuse Question.
The instruction contains no question objects.	Error	A Reuse Instruction must at least contain one Reuse Question.
At this node exist at least two questions with name “Name of Question”.	Error	It is not allowed to use two Reuse Question with an identical name at the same level.
At least one used object has not been created before!	Error	An operation can only use aspect object that are created by previous operation.

Table 22. Instruction Generator Checks

Message	Type	Rule
At least one used aspect has not been created before!	Error	An operation can only use aspects that are created by previous operations.
Error: Substitution “Name of Substitution“ is used but not defined! (in object: “Name of Answer Object”)	Error	All substitutions that are used by the operations must be defined in the Reuse Substitution aspect of the Reuse Instruction.

Generation of the Result

The Instruction Generator stores the result of Check & Generate in the Reuse Instruction aspect of the object type that is defined as Destination.

You can view this result using the Plant Explorer.

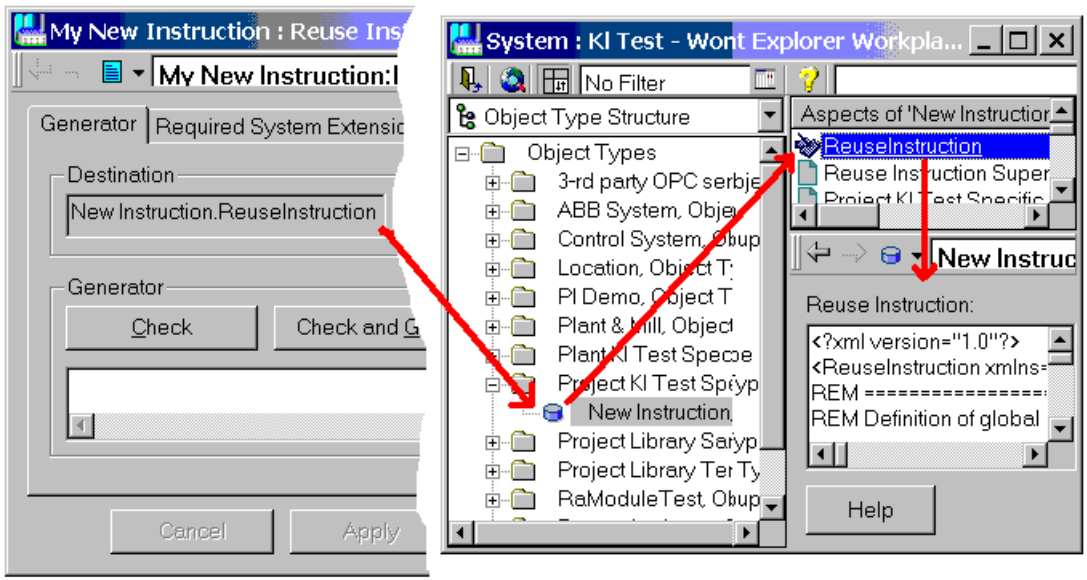


Figure 207. Object Type with Reuse Instruction

The content of the Reuse Instruction aspect is generated by the Reuse Instruction Generator. You can not alter this content.

The object type that is created by the Reuse Instruction generator is configured so that an instance of this type will:

- Inherit the Reuse Instruction aspect.
- Create one aspect of the category Reuse Assistant Builder.
- Inherit an aspect of the category Alarm and Event List.

You can create an instance of the object type to test your Reuse Instruction.

Testing Reuse Instruction

To test the Reuse Instruction, you need to create an instance of the object type that holds the Reuse Instruction aspect. Use the New Object dialog from the Plant Explorer to create a new instance. The following example shows the newly created object in the Functional Structure.

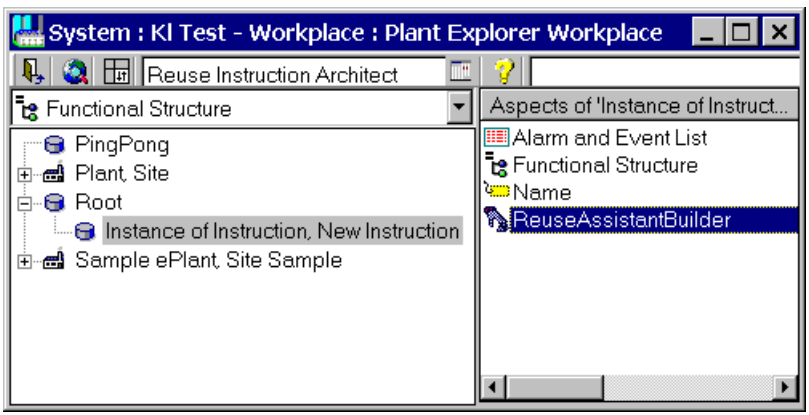


Figure 208. Testing a Reuse Instruction

The aspect ReuseAssistantBuilder can execute your instruction. For details, please look into the Architect Build Mode.

Working with Operations

A Reuse Instruction that consists of just questions and answer is not very useful, because it will not perform any action. So, to get a useful instruction, you need operations. The Reuse Assistant in architect mode supports different types of operations. They are all implemented as different aspect categories.

Reuse Answer Operations

An aspect of the category Reuse Answer Operations is automatically created with each object of the type Reuse Answer Operation. The operations that are defined in

the Reuse Answer Operations are carried out (by the Builder) if this answer is selected by the user.

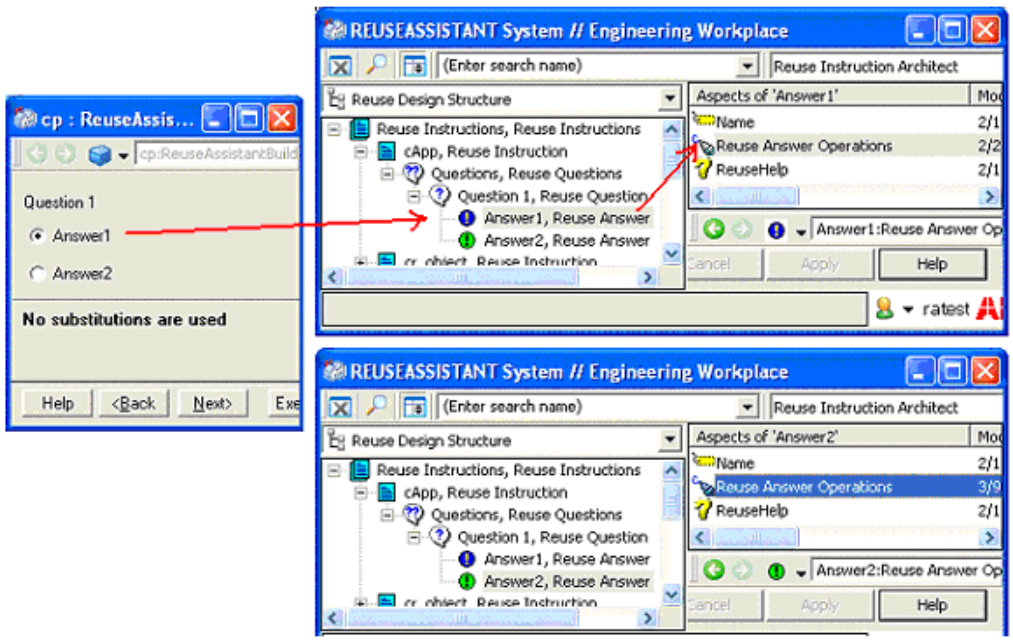


Figure 209. Answer Operations

In the example shown in Figure 209, the operations that are defined in the Reuse Answer Operations aspect of the object Reuse Answer 1 are executed (by the Builder). The operations that are defined in the Reuse Answer Operations aspect of the object Reuse Answer 2 are not executed.

Reuse Pre Operations and Reuse Post Operations

An aspect of the category Reuse Pre Operations and Reuse Post Operations can be attached to the Reuse Instruction object itself.

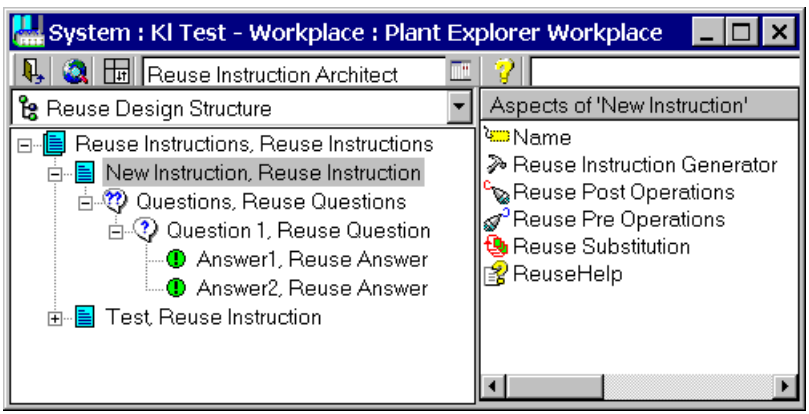


Figure 210. Pre and Post Operations

The operations that are defined by the aspect Reuse Pre Operations and Reuse Post Operations are always executed (by the Builder) independent of the selected answers.

The Reuse Pre Operations are executed before all operations that depend on an answer get executed.

The Reuse Post Operations are executed after all operations that depend on an answer have been executed.

The following figure illustrates this.

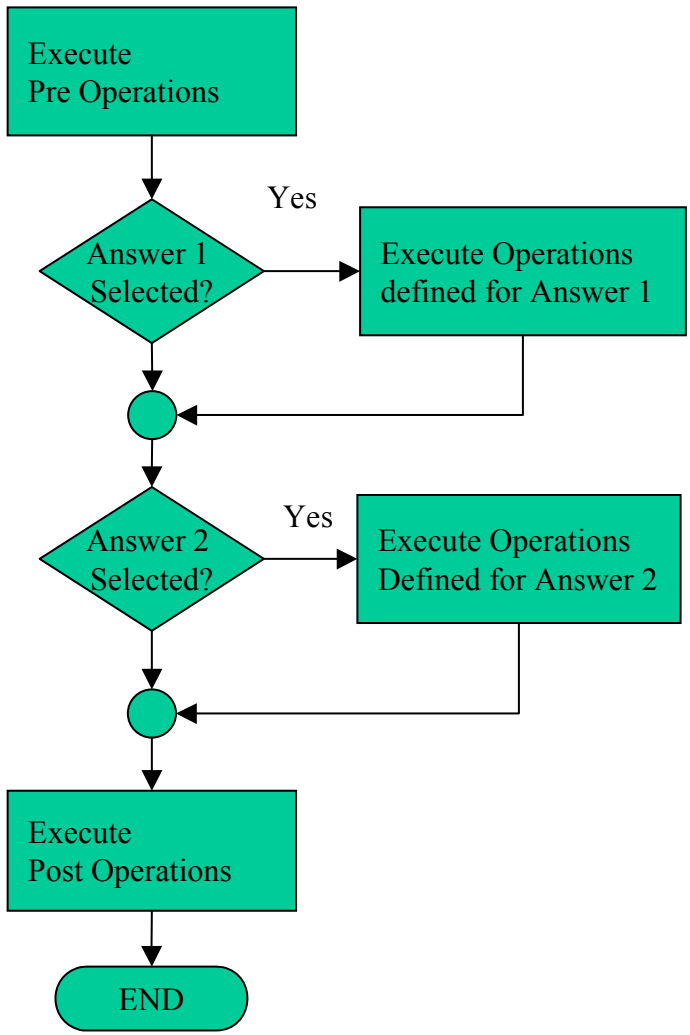


Figure 211. Processing Operations

Reuse Global Operations

Reuse Global Operations are implemented by one or more aspects of the category Reuse Global Operations. This aspect can be added to the Reuse Operation object. You can think of this aspect as of a library. Each of this aspects can contain one or more functions or sub routines. You can write them using the VBScript language. The functions that are defined in the Reuse Global Operations aspect(s) can be called from any other operation that is part of the same Reuse Instruction.

User Interface to Define Operations

You can define the content of the operations (Answer Operations, Pre Operations, Post Operation) using a graphical user interface.Reuse Global Operations are defined using a textual user interface.

The following figure shows the aspect page of a Reuse Answer Operation. It looks the same for the Reuse Pre Operations and the Reuse Post Operations.

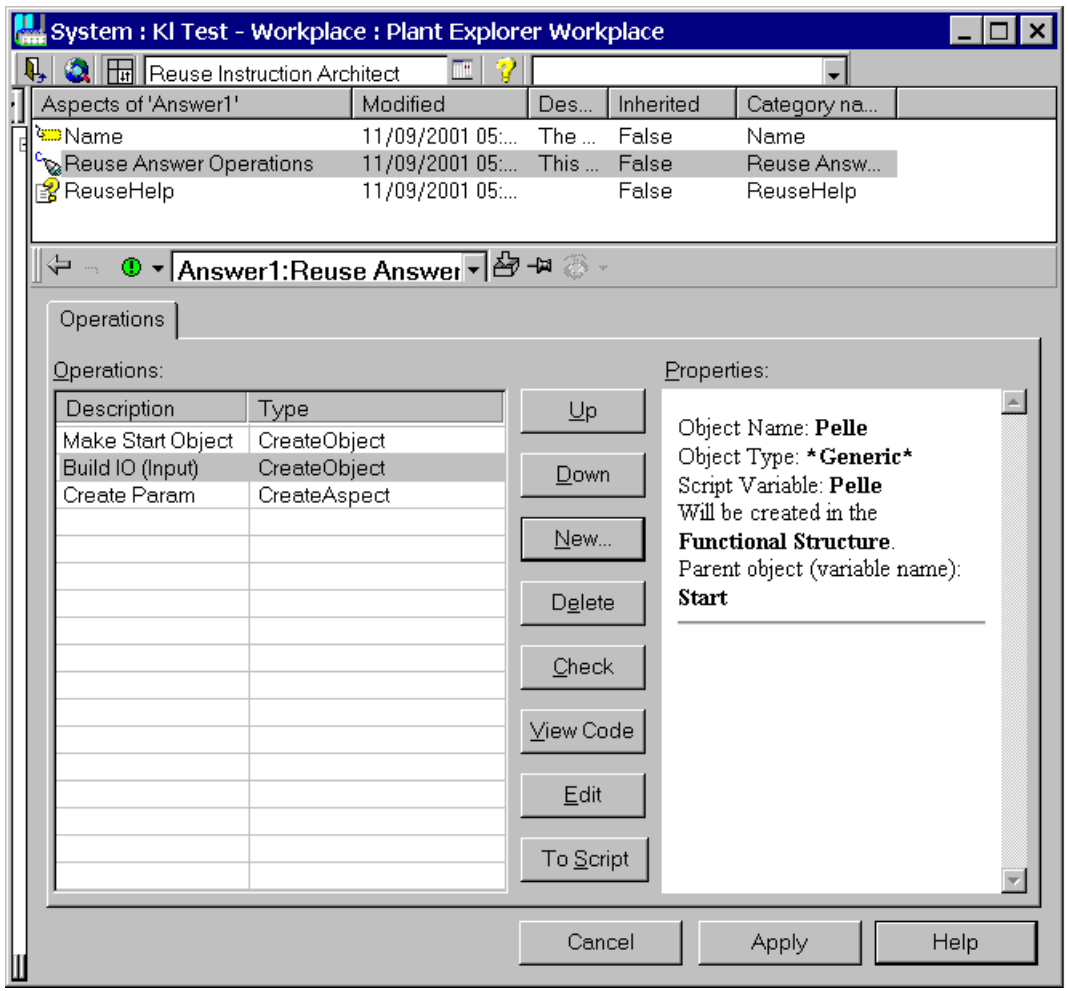


Figure 212. Operations Aspect Page

The left panel of the page shows the list of operations. The right panel displays information about the selected operation.

The following table describes the function of the dialog buttons.

Table 23. Buttons on the Reuse Operations Page

Button	Function
Up	Move the selected operation one position upwards in the list. The order in the list on the left hand side of the dialog defines the order in which the operations are executed.
Down	Move the selected operation one position down in the list.
New	Add a new operation. See Appendix F, Reuse Assistant Architect - Reference for a description of available operations.
Delete	Delete the selected operation.
Check	Check, if the current state is consistent. The following checks are carried out: All Substitutions that you are using in the operation must be defined (see Architect - Working with Substitution Variables). There must not be any unresolved references between the individual operations (see Green, Yellow and Red References for details). Each time you press the Apply button, the check is performed.
View Code	All the operations that you define are compiled into script code in the end. This button displays the script code that will be generated. It is useful to: Get a better understanding of how the Reuse Assistant works. Understand the reason for errors when the script is executed by the Builder. See how your own script code (see Architect - Script References) fits into the generated code.

Table 23. Buttons on the Reuse Operations Page (Continued)

Button	Function
Edit	This button lets you change an existing operation. You can also double click the operation in the left panel of the dialog.
To Script	The button lets you convert the selected operations into a single operation of the type Script Block. This allows you the change the code that the Reuse Assistant has generated.

User Interface to Define Global Operations

Global operations are defined in script code (VBScript). The aspect provides the following user interface to edit the script code.

Figure 213. User Interface to define Global Operations

Working with References

In many cases, you will write Reuse Instructions that are creating several objects and aspects. An operation could, for example, first create two objects and then create an additional aspect to one of these two objects.

So, within the Create Aspect operation, you will need to point to the object for which the new aspect should be created. This is called a Reference.

The Reuse Assistant is using strings to build and maintain references. This string is called a variable. The Reuse Assistant automatically assigns a variable to each object and each aspect that is created using one of the predefined operations. You can view and change this variable name when:

- Creating the operation
- Editing the operation.

The following dialog shows an example.

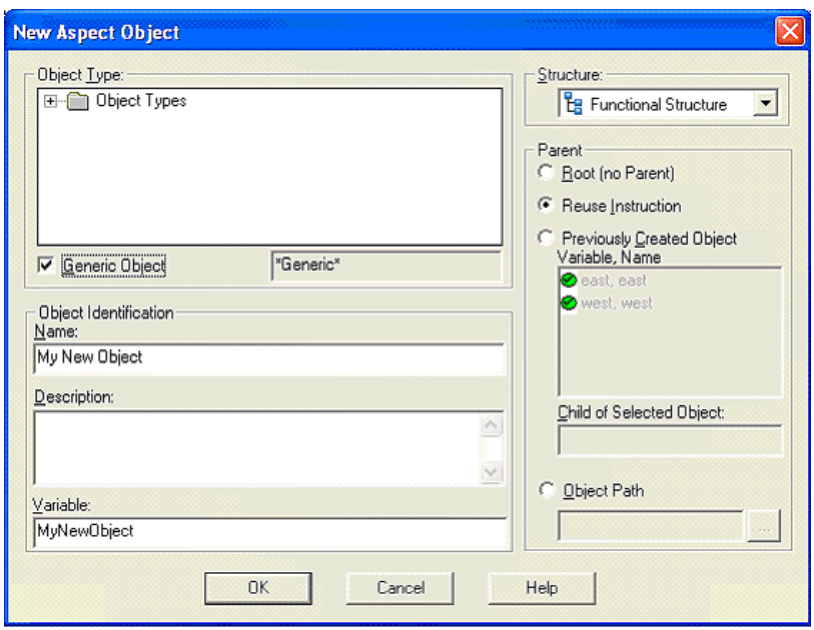


Figure 214. New Object Operation

This operation (the New Object Operation) will create a new aspect object. The name of the new object will be “My New Object”. The name of the variable that is assigned to this object is “MyNewObject”.

Assume now, you would like to add a new aspect to your object “My New Object”. The operation to do this is the New Aspect operation. This will display the following dialog:

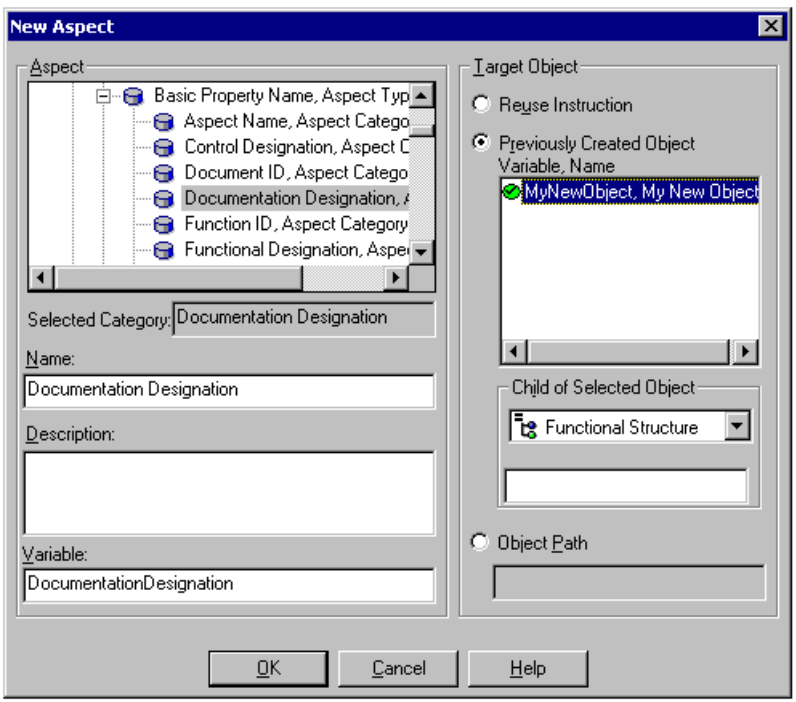


Figure 215. New Aspect Operation

As you can see in Figure 215, the object with the variable name “MyNewObject” is listed at the right part of the dialog as one of the possible targets for the new aspect.

You may also notice the green flag at this aspect. The meaning of this flag is described in the following subsection.

Green, Yellow and Red References

This subsection is important to be understood in order to design advanced Reuse Instructions.

Assume the following Reuse Instruction that is designed to create some objects in the functional structure.

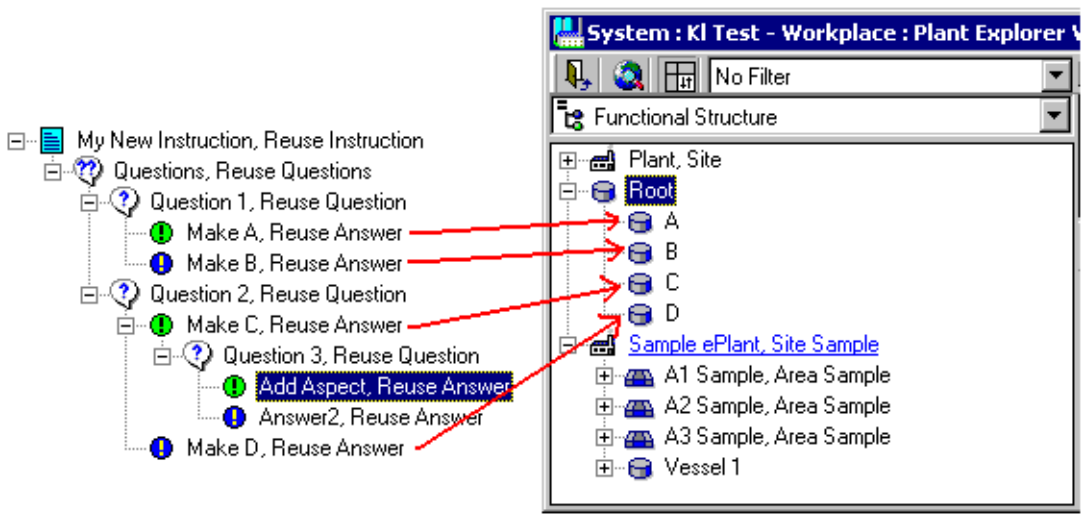


Figure 216. Using Mixed References

Now assume, we would like to add a new aspect to one of the objects created by the operation if the answer “Add Aspect” of the question “Question 3” is selected. Let’s analyze which objects exist when the operation of the answer “Add Aspect” gets executed:

- If the user has selected the answer “Make A”, then the object A exists.
- If the user has selected the answer “Make B”, then the object B exists.
- The object C always exists if the operation of the answer “Add Aspect” gets executed.

- The object D never exists, if the operation of the answer “Add Aspect” gets executed.

The dialog of the New Aspect operation reflects these using different colors for the references.

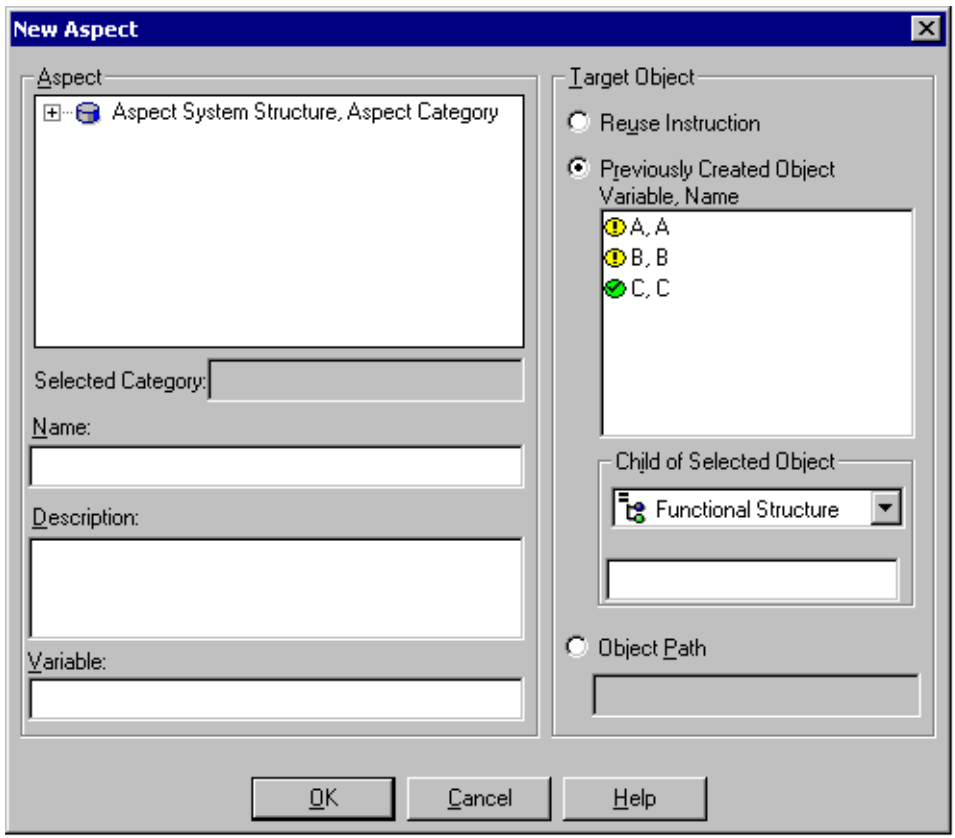


Figure 217. Color of References

The Target Object list contains the objects A, B and C. The object D is not listed, because we know that it will not exists when this operation gets executed.

The objects A and B are flagged yellow, telling you “be careful, this object may not exist when the operation is executed”. The object C has a green flag to indicate that it is safe to use this object as target of the operation.

However, in some cases it is useful to use yellow references.

Assume the following Reuse Instruction:

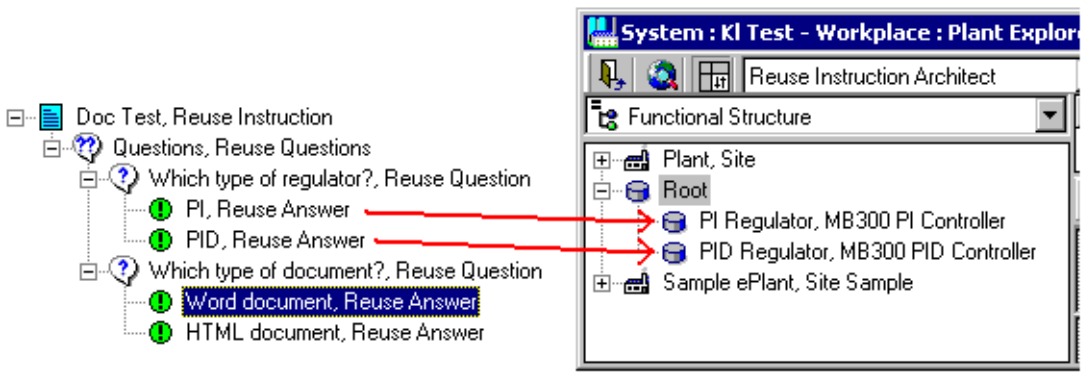


Figure 218. Using Yellow References

Depending on the user selection, this instruction will either create a PI or a PID regulator in the functional structure. Depending on the answer to the 2nd question, you want to add a Word document aspect or a HTML document aspect to the regulator. You want to do this independent of type of regulator that has been created.

To be able to do this, you need to assign the same variable to both regulators, the PI and the PID as shown in the following figure.

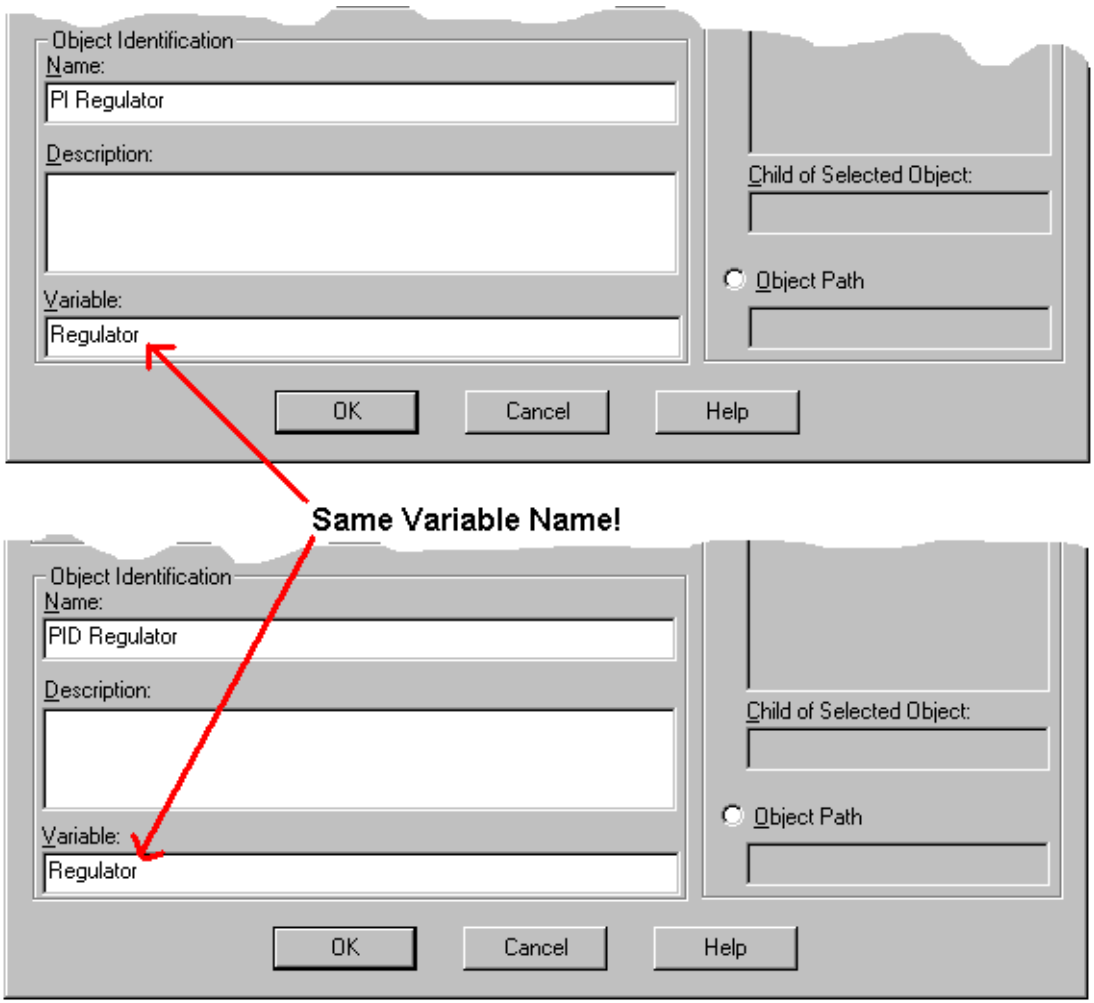


Figure 219. Variable Names

Please note that it's completely legal to use the same variable name many times! If you now select the Reuse Answer Operation of the answer "Word Document" and

define a new operation to create an aspect, the dialog will show the following references:

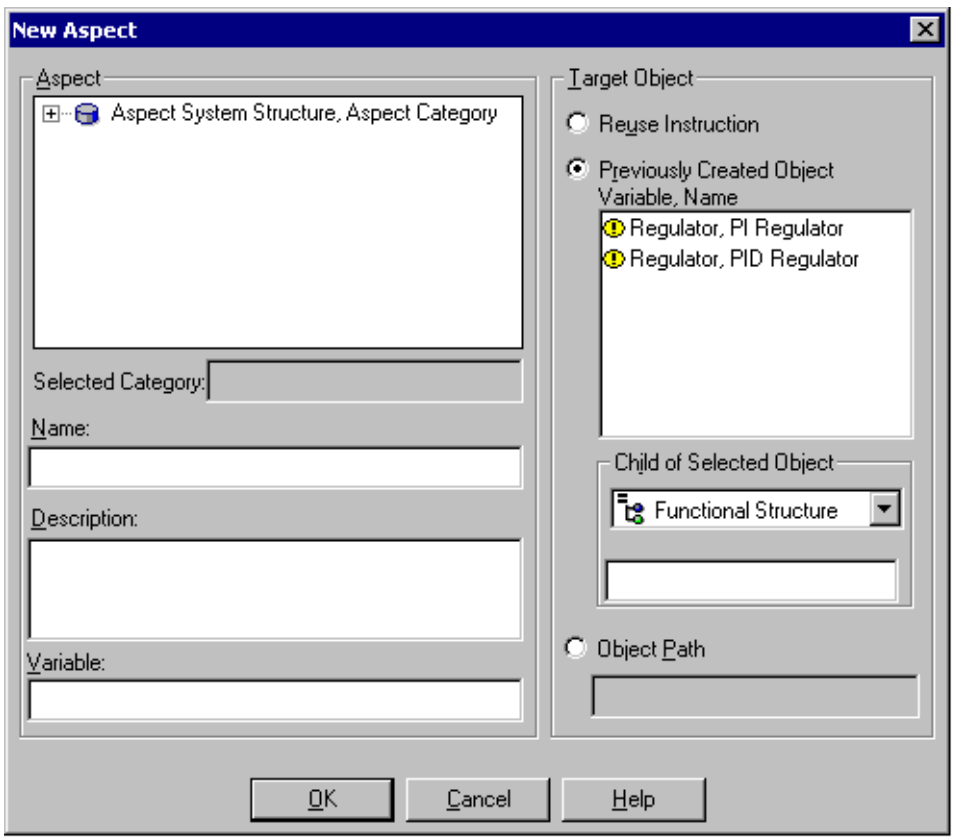


Figure 220. Create Aspect Operation

The dialog offers you two yellow references, both with the same variable name “Regulator”. Within the Reuse Instruction references are always build via variable names. So independent of which one of the two objects you select from the list, you are just pointing to an object that is identified by the variable with the name “Regulator”.

So, what will be the result in our case (see Figure 218)?

If the user selects the “PI” regulator, an aspect object of the type PI is created, and the variable Regulator is pointing to this aspect object. If the user now selects “Word Document” as document type, the Reuse Instruction will create a new document aspect for the aspect object pointed out by the variable “Regulator” which is the PI aspect object in this case.

If the user selects the “PID” regulator, an aspect object of the type PID is created, and the variable Regulator is pointing to this aspect object. If the user now selects “Word Document” as document type, the Reuse Instruction will create a new document aspect of the aspect object pointer out by the variable “Regulator” which is the PID aspect object in this case.

So, we have seen “Green” and “Yellow” references. What’s about the “Red” ones now?

A “Red” reference is a reference to an object (or an aspect) which is:

- created after the operation which wants to use the reference or
- created in a path that is never executed if the operation that wants to use the reference is selected.

So, “Red” reference will always lead to runtime errors when the Reuse Instruction is executed. As a consequence, none of the dialogs will ever offer you the possibility to create a “Red” reference. However, it may happen that you by accident create a “Red” reference. Looking at the example Reuse Instruction in Figure 218, you could end up having a “Red” reference is you move the Reuse Question object “Which Type of Document?” in front of the Reuse Question object “Which type of regulator?”.

In this case, the Reuse Instruction Generator will throw an error message when you try to generate the Reuse Instruction:

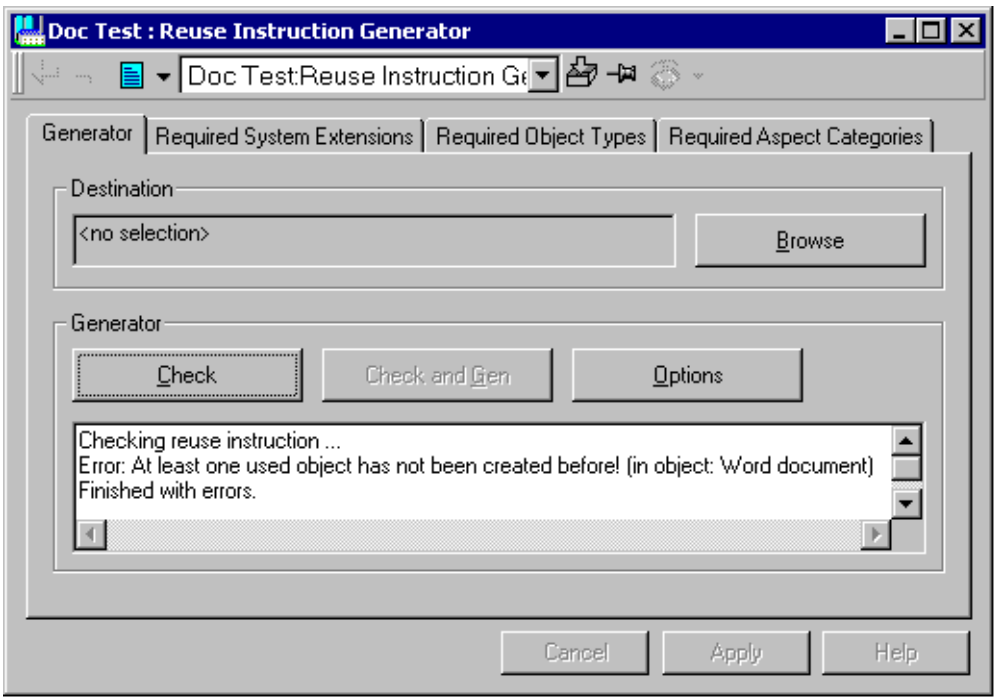


Figure 221. Error Caused by a Red Reference

Architect - Working with Substitution Variables

A Reuse Instruction can have a number of Substitution Variables. These so called *Substitutions* are variables which can only be defined within a Reuse Instruction during a design session (architect mode) and will be substituted with real values during execution of Reuse Instructions (build mode) of the Reuse Assistant. For more information about using Substitutions in the build session refer to Architect - Working with Substitution Variables.

The following figure shows an example how Substitution variables are displayed in the Plant Explorer preview area:

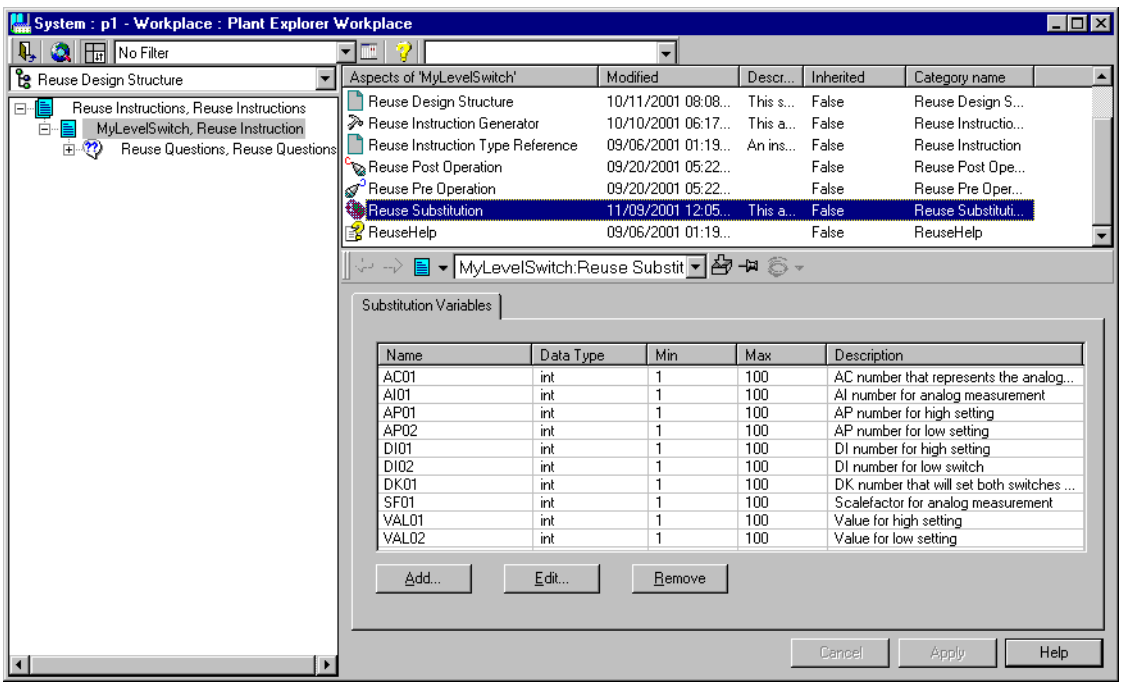


Figure 222. Example of Substitutions displayed in Plant Explorer

General

Substitution Variables are stored in an aspect of category “Reuse Substitution” which belongs to the Aspect Object of type “Reuse Instruction”. The Substitutions must have a variable name and a data type. They can also have minimum and maximum allowed values and they can have a textual description. The minimum allowed value must be less or equal to the maximum allowed value.

The allowed characters for the variable name are a-z, A-Z, 0-9, and _. The name must start with a character and the length has to be between 1 and 255.

The supported data types for Substitution Variables are:

- String (the minimum and maximum length of the string value can be set)
- SignedInteger (value from -2.147.483.647 to 2.147.483.647)
- Real (value from 1.17E-38 to 3.4E+38)
- Bool (value “TRUE” or “FALSE”)
- AspectPath (value also handled as the “String” data type)
- ObjectPath (value also handled as the “String” data type)
- ValuePath (value also handled as the “String” data type)
- Integer (value from -32768 to 32767)
- UnsignedInteger (value from 0 to 65535)
- Double (value from 2.2E-308 to 1.7E+308)
- Word (value from 0 to 65535)
- DoubleWord (value from 0 to 4294967295)
- CBM Name (value handled as “String” data type with IEC 61131 name conventions)
- Time (value handled as “String” data type with format: 0d0h0m0s0ms)
- Date And Time (value handled as “String” data type with format: YYYY-MM-DD-hh:mi:ss.ttt)

Usage

Substitutions can be used in operations of the following categories:

- Reuse Answer Operations
- Reuse Pre Operations
- Reuse Post Operations

To use a substitution, the operation has to declare that it is using the substitution. The Reuse Assistant in Build Mode requires user input for all substitutions which

- Have been declared as “used” by one of the operations associated with the selected answers
- Have been declared as “used” by the “Reuse Pre Operations” or by the “Reuse Post Operations”

The following chapters describe how to add, remove and edit Substitution Variables.

Adding Substitutions

Open the preview area of the Reuse Substitution aspect and press the **Add** button. The following dialog appears:

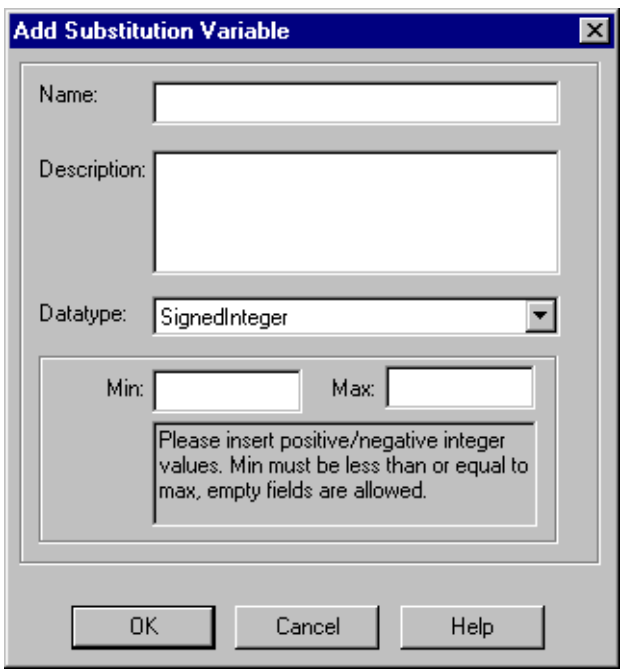


Figure 223. Add Substitution Variable Dialog

You have to enter a variable name (matching the naming constraints described in Architect - Working with Substitution Variables) and select a data type out of the

available list of data types. Optionally you can also define the minimum and maximum allowed values and a description for the substitution.

If the selected data type is AspectPath, ObjectPath or ValuePath you get additional input possibilities. The Add Substitution Variable Dialog looks as following:

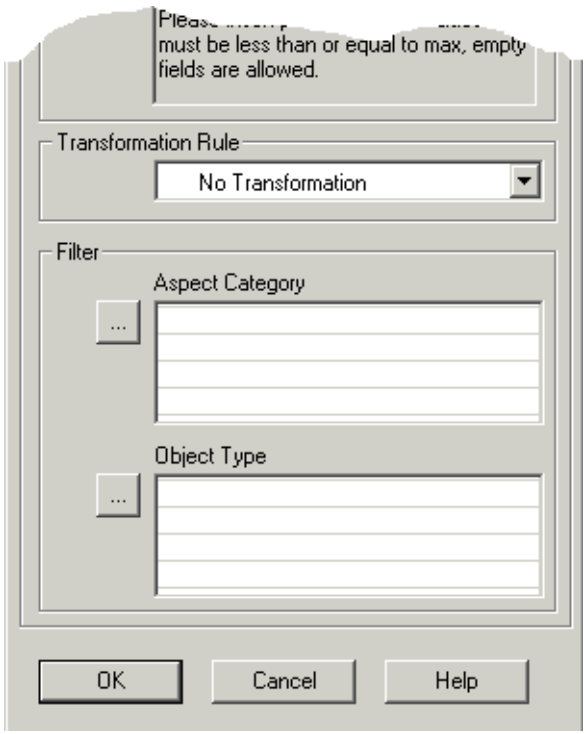


Figure 224. Advanced Add Substitution Variable Dialog

Now you can select a rule out of the available list of Transformation Rules. The Transformation Rule is used in Builder to manipulate a value for a Substitution so, that we get a valid Control Builder M representation.

Transformation Rules are implemented for the data types:

- ObjectPath
- ValuePath

The following table demonstrates the transformation mechanisms implemented in Reuse Assistant Builder.

Table 24. Transformation Mechanisms

Data type	Transformation Rule	Value from Pick Dialog	Transformed Value
ObjectPath	Application Assignment	[Direct][Control Structure]Root/Control Network/CBMProject/Applications/CBMApplication	CBMApplication
ObjectPath	Task Assignment	[Direct][Control Structure]Root/Control Network/CBMProject/Controllers/CBMController/Tasks/Fast	CBMController.Fast
ObjectPath	Library Assignment	[Direct][Control Structure]Root/Control Network/CBMProject/Libraries/ControlStandardLib	ControlStandardLib
ValuePath	IO-Assignment	[Direct][Control Structure]Root/Control Network/CBMProject/Applications/CBMApplication:Application:Variable	CBMApplication:Variable

In Aspect Preview Area of Reuse Substitution each substitution variable name is preceded with an icon that represents the chosen Transformation Rule.

Optionally you can assign filter conditions based on Aspect Categories and Object Types to a Substitution Variable. The buttons in filter section start a structure dialog for Aspect Categories or Object Types. The selected values are shown in the Aspect Category or Object Type list. Delete from these lists is possible via context menu by pressing the right mouse button.

Press the **OK** button and the Substitution will be stored *temporary* in memory. Pressing the **Add** button again to be able to enter the next Substitution Variable.

To store all entered variables permanently you have to press the **Apply** button (refer to Figure 225) of the preview area of the Plant Explorer. Then all added Substitutions will be stored permanently to the disk.

By pressing the **Cancel** button no entered substitutions will be stored but will be removed from the list instead.

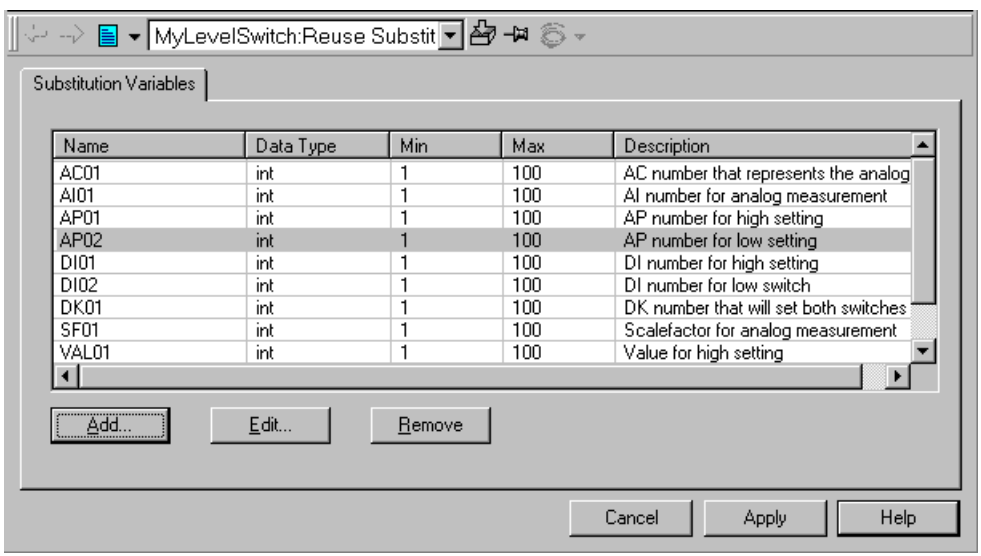


Figure 225. Help, Apply, and Cancel Button of Aspect Preview Area

By pressing the **Help** button of the aspect preview area the on-line help for substitutions appears as in the following figure:

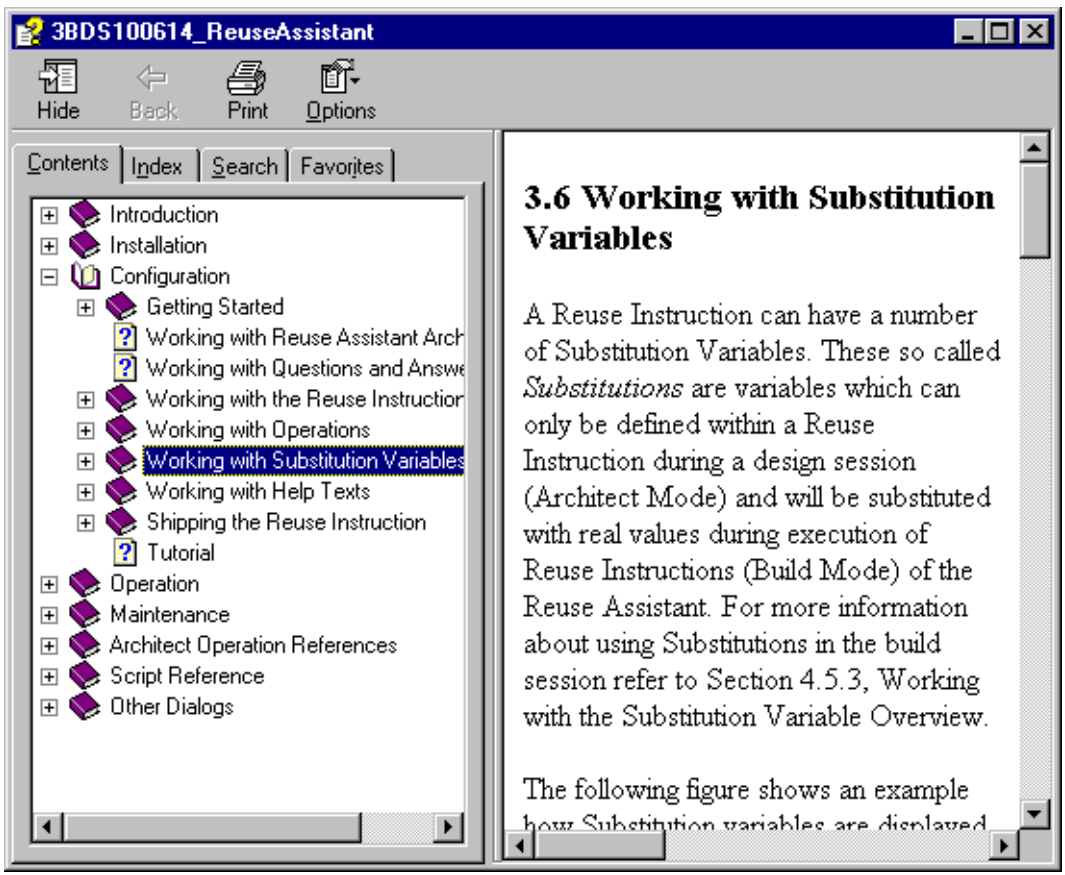


Figure 226. On-line Help of Substitutions

Editing Substitutions

To edit a Substitution Variable open the preview area of the “Reuse Substitution” aspect, select one variable from the list and press the **Edit** button. The following figure appears:

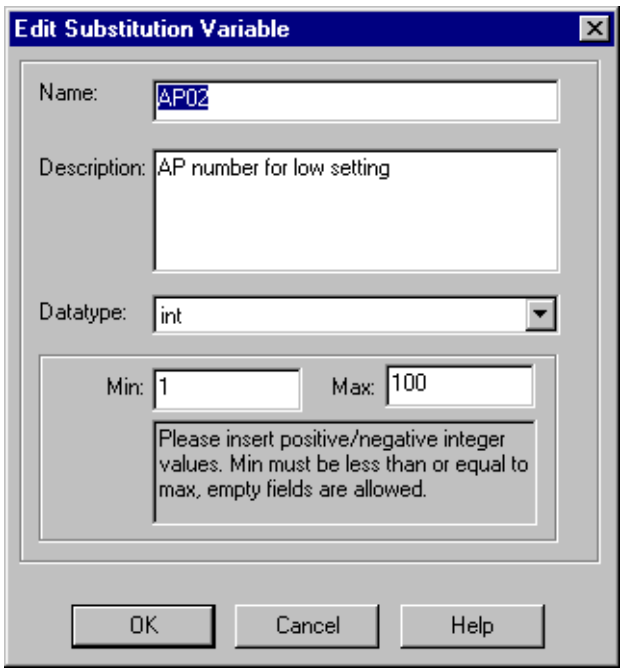


Figure 227. Edit Substitution Variable Dialog

Now you can change the name, description, data type and the minimum and maximum allowed values.

If the selected data type is AspectPath, ObjectPath or ValuePath you get additional input possibilities (described in Adding Substitutions).

Press the **OK** button and the modifications will be stored *temporary* in memory. Pressing the **Edit** button again you can modify the next Substitution Variable.

To store all modified variables permanently you have to press the **Apply** button (refer to Figure 225) of the preview area of the Plant Explorer. Then all modified Substitutions will be stored permanently to the disk. By pressing the **Cancel** button no changes will be stored and the list will be updated.

Deleting Substitutions

To delete a Substitution Variable open the pre-view area of the “Reuse Substitution” aspect, select one variable from the list and press the **Remove** button. The following confirmation dialog appears:

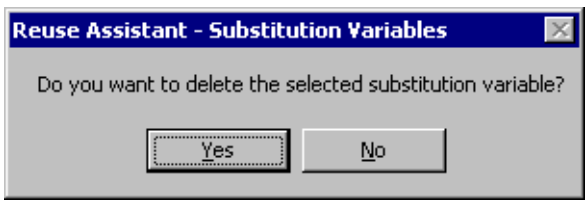


Figure 228. Confirm Delete Substitution Dialog

Press the **Yes** button and the Substitution will be deleted *temporary* in memory. Pressing the **Remove** button again you can delete the next Substitution Variable.

To make the deletion permanent you have to press the **Apply** button (refer to Figure 225) of the pre-view area of the Plant Explorer. Then all temporary deleted Substitutions will be removed permanently from the disk. By pressing the **Cancel** button all changes will not be stored and the list will be updated.

Copying Substitutions

It is possible to copy and paste a substitution variable. By selecting a substitution variable from the list of substitutions and pressing the right mouse button the

following context menu appears shown in Figure 229:

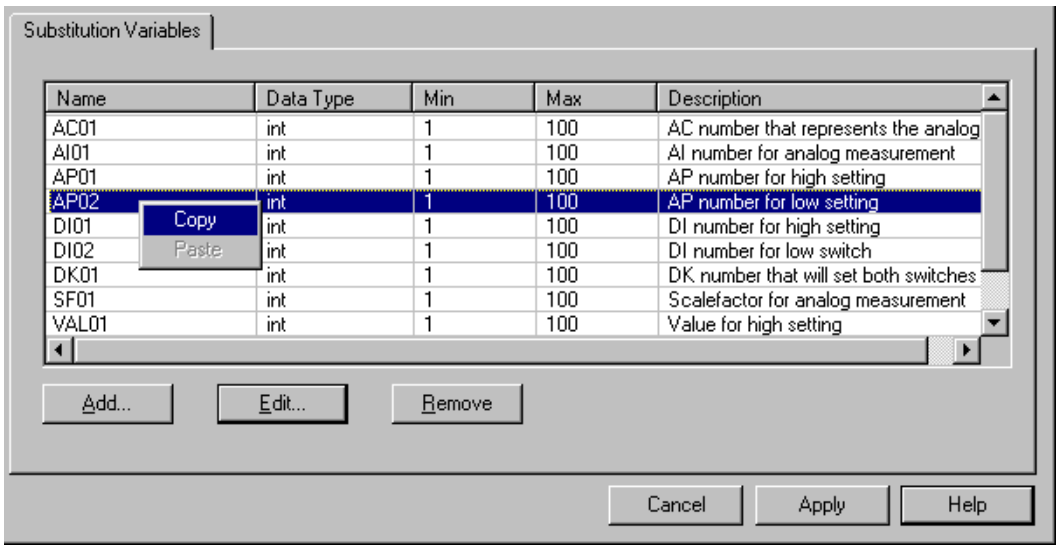


Figure 229. Context Menu for Substitutions

Select the **Copy** menu item to copy the selected substitution and then select the **Paste** menu item to create a new substitution variable with the values copied from the selected one. The variable name is automatically renamed. The string "_Copy" is appended to the existing name. If the name is not unique, then the "_Copy" string is appended once more.

You also can paste the substitution variable into another "Reuse Substitution" aspect located in another Reuse Instructions.

Using Substitution

Substitutions can be used in all operations, including your own script code. To use a substitution within an operation use the following syntax:

```
$Substituion Variable$
```

The following figure shows an example.

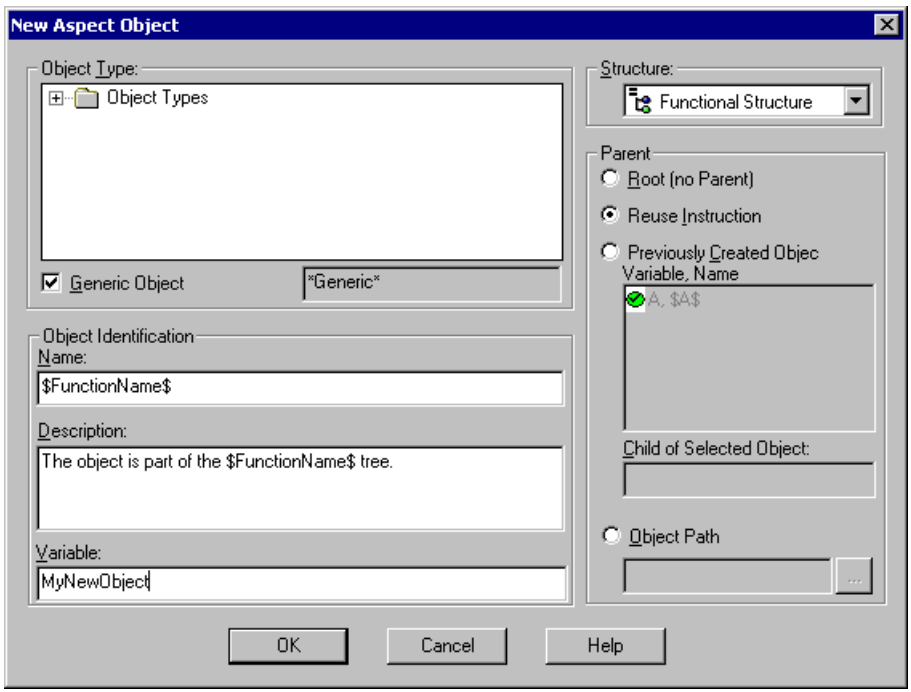


Figure 230. Using a Substitution

As shown in the example, you can incorporate substitutions into strings. Since the "$" character is used to indicate the begin and the end of a substitution, you need to escape this character by a preceding "\" if you want to use it as part of the string.

Substitutions can also be used in numeric expressions. Example:

```
SIN($Ti$) + 3 * ($Tg$ + 0.5)
```

In this example, the substitutions with the names Tg and Ti are used.

In some cases you may even want a string value to be the result of a numeric evaluation. In this case, you need to tell the system "this is not a string, this is an expression". Start your expression with the "#" character to do this.

Example:

```
#"A" + CSTR(SIN($Tg$)) + "B"
```

Assuming that the value of the substitution variable Tg is 1.000, this statement will compose the following string:

```
A0.841470984807897B
```

Please note that you also need to escape the "#" character by a preceding "\" if it should be used in the text itself.

So, what remains to be explained is the usage of substitution in the script code that you can write yourself. All the values that your user enters for the substitution variables are stored in an array called **Substitutions**. This array is a member of the object **Ra** which is always present in all Reuse Instructions. You can access the actual value that the user has given for the substitution using the name of the substitution variable as array index. Example:

```
strfunctionname = Ra.Substitutions("FunctionName")
```

This code fragment will read the actual value of the substitution with the name "FunctionName" and store this value in the variable "strfunctionname".

Architect - Working with Help Texts

You can supply additional information to your users by adding a help text to:

- the Reuse Instruction
- each of the Questions
- each of the Answers.

You define the help text by editing the content of the Reuse Help aspect. Each Reuse Instruction, Reuse Answer and Reuse Question object has a Reuse Help aspect.

The following figure shows, how the different help texts are displayed to the user of the Reuse Instruction.

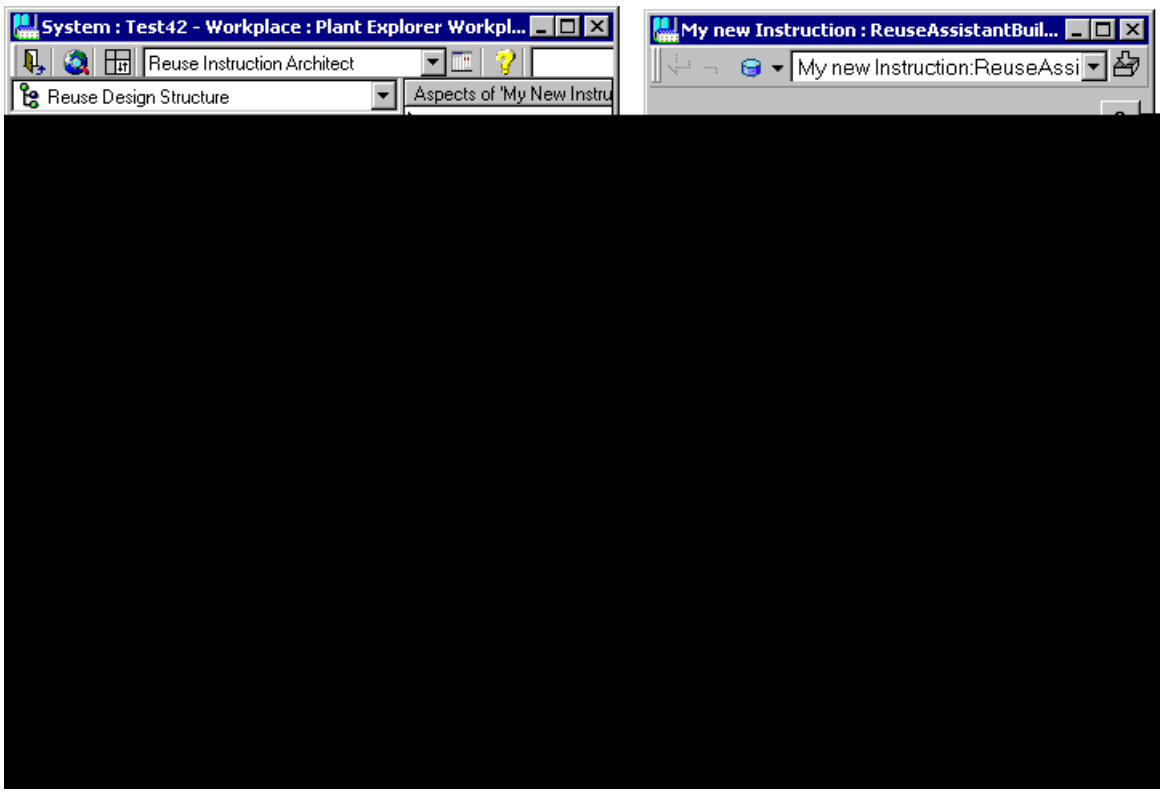


Figure 231. Help Text Definition

The Reuse Help aspect offers you two different views to edit the help text:

- a WYSIWYG view
- a Source text view

The following figure shows the Reuse Help aspect with the WYSIWYG view active.

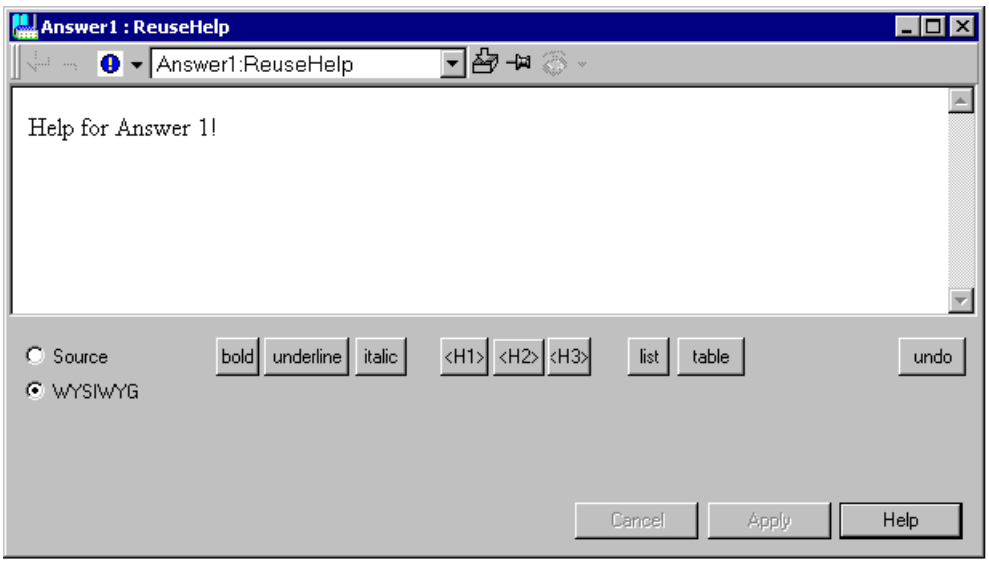


Figure 232. Reuse Help in WYSIWYG View

The following table describes the buttons on this dialog.

Table 25. Buttons on the Reuse Help Aspect Page

Button	Description	Related HTML Tag
Source	Switch the view from WYSIWYG mode into source mode. This will display the HTML source text.	
WYSIWYG	Switch the view from source mode into WYSIWYG mode. This will display the HTML in the same way as the user will see it.	

Table 25. Buttons on the Reuse Help Aspect Page (Continued)

Button	Description	Related HTML Tag
Bold	This button will mark the selected text bold.	<B>
Underline	This button will underline the selected text.	<U>
Italic	This button will change font for the selected text to italic.	<I>
H1	This button will change the selected text to a level 1 header.	<H1>
H2	This button will change the selected text to a level 2 header.	<H2>
H3	This button will change the selected text to a level 3 header.	<H3>
List	This button will change the selected text into a list of list of elements.	<LI>
Table	This button will insert a table into the help test.	<TABLE>, <TR>, <TD>
Undo	This button will undo the last operation.	

To use the full power of HTML, you to switch the view to source mode. In this mode, you can enter any HTML constructs.

Please observe the following limitations:

Links to Other WEB Pages, Graphics, Sound or Video Files

To use a link to any of this type of data you have to make sure that the data can also be accessed when the Reuse Instruction is exiting on the computer of your user. The Reuse Assistant itself does not take care of any of this linked data.

To use this kind of data in a help text you can:

- copy the data manually to the computer where the Reuse Instruction is executed.

- expose the data on a WEB server that be accessed from the computer where the Reuse Instruction is executed.

Working with Tables

Tables are one of the most commonly used HTML functions. The Reuse Assistant Help aspect supports table with the Insert Table dialog.

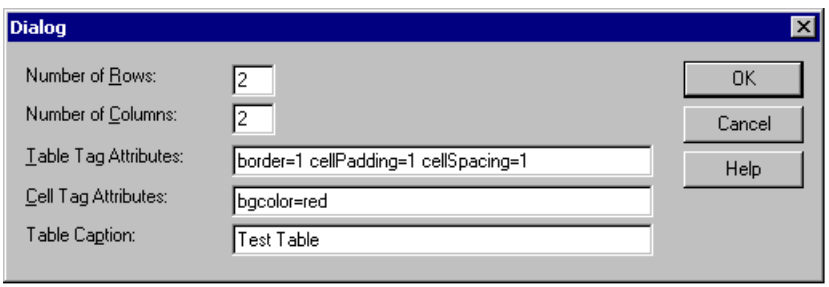


Figure 233. Table Editor

The following table shows the input fields of this dialog.

Table 26. Fields on the Insert Table Dialog

Field	Description	Supported Value
Number of Rows	Enter the number of rows that the new table should contain.	Integer, >= 0
Number of Columns	Enter the number of columns that the new table should contain.	Integer, > 0

Table 26. Fields on the Insert Table Dialog (Continued)

Field	Description	Supported Value
Table Tag Attributes	Enter the HTML attributes for the table	alin=left|center|right background=<link to background image> bgcolor=<color> border=<border width, integer value> borderColor=<color> borderColorDark=<color> borderColorLight=<color> cellPadding=<integer value> cellSpacing=<integer value> class=<table class name> heigth=<table height> width=<table width>
Cell Tab Attributes	Enter the HTML attributes for the cells of the table	align=left|center|right background=<link to background image> bgcolor=<color> border=<border width, integer value> borderColor=<color> borderColorDark=<color> borderColorLight=<color> cellPadding=<integer value> cellSpacing=<integer value> class=<table class name> height=<table height> vAlign=top|middle|bottom|baseline width=<table width>
Table Caption	Enter the title text for the table	

Architect - Shipping the Reuse Instruction

After you have designed and tested your Reuse Instruction, you need to prepare it to be shipped to your customer. What you will ship to customer is an AFW file containing everything that she/he needs to execute the Reuse Instruction.

The AFW file is generated by the Import / Export tool. You can start this tool from the Start menu of your computer.

Start > Programs > ABB Industrial IT 800xA > System > Import Export

The Import / Export tool shows the following user interface.

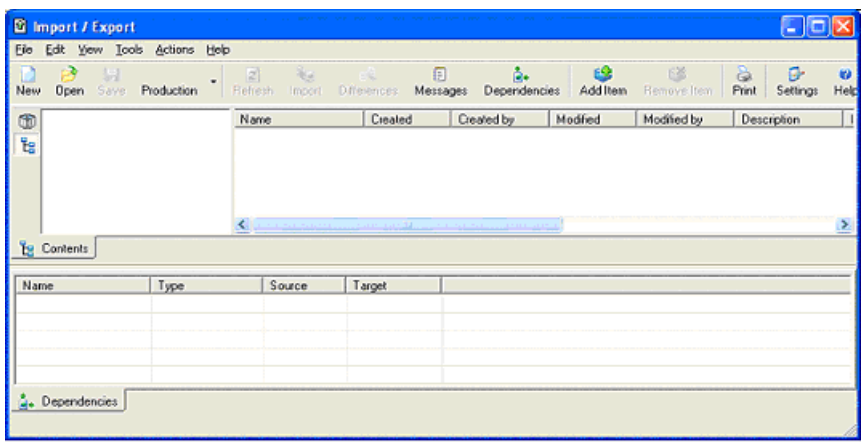


Figure 234. Import / Export Tool

Aspect Object can be exported from the Plant Explorer by drag and drop. Drop the object into the Import / Export tool.

So, it is easy to export something from the Plant Explorer, be what do you need to export, so that your Reuse Instruction can be executed at your customers' computer?

You need:

- The object type with the Reuse Instruction (see Working with Reuse Instruction Generator).

- All the object type that you have used in your Reuse Instruction that are not present on the customers' computer.
- All aspect categories you have used in your Reuse Instruction that are not present on the customers' computer.

The object type with the Reuse Instruction is the one you specified with the Reuse Instruction Generator (see Working with Reuse Instruction Generator). So, locate this object in the object type structure and drop it into the Import / Export tool.

The next more complicated task is the object types that are used by your Reuse Instruction. The Reuse Instruction Generator helps you to find out which object types have been used by the Reuse Instruction with a special list (see Required Object Types). The Reuse Instruction Generator fills this list from information is extracts from the different operations.

Most likely, you will not need to export all of the object types that are present in the list.

There are 3 groups of object types:

1. Object types that are part of System 800xA platform itself
2. Object types which have been loaded by a system extension (like AC400 Controller Integration)
3. Object types which have been specially designed to be used with this Reuse Instruction.

You do not need to worry about the object types that belong to the first group. Since they belong to System 800xA platform itself, you can be sure that they exist on the computer where the Reuse Instruction is executed.

For the object that belong to the 2nd group, you just need to make sure that the user of your Reuse Instruction has loaded the system extension that contains the object type. See Required System Extensions about how to define a dependency on a system extension.

So, what remains are the object types of the 3rd group. Include this object types into your AFW file.

So far we have covered the Reuse Instruction itself and the object types that are used by the Reuse Instruction. The remaining part is the aspect categories that are used by the Reuse Instruction.

Again, the Reuse Instruction Generator helps you to identify the used aspect categories with a special list (see Required Aspect Categories). As already described for the object types, there are 3 groups of aspect categories:

1. Aspect categories that are part of the System 800xA platform.
2. Aspect categories that are loaded by a system extension
3. Aspect categories that have specially designed to be used for the Reuse Instruction.

Again, you only need to export the aspect categories of the 3rd group. However, it is very unlikely that you will design new aspect categories for your Reuse Instruction. In fact, today the only tool which defines new aspect categories “on the fly” is the Parameter Manager.

So, if you have used self defined aspect categories, include these categories in your export file.

Save the AFW file on disk and send it to your user.

Required System Extensions

As described in Architect - Shipping the Reuse Instruction, you specify that your Reuse Instruction requires a certain system extension to be loaded. This is done using the tab Required System Extension of the Reuse Instruction Generator.

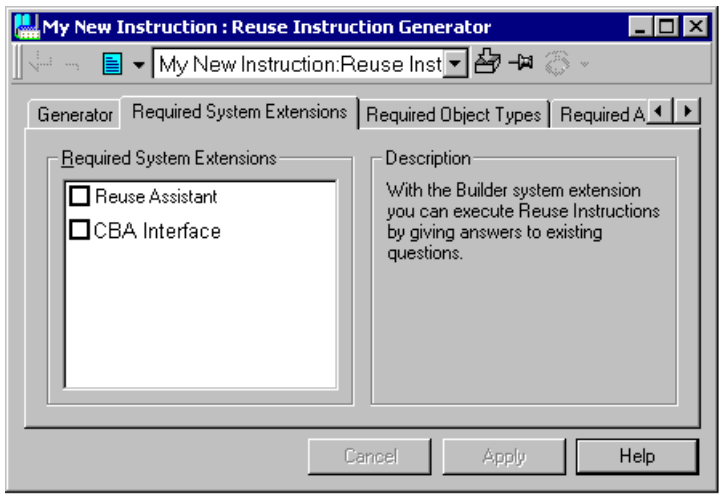


Figure 235. Required System Extensions

Select the system extensions that you require to be loaded into the system where the Reuse Instruction is executed. The Reuse Assistant will check if these system extensions are present before executing the Reuse Instruction (see Consistency Check).

The Instruction Generator lists all the system extensions that are loaded into your system.

Required Object Types

As described in Architect - Shipping the Reuse Instruction, the Instruction Generator can display a list of all object types that are used by the operations that are defined for the Reuse Instruction.

The following figure shows the list that is displayed be the Reuse Instruction Generator.

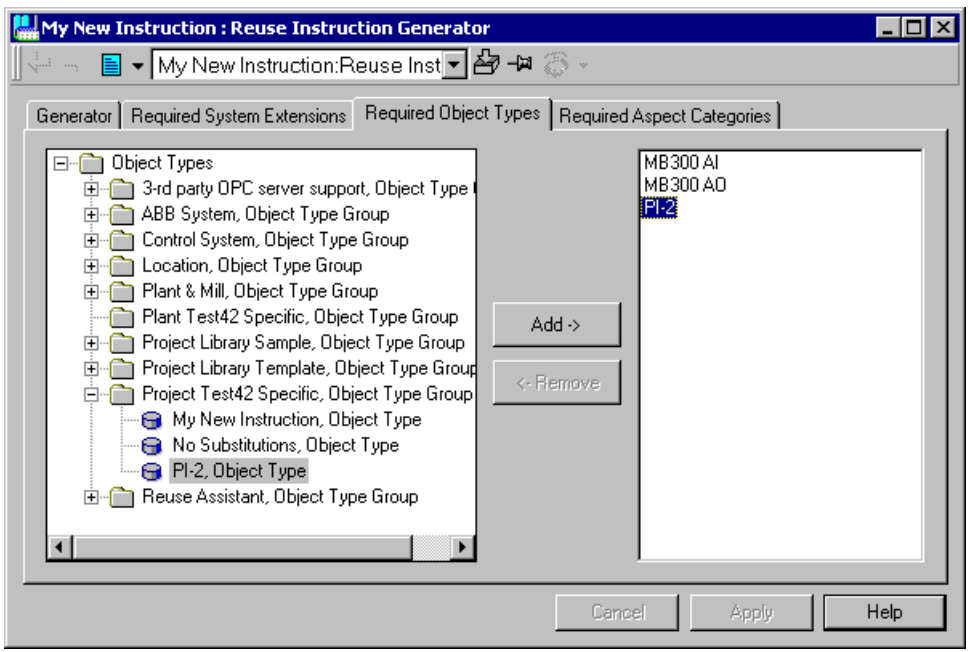


Figure 236. Table for Required Object Types

This table lists all the object types that Instruction Generator has detected are needed by your Reuse Instruction. However, this list may not be complete. Specially, object type that you have used in your hand written script code are not automatically detected.

You can use the **Add** button to insert these object types into the list.

The Reuse Assistant checks if all the required object types are present in the system where the Reuse Instruction gets executed (see Consistency Check for details).

You can use this table also to create the AFW file. Select the object in the list on the right side of the dialog. The system will then automatically select the object in the tree. You can now export the object type from the tree using drag and drop.

Required Aspect Categories

As described in Architect - Shipping the Reuse Instruction, the Reuse Instruction Generator creates a list of all aspect categories that used by the operations of the Reuse Instruction. The following figure shows this list.

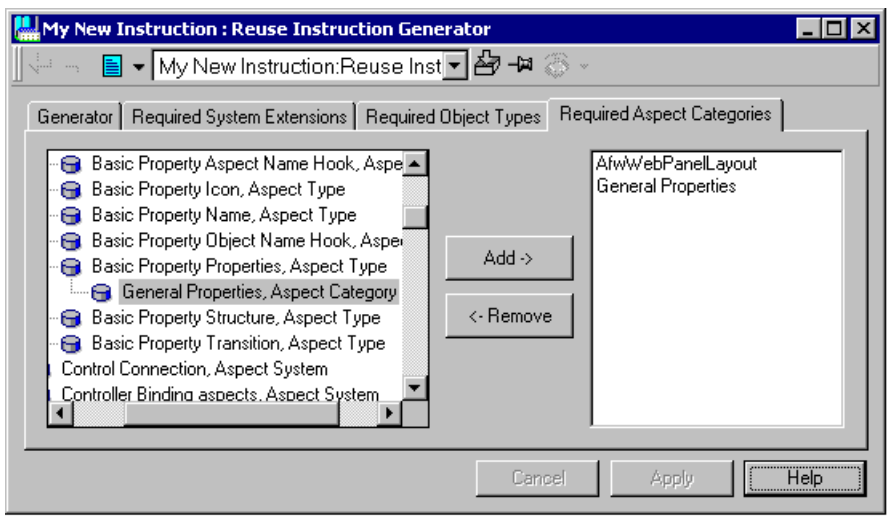


Figure 237. Required Aspect Categories

This table lists all the aspect categories that Instruction Generator has detected are needed by your Reuse Instruction. However, this list may not be complete. Specially, aspect categories that you have used in your hand written script code are not automatically detected.

You can use the **Add** button to insert this aspect categories into the list.

The Reuse Assistant checks if all the required aspect categories are present in the system where the Reuse Instruction gets executed (see Consistency Check for details).

Architect - Transactions

The Reuse Instruction Generator produces code that:

- executes all operations except for Modify Property and the Control Builder M specific operation “Add Variable”, “Add Parameter”, “Modify Alarm” in a single transaction
- executes all Modify Property and Control Builder M specific operations after the transaction has been committed without an additional transaction.

However, transaction handling has its price. In the case of the Reuse Instruction, this means that you need to design the instruction with the transaction in mind. Here are the rules you have to follow.

Access to Object via Name

You can not access an object via the name unless the transaction has been committed.

Example:

Assume that you are creating an object in the functional structure as part of your Reuse Operation. The name of the object is “Kalle” and it is placed at root level of the functional structure.

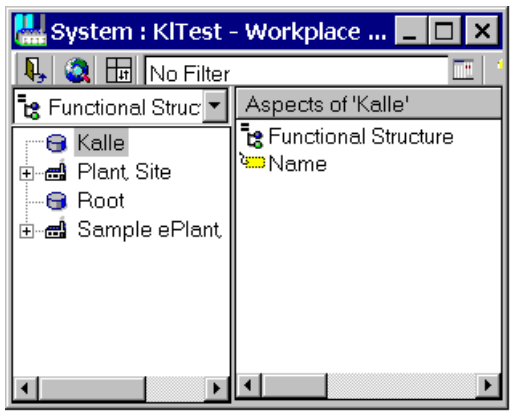


Figure 238. Transactions - Part 1

Assume further, that in a next part of Reuse Instruction you are going to add an aspect to this object.

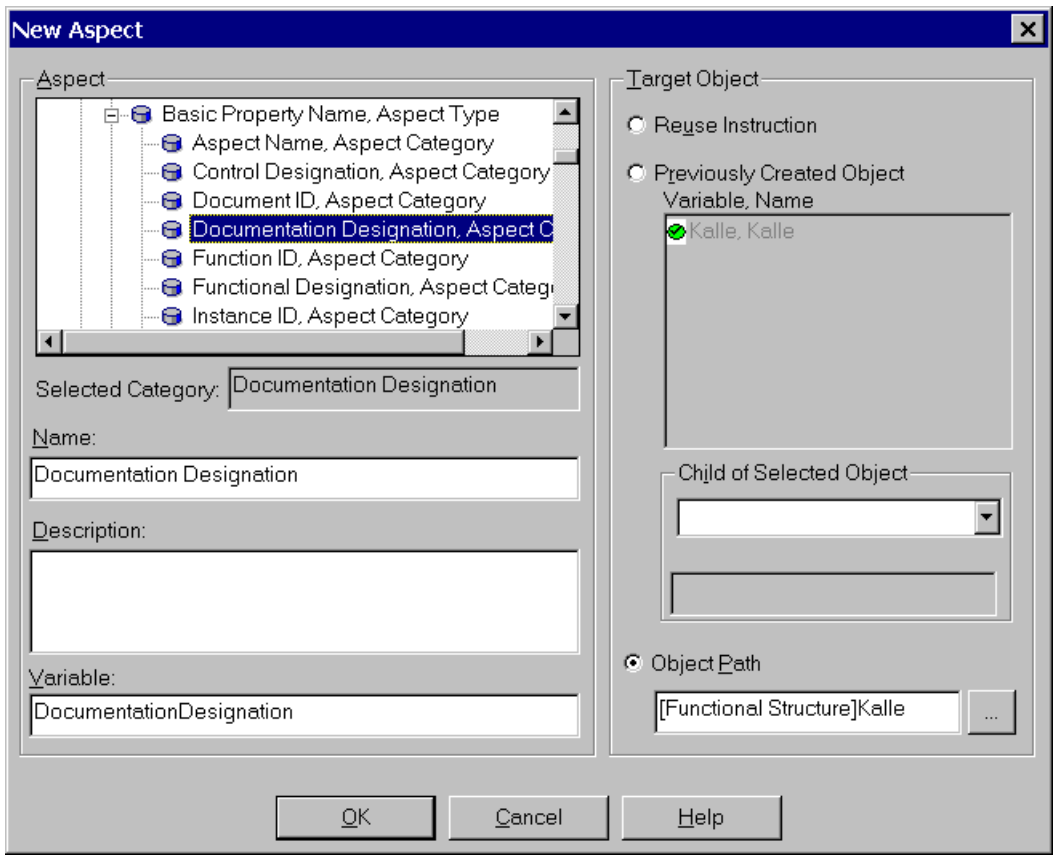


Figure 239. Transaction - Part 2

This operation will always fail! Why? It tries to access an object that is created as part of the same transaction (remember, all operations of one Reuse Instruction are executed in one transaction) via its object name ([Functional Structure]Kalle). To solve this problem, access the object (“Kalle”) via the variable that is assigned to it.

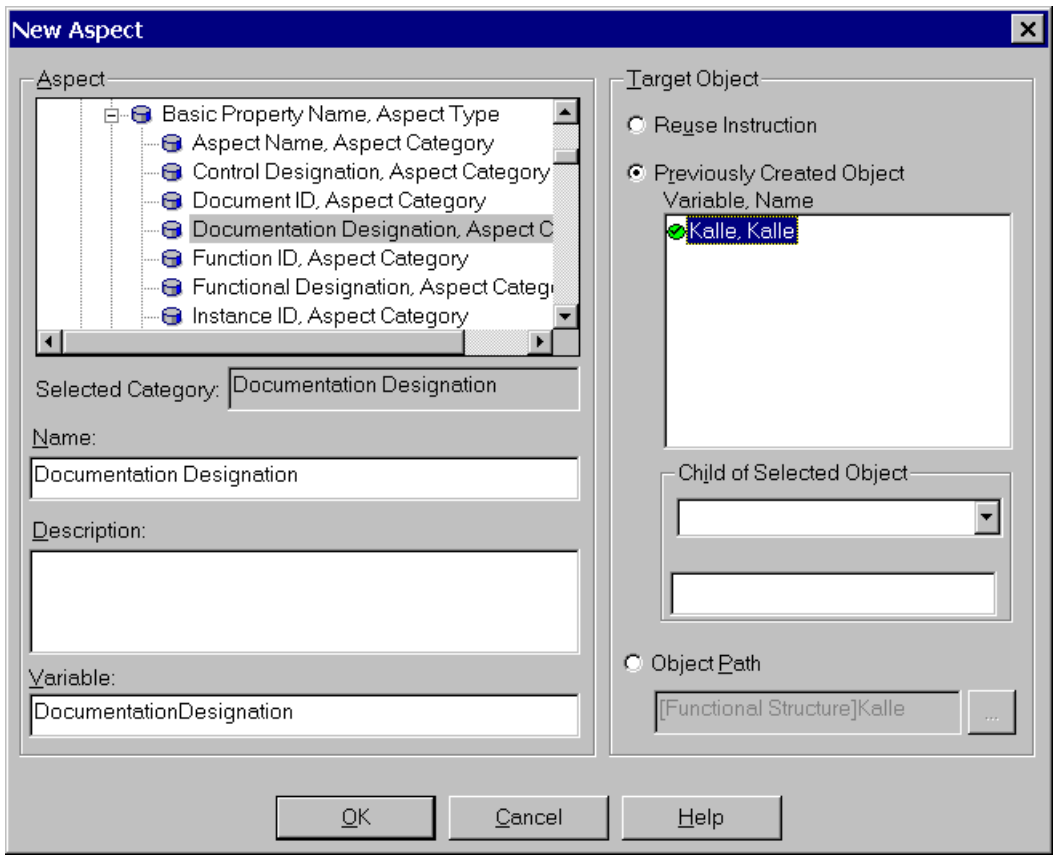


Figure 240. Transaction - Part 3

The same applies to all operations, not just the Create Aspect operation shown in this example.

Architect - Other Dialogs

Reuse Assistant Architect

This subsection lists the description of dialogs which are no covered by other parts of this subsection.

Instruction Generator Options

This dialog lets you change some functions of the Reuse Instruction Generator.

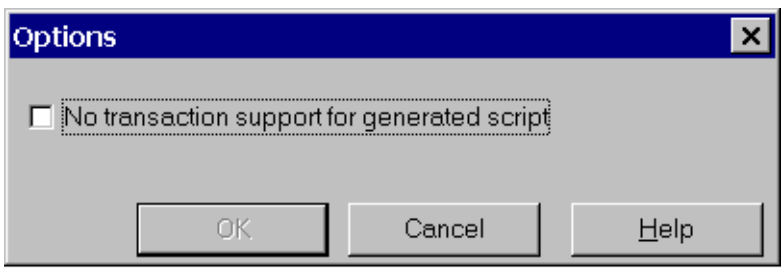


Figure 241. Reuse Instruction Generator Options

If the checkbox is selected, the Reuse Instruction Generator will produce script code that does not include transaction management for the contained operations. This option is useful for debugging the Reuse Instruction.

Some Control Builder Object requires “No Transaction Support”. Control Builder objects (like Control Application) create new objects in the Control Builder which are then reflected into the aspect directory. If your Reuse Instruction accesses one of these objects, you need to set the switch “No Transaction Support” when generating the Reuse Instruction

Instruction Generator New Destination Object Type

This dialog lets you create a new aspect object type which can be used as a destination for your Reuse Instruction.

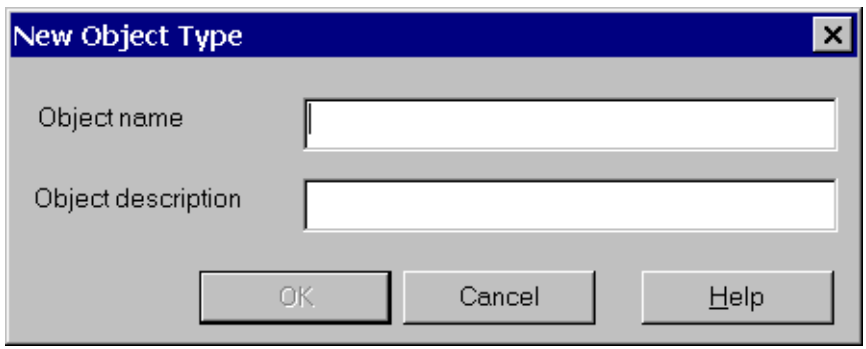


Figure 242. New Object Type

The dialog lets you define the object name and description for a new object type that is used as the destination for the Reuse Instruction.

Browse for Object

This dialog lets you define an object path by browsing for the object in your system. The dialog will compose a string for you that points to the object (the object path). Please keep in mind that exactly this string will be used to locate the object on the computer where the Reuse Instruction is executed.

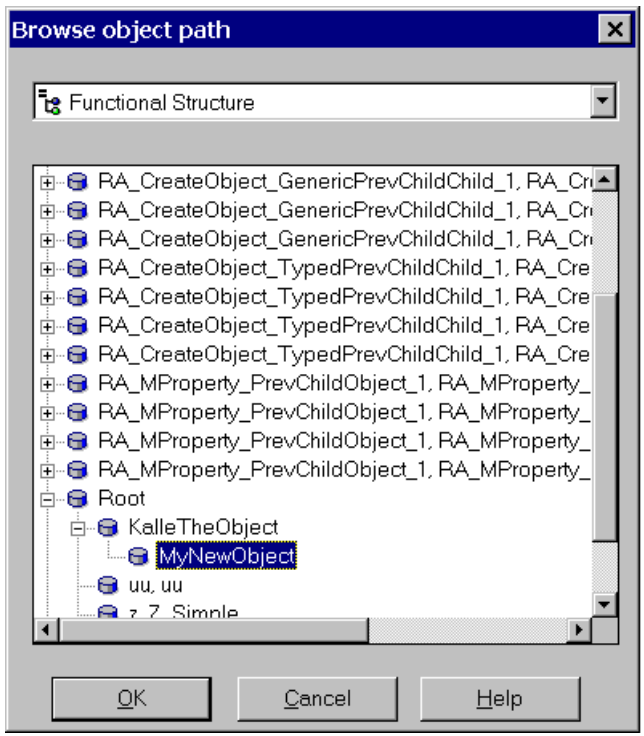


Figure 243. Object Path Browse Dialog

Table 27. Fields on the Browse Object Path Dialog

Section	Button or Field	Description
	Structure Selector	Select the structure where is object is located. The selection will end up in the [structure] part of the object path.
	Object Browser	Select the object. The selection will end up in the body part of the path.

Architect - Tutorial

This subsection describes everything you need to create a very simple Reuse Instruction, consisting of just one question and two answers.

Our Reuse Instruction will create an object of the type “Location” as a new root object in the functional structure. Then it should add:

a. an object of the type “Building” as child to the new root object **or**

b. an object of the type “Room” as child to the new root object.

This should depend on the user’s selection.

So, let’s analyze our problem.

Analyzes

Variants of the Solution

The instruction can produce the following variants of the solution:

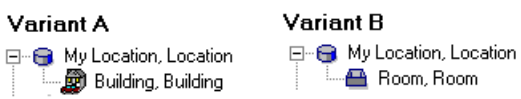


Figure 244. Variants of the Solution

Required Object Types

Within this example we will use 3 new object types: “Location”, “Building” and “Room”.

User Decisions

The user has to decide, if she/he wants a room or a building as part of the location. So, the question we could ask the user would be:

“What should your location consist of?”.

The possible answer would be “A building” (which would result in solution A) or “A room” (which would result in solution B).

System Actions

So, what kind of task you we need to perform to build the two variants? For both, we first need to create the Location object in the functional structure. This action must be done prior to any other action, because the object must exist in order to either place the room or the building object as a child under the location object. So, creating the location object will be part of a Pre Operation.

The other actions (creating the room or the building) are depending on the user selection. They will go into the Answer Operations. In addition, we want the user to enter a name for:

- The location
- The root
- The building

In each case, the operation is a “Create Object” operation.

Order of Questions

Since we only have a single question in the instruction, we do not have a problem with the order of the questions. We just need to decide on which of the answers should be the default answer. In our case, we will create a room by default.

So, after analyzing the problem, we are well prepared to implement the our new Reuse Instruction. But wait, we missed one tiny little detail, the name of the Reuse Instruction. Lets call the Reuse Instruction “Location Generator”.

Implementation

Let’s first design our Location object type. Execute the following steps:

1. Go to the object type structure
2. Create a new object type with the name Location.

The result should look like this:

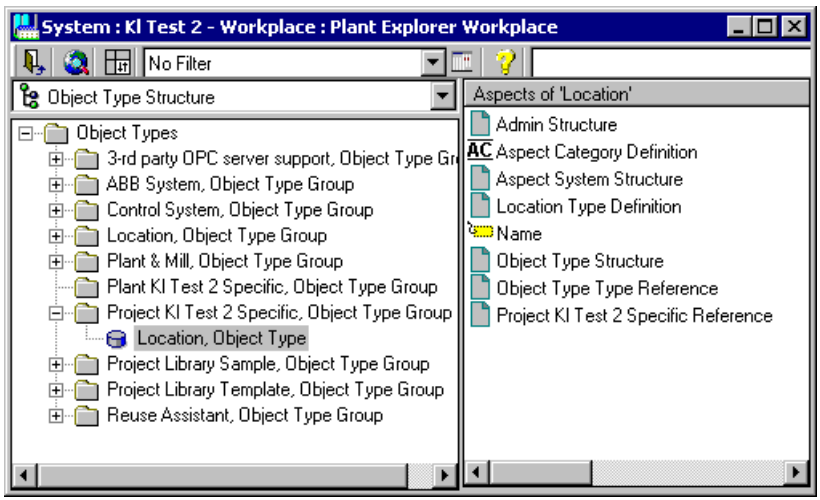


Figure 245. Location Object

The same action then has to be performed for the “Building” and “Room” object types.

The next task is to create the Reuse Instruction itself. Execute the following steps:

1. Go to the Reuse Design Structure.
2. Select the root object with the name Reuse Instructions.
3. Create a new object of the type Reuse Instruction. Give it the name Location Generator.

The result should look like this:

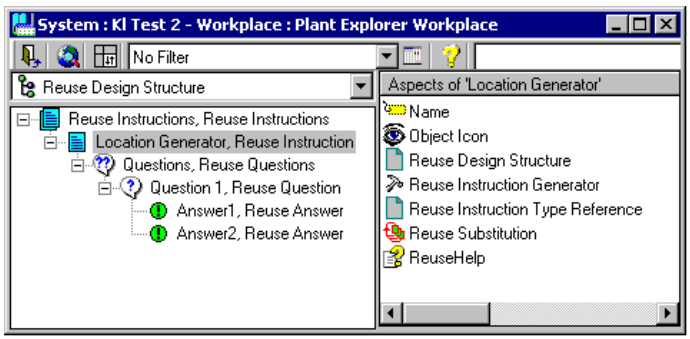


Figure 246. Reuse Instruction

Now we will adopt the existing questions and answers. Execute the following steps:

1. Select the object with the name Question 1. Rename this object to “What should your location consist of?”
2. Select the object with the name Answer 1 and rename it to “A Building“.
3. Select the object with the name Answer 2 and rename it to “A Room”.Mark this answer as the default answer.

Now, your Reuse Instruction should look like this.

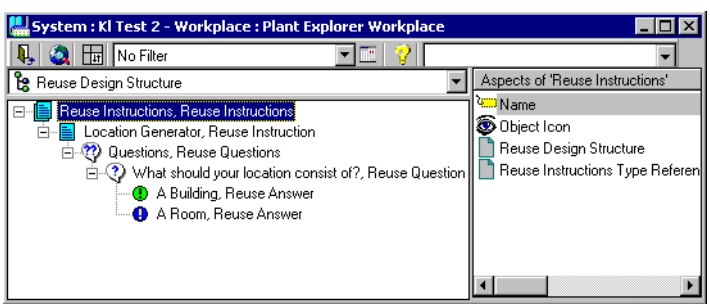


Figure 247. Reuse Instruction with Texts

Now we can add our operations to the instruction. But first, we need to take care of the substitutions. Execute the following steps:

1. Select the object with the name Location Generator.
2. Select the Reuse Substitution aspect
3. Add 3 new substitutions to the list. The data type is String. The minimal required string length is 1. The names are:
    a. LocationName
    b. RoomName
    c. BuildingName

Now add the first operation to the Reuse Instruction. Execute the following steps:

1. Select the object with the name Location Generator
2. Add a new aspect of the category Reuse Pre Operations to this object.

3. Select the Reuse Pre Operations aspect and create a new “New Object” operation. Enter the following values into the dialog:

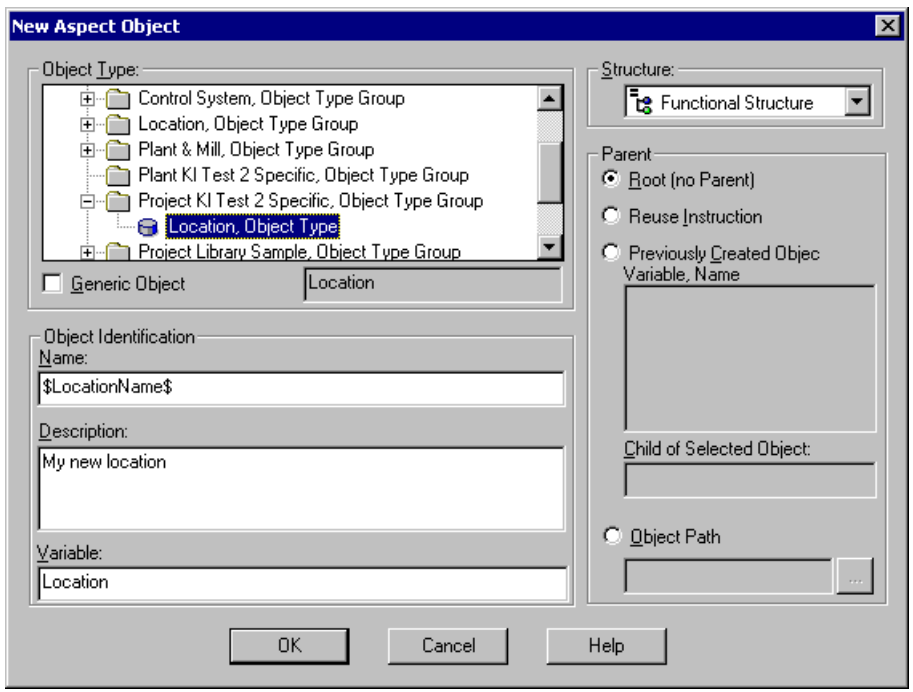


Figure 248. Parameters for New Location

Now add the operations to create the Room object and the Building object to the corresponding answers. Execute the following steps:

1. Select the object with the name A Building. Open the Reuse Answer Operations aspect.

2. Add a “New Object” operation. Enter the following values into the dialog:

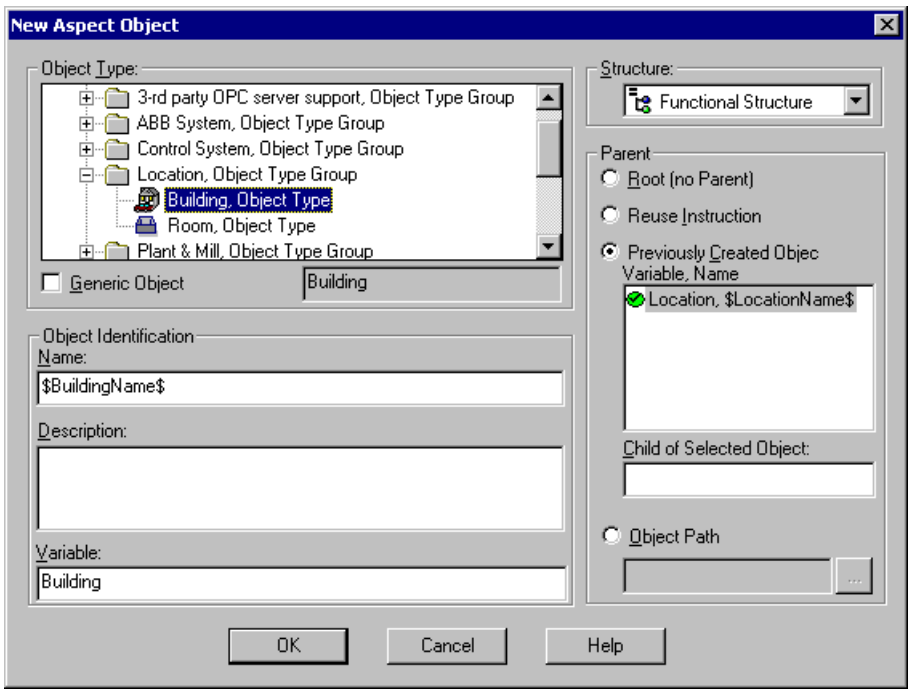


Figure 249. Parameters for New Building

3. Select the object with the name A Room. Open the Reuse Answer Operations aspect.
4. Add a “New Object” operation. Enter the following values into the dialog:

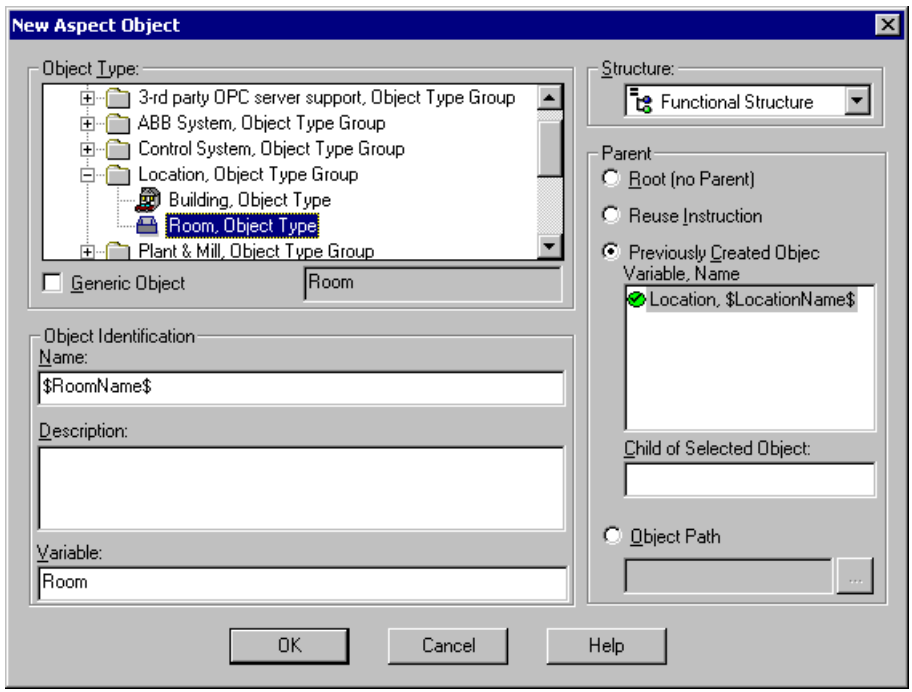


Figure 250. Room Parameters

Now our little instruction is ready to be compiled. We still need to decide on the name of the object that our user should create in order to execute the Reuse Instruction. To keep it simple, we will call this object Location Generator as well.

Execute the following steps:

1. Select the object with the name Location Generator. Open the aspect Reuse Instruction Generator. Click on the Browse Button.
2. Create a new object type with the name Location Generator in any of the available object type groups.
3. Press Apply.

4. Check the Check & Gen button to generate the instruction.

Now, the Reuse Instruction Generator aspect should look like this.

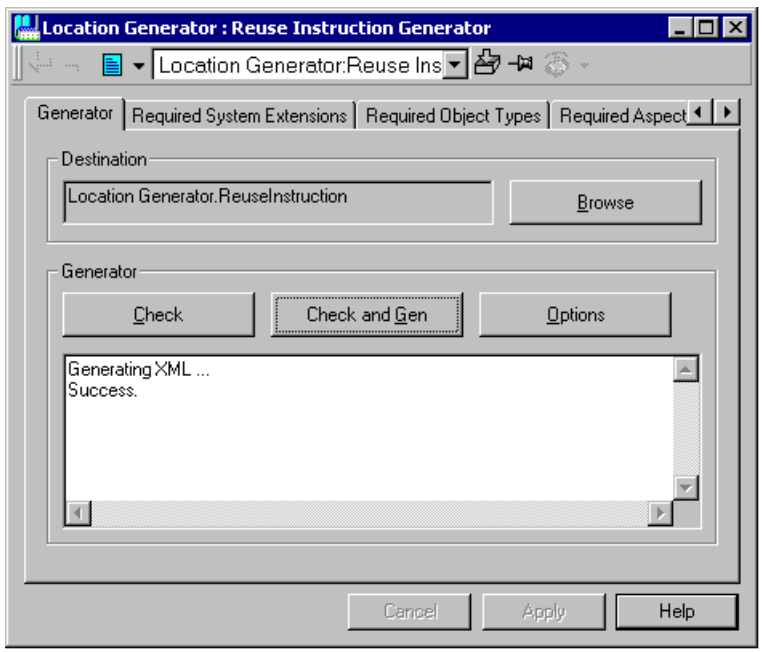


Figure 251. Instruction Generator

Testing the Reuse Instruction

As any software, Reuse Instructions are potentially buggy. To check, if the instruction to doing what it should do, we need to create an instance of the instruction and to execute it.

Execute the following steps:

1. Go to the Function Structure.
2. Create a new root object of the type Location Generator

3. Open the aspect with the name ReuseAssistantBuilder. Now test all (two) variants.

Shipping the Reuse Instruction

To send out the Reuse Instruction to any of your users, you need to pack it into an AFW file. This file is generated by the Import / Export tool (part of System 800xA platform).

Execute the following steps:

1. Start the Import / Export tool
2. Drop the object type with the Reuse Instruction (Location Generator) into the Import / Export tool.
3. Drop the object type Location into the Import / Export tool.
4. Save the content on disk. Select the file name Location Generator.

Now you can send this file to your user. You may also put this file under revision control.

Architect - Script References

Working With VB Script

This chapter contains information for advanced users who want to include their own script code into a Reuse Instruction.

This chapter is not an introduction into the VBScript language itself, it only covers the Reuse Instruction specific issues of VBScript. For a description of the VBScript language itself, please check one of the following books.

Table 28. Books on VBScript

Title	Publisher	ISBN Number
VBScript in a Nutshell A Desktop Quick Reference	O'Reilly	1565927206
VBScript Programmer's Reference	Wrox Press	1861002718

In addition to knowing the VBScript language it is essential to know the OLE Automation Model of the System 800xA platform. This model is described in *Industrial^{IT} 800xA - System, Programmer's Guide*.

All the operations that are implemented in the Reuse Assistant are working with this OLE Automation Model.

The Reuse Assistant executes the operations contained in the Reuse Instruction using Mircosoft's scripting engine. This software is part of the Microsoft Windows operating system. As a script programmer you have full access to all functions of this scripting engine. In addition, the Reuse Assistant provides the following functions to you as a script programmer:

- Access to a properly initialized ABBSystem object. This object is the entry point into the world of aspects and objects. See Global Variables.
- Read access to all substitution values. See The Object RA.
- Read access to all the answers that the user has selected. See The Object RA.
- Read and write access to an array containing the GUIDs of all objects that are created as part of the Reuse Instruction. See The Object RA.
- Read and write access to an array containing the GUIDs of all aspects that are created as part of the Reuse Instruction. See The Object RA.
- A number of VBScript subroutines which makes easier to created objects, aspects and to modify properties. See Global Functions.

Coding Conventions

To avoid conflicts between the script code that the Reuse Assistant generates for you and your own script code, it is important to stick to some simple coding conventions.

- Do not use any local or global variable starting with the “RA”.
- Do not define any FUNCTION or SUB starting with the “RA”
- Do not change the content of any variable that is used by the Reuse Assistant. All this variables are string with “RA”.

Global Variables

The Reuse Assistant defines the following global variable.

Table 29. Global Variables

Variable	Data Type	Description
RaSystems	IABBSystems	This variable points to the COM object “ABB.ABBSystems”. The properties and methods of this object are described in *Industrial*^{IT} *800xA - System, Programmer’s Guide* and further documents related to this guide.
RaSystem	IABBSystem	This variable points to a COM object containing the ABB System object of the system in which the Reuse Instruction is executed.
RaTransaction	ABBTransaction	This variable points to the transaction, in which all the operations are executed.

The Object RA

The object with the name RA is always present in the script of the Reuse Instruction. This object exposes the following properties.

Table 30. Properties of the RA Object

Name	Parameter	Data Type	Access	Description
Ra.Answers	String, GUID of the Reuse Answer object.	BOOLEAN	R	Returns TRUE if the Reuse Answer (identified by the object GUID) has been selected by the user.
Ra.Instruction ObjectId	-	String	R	Returns the object Id (GUID) for the aspect object from which the Reuse Assistant Builder was started.
Ra.Instruction StructureId	-	String	R	Returns the Id (GUID) of the structure that was selected in the Plant Explorer when the Reuse Instruction is executed.
Ra.Result	-	BOOLEAN	R/W	If this value is set to FALSE, the Reuse Assistant will abort the transaction and report an error to the user. This is value is TRUE after all operations have been executed, the Reuse Assistant will commit the transaction and report “Success” to the user.
Ra.Substitution	String, name of the substitution	Variant	R	Returns the value of the substitution.
Ra.SystemId	-	String	R	The GUID of the system in which the Reuse Instruction is executed.

The object RA exposes the following methods.

Table 31. Methods of the RA Object

Name	Parameters	Description
Ra.AspectAspId	[in] String Variable Name [out, retval] String Aspect ID	This method retrieves the aspect Id (GUID) of an aspect that has been saved before using the method SaveAspect.
Ra.AspectObjId	[in] String Variable Name [out, retval] String Object GUID	This method retrieves the object Id (GUID) of an aspect that has been saved before using the method SaveAspect.
Ra.ObjectId	[in] String Object Variable Name [out, retval] String Object GUID	This methods returns the GUID of an object that has been created by the Reuse Instruction. You have to give the variable name of the object as input parameter.
Ra.ReportError	[in] String MsgId [in] String Source [in] String Text	Reports the text together with the source of the error and the message ID as an error into the system event and alarm list.
Ra.ReportEvent	[in] String Message	Writes the given text as a event to the system event and alarm list.
Ra.SaveAspect	[in] String Variable Name [in] String Object GUID [in] String Aspect GUID	This method saves the object ID (GUID) and the aspect ID (GUID) of a given aspect for later use. The two IDs are connected to the variable name of the aspect.
Ra.SaveObject	[in] String Object Variable Name [in] String Object GUID	This method saves the GUID of a of an object that has been created by the Reuse Instruction. The GUID gets connected to the variable name of the object.

Global Functions

Before executing the Reuse Instruction, the Reuse Assistant adds a number of VBScript SUB routines to the script code. This SUB routines are used be the operations to perform there tasks.

You may use this SUB routines in you own script code as well.

The following table lists the available SUB routines.

Table 32. SUB Routines

Name	Parameters	Description
RaCreateAspect	[in] Object target object [in] String aspect category id [in] String aspect category name [in] String name [in] String description [out, retval] Object created aspect	This function creates a new aspect.
RaModifyProperty	[in] Object target aspect [in] String property name [in] Variant value	This SUB routine changes the value of a property.
RaInsertChildObject	[in] Object object to be inserted [in] Object structure cursor with the insertion point [in] String GUID of the destination structure [in] String name of the destination structure	This SUB routine inserts an object into a structure at a given location. The location is pointed out by the structure cursor.

Table 32. SUB Routines (Continued)

Name	Parameters	Description
RaOverrideAspect	[in] Object target object [in] String aspect name [in] String aspect category GUID [in] String id (GUID) of the object type of the target object [in] String aspect name of the new aspect [in] String description text of the new aspect [out, retval] Object the newly created aspect	This function creates a new aspect by overriding an existing, inherited aspect.
RaSetSuperType	[in] Object target object type [in] String id (GUID) of the super type	This SUB routine defines the super type for the target object.
RaSetAspectCategory Control	[in] Object target object type [in] String id (GUID) of the aspect category for which the aspect control should be defined. [in] Bool create when object is created [in] Bool propose when object is created [in] Int minimal number of aspects of the given category [in] Int maximal number of aspects of the given category	This SUB routine defines the aspect control for the target object type.

Table 32. SUB Routines (Continued)

Name	Parameters	Description
RaChildControl	[in] Object target object type [in] String id (GUID) of the child object type [in] String id (GUID) of the structure for which to define the child control [in] Int minimal number of child objects [in] Int maximal number of child objects	This SUB routine defines a child control rule for the given object type.
RaAspectControl	[in] Object target object type [in] String target aspect name [in] Bool flag to indicate that the aspect should be used as template [in] Bool flag to indicate that the aspect should be copied when an instance of the type is created. [in] Bool flag to indicate that the aspect should be inherited when is object is created	This SUB routine defines the aspect control for the target object type.
RaFindChildPath	[in] Object structure cursor with the start point [in] String child path [out, retval] found child object	This function finds child objects of uncommitted objects. See Architect - Transactions for a discussion on transaction issues.

Examples

The following example shows a user defines operation, implemented as a script block. The operation creates a aspect object in the functional structure. The type of the aspect object is coming from a substitution variable called “ObjectType”.

```
DIM kl_strmessage0
DIM kl_strmyobject1
DIM kl_strmyobject0
DIM kl_curmyobject0
DIM kl_cur0
DIM kl_ObjectTypeName

ON ERROR RESUME NEXT
REM General variable prefix is: kl
REM ----------------------------------------
REM CreateObject
kl_strmyobject0 = "MyObject"
kl_strmyobject1 = "Description text for MyObject"
REM find the given object in the object type structure
kl_ObjectTypeName=Ra.Substitutions("ObjectType")
SET kl_cur0 = RaSystem.StructureCursor

REM Creating new object
SET kl_curmyobject0 = kl_cur0.NewRootObject(kl_strmyobject0,
kl_strmyobject1, "Functional Structure", kl_ObjectTypeName)
IF NOT Err.Number = 0 THEN
  ON ERROR GOTO 0
  CALL Err.Raise(1, "My Operation", "" + "Failed to create
object."
  END IF
CALL Ra.WriteEvent("Root object created." + "::" +
kl_strmessage0)
CALL Ra.SaveObject("MyObject", kl_curmyobject0.Object.Id)
```

The more interesting parts of the code are marked bold.

Builder - Overview

In the operation phase of the Reuse Assistant, a Reuse Instruction is applied to a specific case. That is, the user of the Reuse Assistant gets an Instruction that was created in the Architect Mode, answers the questions posed by the Instruction, specifies values of the substitution variables, and executes the Instruction so that it builds the appropriate Aspect Objects. This operation phase is referred to as the **Build Mode** of the Reuse Assistant.

Builder - Getting Started

It is likely that there is documentation that has been written to describe a certain Reuse Instruction. Such specific documentation should be used as the primary description of the use of the instruction, whereas this general user's guide should be used secondarily to support that specific documentation with information about how the Reuse Assistant works as a general product.

The starting point of the Reuse Assistant in Build Mode is to obtain the Reuse Instruction. The reuse instruction may have been developed at a distant location and sent to the location where it is to be used in the Build Mode. This is the case when instructions are developed by central groups and distributed to many plant engineering groups. In this situation, use the Import / Export application as described in Importing an Instruction

If the Reuse Instruction has already been imported, start as described in Creating an Instance of an Instruction

The Instruction may have been developed on the same system on which the Build Mode is being used. In this case, start as described in Creating an Instance of an Instruction.

User Interface

The Build Mode of the Reuse Assistant has its own user interface. This user interface is the window of the ReuseAssistantBuilder Aspect. It has various views which are detailed in detail later in this document. The Wizard and the Tree views are two alternative views that guide the user through the questions and substitution variables of the Reuse Instruction. Each of these two views and its interaction is described in detail in corresponding later subsections. The ReuseAssistantBuilder

Aspect window also has two more special purpose views. The Substitution Variable Overview shows a table of all pertinent substitution variables and the Option view allows the user to set options to control how some procedures are done.

Figure 252 shows the Reuse Assistant Build Mode user interface. The toolbar just below the title bar and the command buttons at the bottom of the window are common to the various views of the user interface.

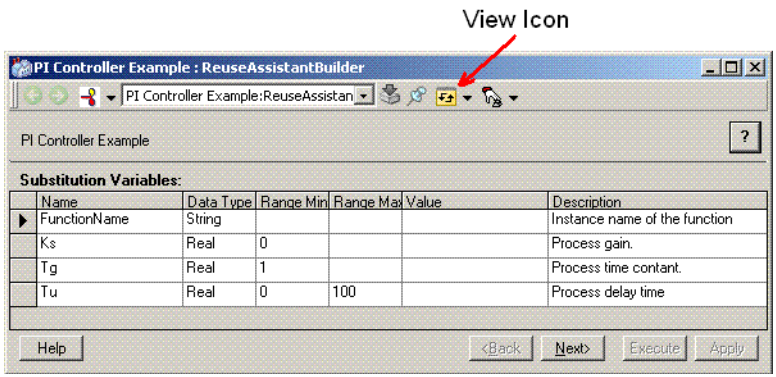


Figure 252. Reuse Assistant Build Mode User Interface

The view icon on the toolbar is significant within the Reuse Instruction Build Mode user interface. By clicking on the down arrowhead to the right of the view icon, a dropdown menu appears to allow switching to the four different views that were described above.

The other tools on the toolbar are standard controls of the Plant Explorer which generally allow navigation from one Aspect to another.

The “Help” command button is at the bottom left of each view of the ReuseAssistantBuilder. Clicking on this button displays a pop-up window with help for the Reuse Assistant. This is help for the Reuse Assistant product itself, not for the specialized Reuse Instruction. This help contains the same information as this present document. (There are “?” buttons in some of the views that provide information about the specialized Reuse Instruction.)

The “Apply” command button stores the current choices of Answers and values of Substitution Variables. Then the ReuseAssistantBuilder Aspect can be closed and

the data will still be available when it is again started. Any answer choices or substitution variable value changes that are made after the last “Apply” or “Execute” will be discarded if the ReuseAssistantBuilder Aspect is closed without saving. This allows the user to make temporary changes and avoid applying them until they are deemed desirable to apply. If the ReuseAssistantBuilder Aspect is closed while there are current answer choices or substitution variable value changes that have not yet been applied, then a pop-up dialog appears to give the user the opportunity to save the current choices and values. The “Apply” button is disabled when there are no unsaved answer choices or substitution variable values.

The “Execute” command button directs the Reuse Instruction to store the current choices of Answers and values of Substitution Variables and to construct the objects that it was designed to construct given the choices and values. The “Execute” button is disabled if there is not a complete path of chosen answers or if any of the pertinent substitution variables does not have a value.

Application Start-up

The Build Mode of the Reuse Assistant may use the Import / Export application and it uses the Plant Explorer. These two applications can be found in the Start Menu of a computer which hosts an 800xA workplace.

There must be an 800xA system running before starting either the Import / Export application or the Plant Explorer. If the instruction is to be used in an existing system, then the system must be running. The manager of the project would have information about how to access that system. If, however, the Reuse Assistant is being used in isolation, perhaps as a training exercise, then a new system can be started or an existing one that is used for practice can be used. If a system needs to be started, it can be started with the Configuration Wizard tool. This tool can be started from the Windows Start menu.

Importing an Instruction

It is common for Reuse Instructions to be developed at one site in order to be distributed and used by other sites. Such Reuse Instructions are sent to the user site as files which are to be imported into an 800xA system using the Import / Export application.

To import a Reuse Instruction, run the Import / Export application and the Open command from the File menu. Open the file which contains the Reuse Instruction. The objects in the file will appear in the hierarchy display of the Import / Export application. Figure 253 shows an example of the Import / Export after a file has been opened.

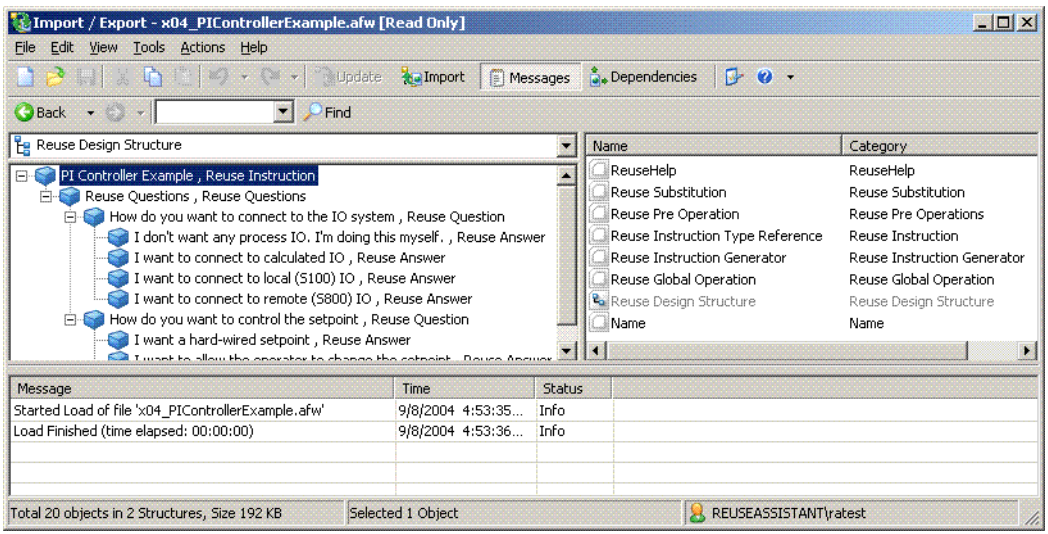


Figure 253. Import of Reuse Instructions

If the developer of a Reuse Instruction provided documentation about how it should be imported, then that documentation should be followed. The Reuse instruction may have been developed in a way that is dependent upon other objects and documentation provided with the Import file should explain any such dependencies.

Import the appropriate objects. There will be an object of category ReuseAssistantBuilder imported into the system.

Once the import is done, the objects are in the library available for all users of the system.

A consistency check should be run after the import. It is possible that the import has not included all of the object types that will be needed to execute the Reuse Instruction. It might be easier to find the missing material while still in the process

of obtaining such material rather than letting the problem go undiscovered until a less convenient time later. Running a consistency check is described in Consistency Check.

Creating an Instance of an Instruction

A Reuse Instruction, as provided to the Build Mode by the Architect Mode, is an Object Type which has an Aspect of type ReuseAssistantBuilder. Figure 254 is an example of such an Object Type, the Filter Cascade.

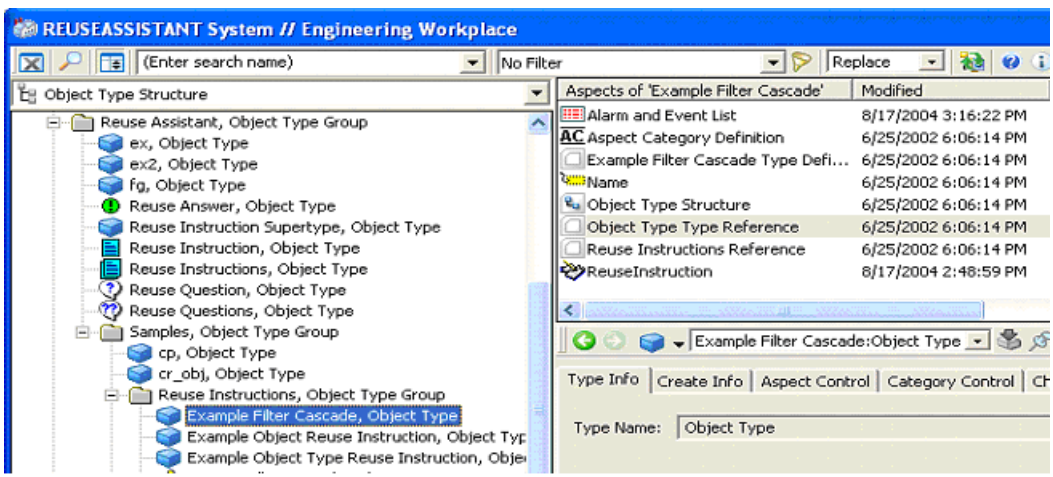


Figure 254. Reuse Instruction in the Object Type Structure

Begin using the Build Mode by creating an object of such a type. Select the structure where the object should be, for example the Functional Structure. Create an object by clicking right in the structure pane and selecting “New Object...”. In the New Object dialog, select the object type that has the Reuse Instruction capability. The new object will be placed in the structure.

Figure 255 shows how the Plant Explorer Workplace might look after an object with the Reuse Assistant capability is created. The new object, “Filter1” has been clicked in the structure tree to put it into focus so that its Aspects are listed in the Aspects pane. The ReuseAssistantBuilder Aspect has been clicked in the Aspects pane to display the user interface of the Reuse Assistant Build Mode in the preview pane.

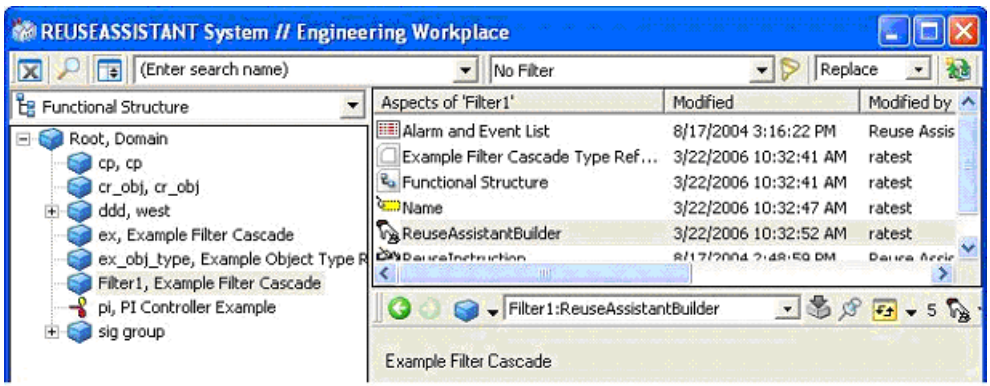


Figure 255. Reuse Instruction

Another Aspect that is always associated with the ReuseAssistantBuilder is the Reuse Instruction. The Reuse Instruction contains the script that was developed using the Architect Mode of the Reuse Assistant. The user of the Build Mode does not change this script. It is not necessary for the user of the Build Mode to understand the contents of script.

Build Mode

The Build Mode of the Reuse Assistant presents the questions and substitution variables of a Reuse Instruction. The user answers the questions and provides values for the substitution variables. When the answers and substitution variable values are complete, the user executes the instruction, and the instruction responds by following its script to create objects and parameterize and interconnect them according to the user's answers and values.

The sequence of questions and the set of pertinent substitution variables depends upon the answers that are chosen. It may help the understanding of the user of the Build Mode to know how these dependencies work, although this knowledge is not required in order to use the Build Mode. The inter-relationships of questions and answers are shown by the example in Figure below. (Figure does not depict a display in the Build Mode.)

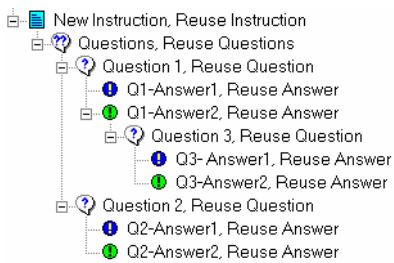


Figure 256. Structure of a Reuse Instruction

As shown in the example, answers are subordinate to questions. Further questions can also be subordinate to answers. In the example, Question 3 is subordinate to the answer Q1-Answer2. By being subordinate to Q1-Answer2, Question 3 will only pertain to the result if Q1-Answer2 is chosen. In the Wizard view that will later be described as a part of the Build Mode, if Q1-Answer1 (not Q1-Answer2) is chosen, then the Wizard will proceed to Question 2 without ever presenting Question 3. Note that the user of the Build Mode does not rearrange the questions and answers.

The answers have Operations associated with them. The Operations of the chosen answers are executed when the Execute command button in the Build Mode is clicked. These Operations create Aspect Objects and Aspects and parameterize them. The Build Mode does not explicitly show the Operations.

A Reuse Instruction also has a set of Substitution Variables. A Substitution Variable sets a value in the Operations. Thus, it can determine what objects are created and how they will be parameterized. In the Build Mode, these details are all handled automatically by the execution of the Instruction. The name and the description of a Substitution Variable and the documentation provided with the individual Reuse Instruction should provide the user of the Build Mode with sufficient information to understand the Substitution Variable.

Substitution Variables are used in pre-operations, post-operations, and the Operations of chosen Answers. Some of the views in the Build Mode display only the Substitution Variables that pertain to a single chosen Answer. Other displays provide an overview of the Substitution Variables with includes only those

Substitution Variables that pertain to the pre- or post-operations or to one of the answers on the path of chosen answers.

The preceding discussion describes the structure of Questions, Answers, Substitution Variables, and Operations that is given to the user of the Build Mode. The Build Mode user does not change these structures, but it may be easier to understand what is happening in the Build Mode if the nature of these structures is understood.

The Build Mode provides two alternative views for the user to answer the questions. These are the Wizard and the Tree views. The Wizard presents the questions in sequence, one-at-a-time. The Tree presents the hierarchy of the questions and answers. The user of the Build Mode can decide between the simplicity of the Wizard and the structure of the Tree.

To use the Build Mode of the Reuse Assistant, open the Plant Explorer Workplace, select the structure and navigate to the object that has the Reuse Instruction, and click on the ReuseAssistantBuilder Aspect so that it is presented in the preview pane. Figure 263 shows an example of how the Workplace might appear as a result of these steps. The Wizard view is the default view.

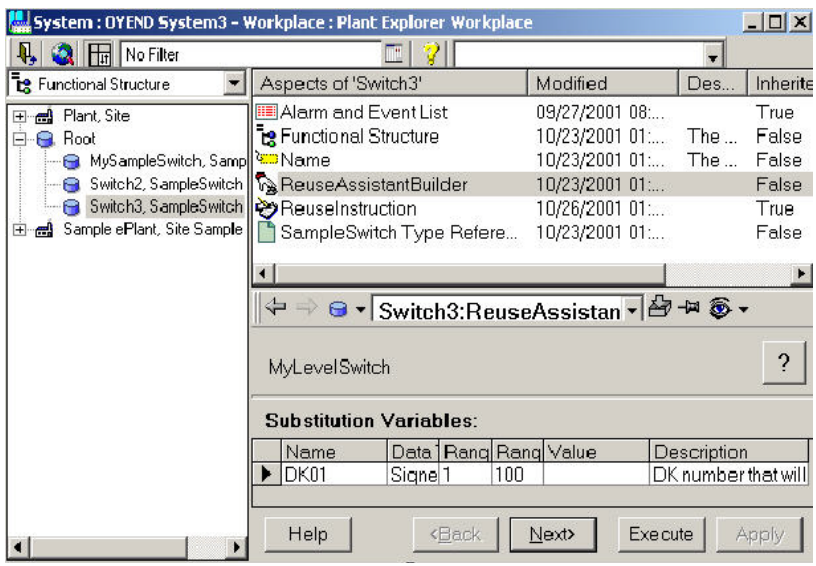


Figure 257. ReuseAssistantBuilder Aspect

Double click (instead of simply left clicking) the ReuseAsistantBuilder in the Aspect pane in order to present it in its own window as in Figure 258.

The figures in the remainder of this document are typically this window rather than the preview pane.

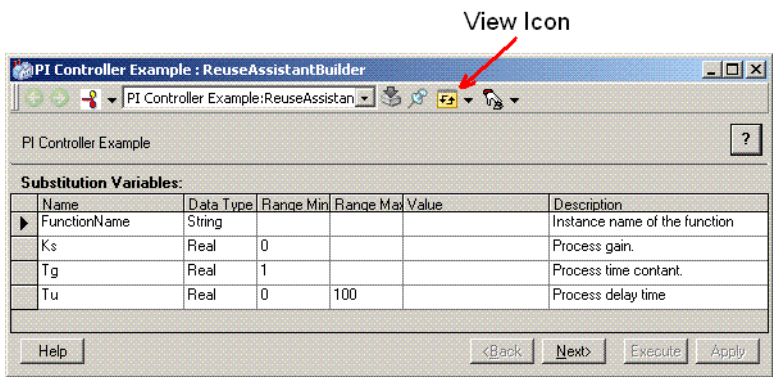


Figure 258. ReuseAssistantBuilder Aspect

The various views can be selected by clicking on the down arrowhead to the right of the view icon. This pulls down a list of four views, among which are the Wizard and the Tree.

Note the toolbar and command buttons that have been described earlier in User Interface.

Reuse Assistant in Build Mode (Wizard View)

The Wizard view presents the Instruction's questions in sequence. The sequence of questions depends upon the answers chosen. A table below contains the list of possible answers and also the substitution variables that pertain to the chosen answer.

The initial display shows the name of the Reuse Instruction and its global substitution variables. Figure 259 shows an example of the initial display.

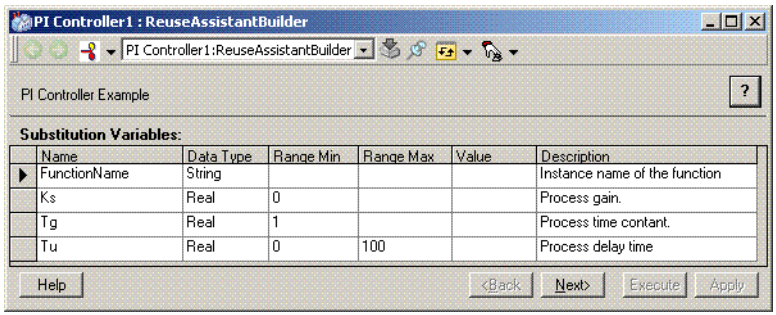


Figure 259. Wizard Initial Display

In this example, the name of the instruction is “PI Controller Example”. The “?” command button to the right of the instruction’s name will display the description of the instruction if clicked. Figure 260 shows the pop-up window that appears when this command is clicked.

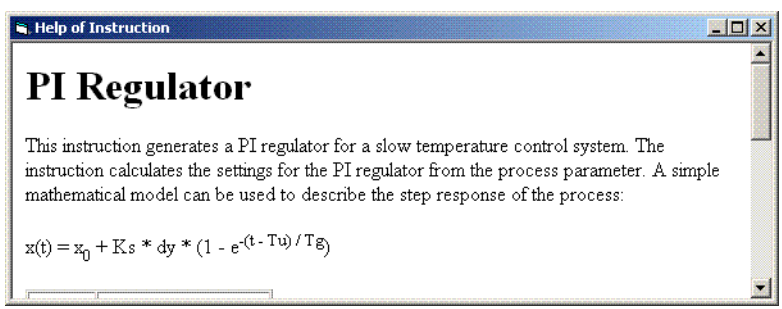


Figure 260. Reuse Instruction Description

The table in the bottom of the initial display (Figure 259) of the Wizard lists the global Substitution Variables of the Reuse Instruction. The global Substitution Variables are used by the Instruction’s pre- and post-operations. Substitution

variables can be given values now, after all of the questions are answered, or at any time by switching to the Substitution Variable Overview.

The “<Back” command button is disabled for the Reuse Instruction name display.

Click the “Next>” command button to move from the initial display to the first question. The display will appear similar to that shown in Figure 261.

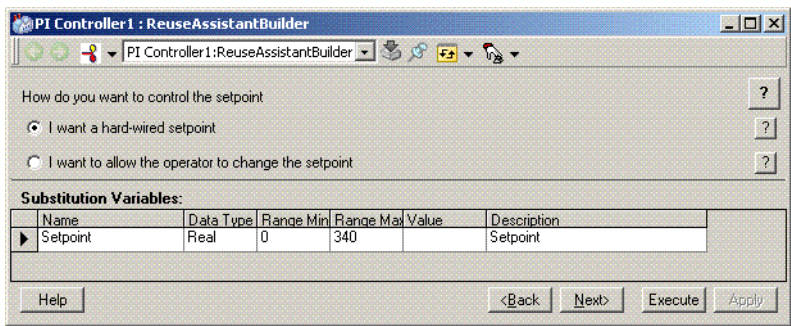


Figure 261. Question in the Wizard View

The question is followed by the choices of answers. One answer may be chosen by default or no answer may initially be chosen.

The “?” command buttons by the question and by each of the answers will pop up a display of the help for the question or answer that was provided by the architect of the Instruction. No “?” button is shown if the corresponding question or answer has no help to display.

Below the answers the substitution variables that are used by the chosen answer are shown. If a different answer is clicked, then the table of substitution variables immediately updates with the corresponding set of variables.

Click the “<Back” command button to return to the preceding question, if desired. From the first question, the “<Back” button goes again to the name of the Reuse Instruction.

If no answer is chosen, then the “Next>” command button is disabled.

Click the “Next>” command to proceed to the next question.

The sequence of questions can change depending upon the answers that are chosen. The structure of Reuse Instructions and questions that are dependent upon preceding answers was explained in Build Mode. An understanding of that structure is not necessary in order to use the Reuse Assistant, but may be helpful.

When the last question in the sequence is being displayed and the “Next>” command button is clicked, then the Reuse Assistant displays a table of all of the pertinent Substitution Variables as shown in Figure 262.

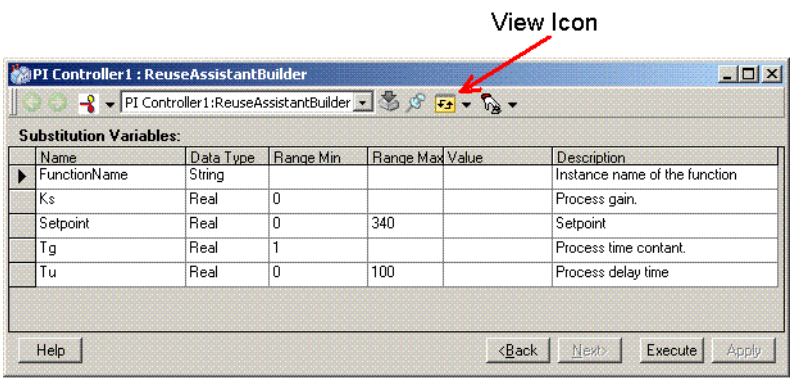


Figure 262. Substitution Variable Overview of Wizard View

The Substitution Variable Overview shows only the pertinent Substitution Variables. These are the ones that are used in the pre- or post-operations and the ones that are used by the chosen answers. The only difference between this Overview and the Substitution Variable Overview that is gotten by selecting it from the view icon list is that this one has the “<Back” button to return to the last question.

Reuse Assistant in Build Mode (Tree View)

The Tree View presents all of the questions of the Reuse Instruction in a hierarchy. A table below the list of possible answers shows the substitution variables. Build Mode earlier described how to navigate to the Tree view. It is one of the views that can be accessed by selecting it from the drop-down list by the view icon in the toolbar.

Figure 263 shows an example of the Tree view.

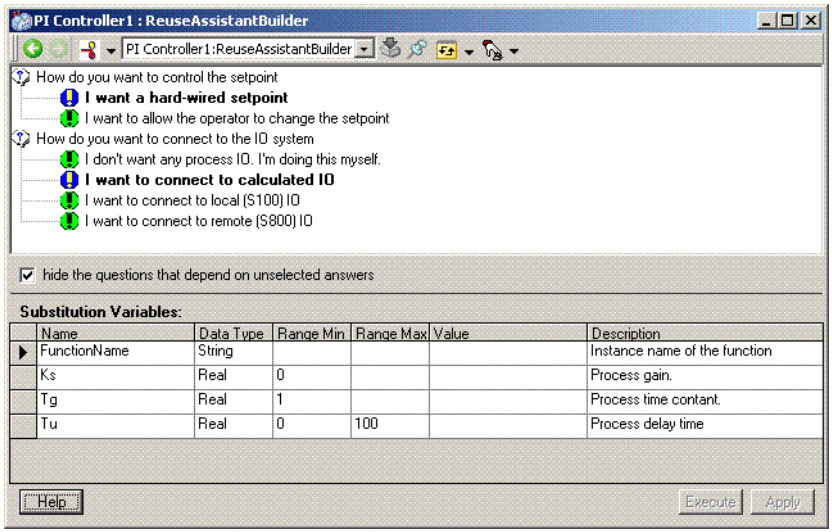


Figure 263. Tree View

The questions and answers of a Reuse Instruction are organized as a hierarchy as described in Build Mode. Answer a question in the Tree view by right clicking the desired answer and clicking the “Choose” command from the context menu.

Right-click a question or answer and click the “Help” command from the context menu in order to display the help that is specific to that question or answer.

The Tree view provides a checkbox labeled “hide the questions that depend on unselected answers”. If this box is checked, then the display shows only pertinent questions. Choosing an answer when this box is checked will reshape the tree if the new choice of answer changed the set of pertinent questions.

If the box is unchecked, then all questions are displayed. In this case it is possible to answer questions that do not pertain, but those answers have no impact on the result of executing the Instruction.

A table below the tree shows the Substitution Variables that pertain to the answer that is in focus. The value column of this table can be edited.

Working with the Substitution Variable Overview

The Substitution Variable Overview presents a table of the pertinent Substitution Variables of the Reuse Instruction, which is shown in Figure 264.

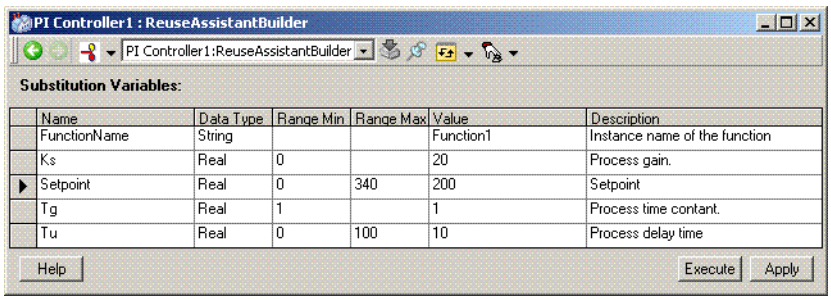


Figure 264. Substitution Variable Overview

The pertinent Substitution Variables are the ones that are used in the pre- or post-operations or in the chosen answers.

The “Value” column of the Substitution Variable table can be edited. It is the only one that can be edited. Whenever the focus leaves a Value entry that has been changed, the value will be validated against its data type and range restrictions and will continue to have its unedited value if the validity check fails. In this event, a pop-up window will announce the error.

For each of the following Data Types of a Substitution Variable there exists a pick up dialog, an easy way to get a value in a valid Control Builder M representation. Clicking on the “Value” field starts the pick up dialog. The selected value from this dialog is taken over into the value field of the substitution variable.

- Time

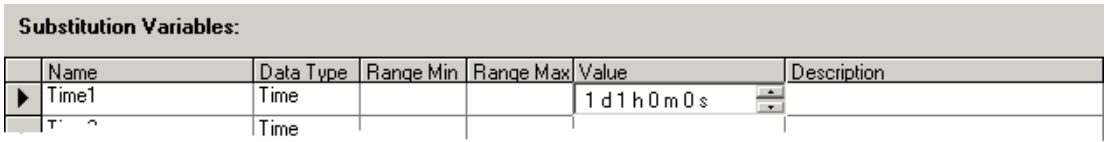


Figure 265. Pick Up Dialog for Data Type "Time"

Select the day, hour, minute and second position and use the arrow up and down keys to set the value.

- DateAndTime

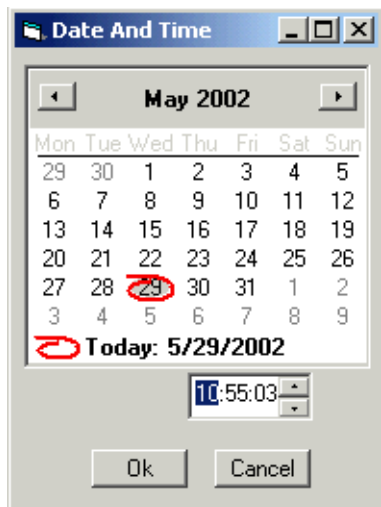


Figure 266. Pick Up Dialog for Data Type "DateAndTime"

The "Ok" command button stores the current choices of Date and Time in the value field.

- AspectPath

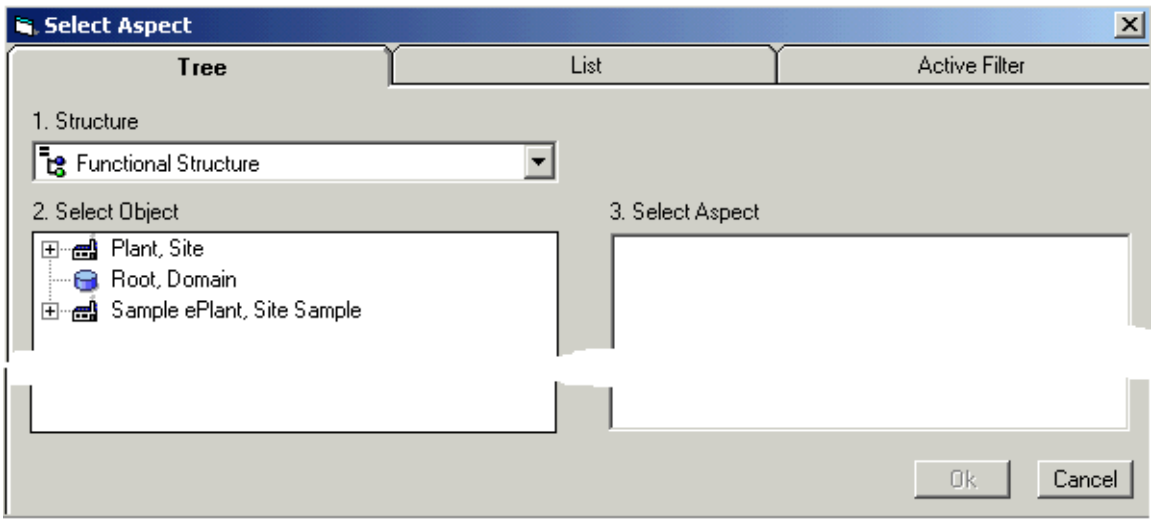


Figure 267. Pick Up Dialog for Data Type "AspectPath"

The "Select Aspect" pick dialog presents three tabs. The tree and the list presentation support the user to search for an aspect.

Tree presentation:

Select a structure, select an object in tree and all aspects of the selected object are listed in the "Select Aspect" view.

List presentation:

Select a structure and choose an object name or object type name (pattern with wild cards are permitted). The object list is filled by pressing the "Search" command button. Select an object out of the list and all aspects of the selected object are shown in the "Select Aspect" view.

The Active Filter tab lists all predefined (in Reuse Architect) Object Types and Aspect Categories. The “Ok” command button is enabled, if the selected object and aspect matches the predefined filter conditions.

In case that no filter conditions are predefined in architect mode, the active Filter tab is not offered.

Pressing the “Ok” command button stores the current choice of an aspect in the value field.

- ObjectPath

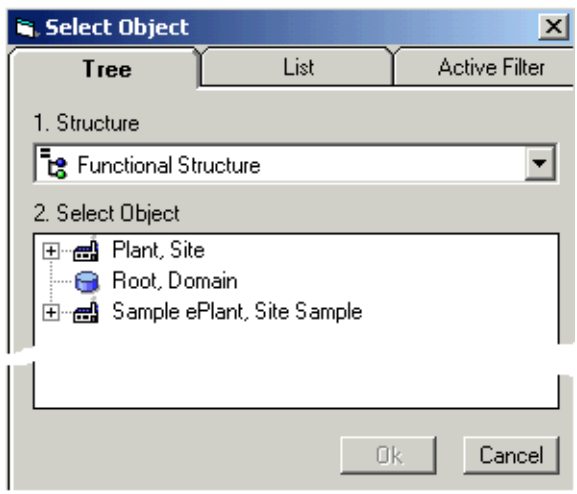


Figure 268. Pick Up Dialog for Data Type “Object Path”

The “Select Object” pick dialog presents three tabs. The tree and the list presentation supports the user to search for an aspect object. The “Active Filter” tab shows all predefined (in Reuse Architect) Object Types. No one of these types can be edited in Build Mode. The type of the chosen aspect object must be an element of this list. Only in this case “Ok” command button is enabled. The functionality of tree and list presentation is similar to the “Select Aspect” dialog, described above.

- ValuePath

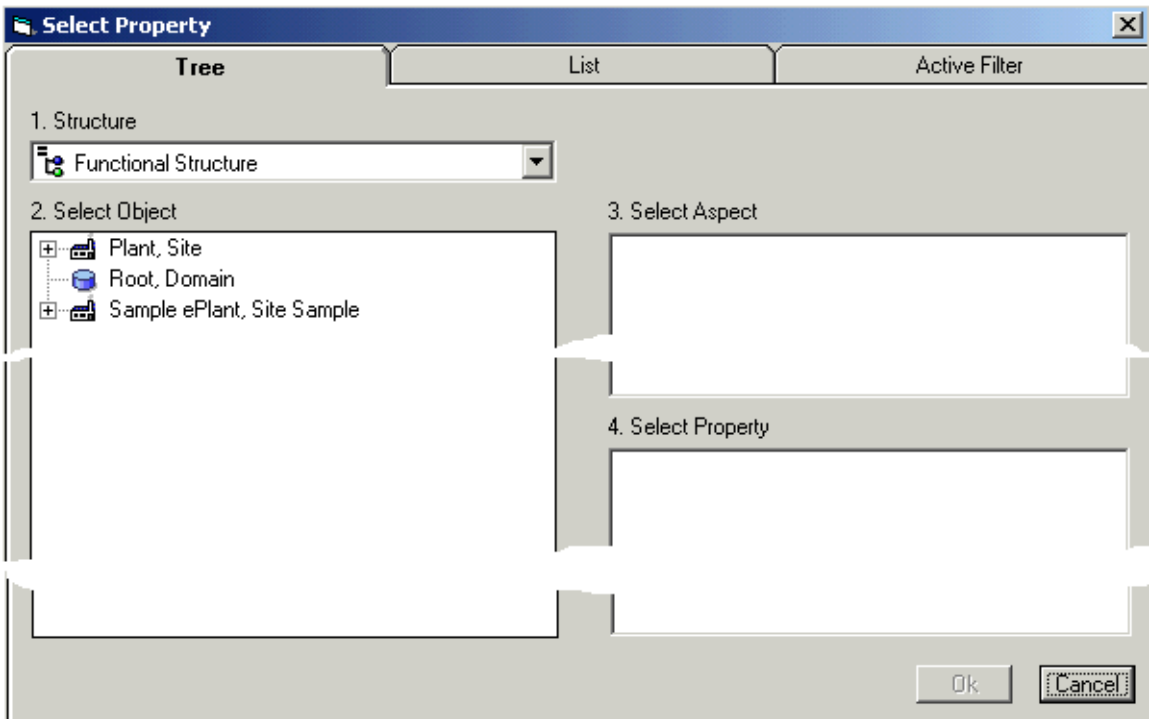


Figure 269. Pick Up Dialog for Data Type "ValuePath"

The "Select Property" pick dialog is similar to the "Select Object" dialog. The "Active Filter" tab shows the predefined (in Reuse Architect) Object Types and Aspect Categories. The type of the chosen aspect object must be an element of the predefined object types. The chosen aspect category must be an element of the predefined aspect categories. The "Select Property" list shows all properties of the chosen aspect. Select a Property and the "Ok" command button is enabled. Pressing this button stores the current choice of a property in the value field.

If a value of Substitution Variable has been changed or an answer has been chosen but not applied, then the "Apply" command button is enabled. Clicking the "Apply" stores the Substitution Variable values and the choices of answers so that they

persist even if the Reuse Assistant Builder Aspect is closed or the Workplace is closed.

If all of the pertinent Substitution Variables have values and there is a complete path of chosen answers, then the “Execute” command button is enabled.

Working with the Option View

The Option View is intended to be used only by people who are primarily using the Reuse Assistant Architect Mode. Most users of the Build Mode will keep the “Perform Operations” option.

The user must be in the “System Engineers”, “Application Engineers”, or “Software Developers” User Group to be able to change this option, otherwise the selection is grayed out and disabled. The organization of users into groups is shown in the “User Structure” of the Plant Explorer.

The “Create Scripting Engine Aspect” option changes the effect of the “Execute” command of the other views. When this option has been chosen, the “Execute” command does not complete the execution of the Reuse Instruction, but rather it creates the Scripting Engine Aspect which the user can later process and watch the operations of the Reuse Instruction be performed as the Reuse Instruction’s script is run.

The Option View is shown in Figure 270.

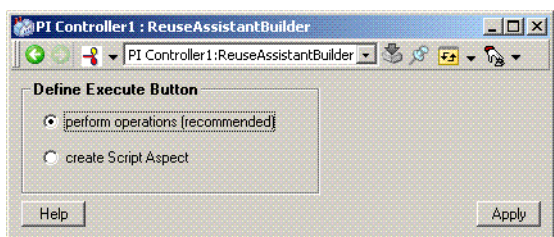


Figure 270. Option View

Builder - Operating Reuse Instructions

How to Execute Reuse Instructions

The objective of the Reuse Instruction in Build Mode is to execute so that it processes the operations that were specified during the Architect Mode. With these operations, it creates objects and parameterizes them according to the answers and values of Substitution Variables that were entered in the Build Mode.

To execute the Reuse Instruction, click the “Execute” command button of the ReuseAssistantBuilder Aspect window. The “Execute” button is enabled only if there is a complete path of chosen answers and the pertinent Substitution variables all have values.

The Reuse Instruction executes by first automatically running a consistency check. Consistency checking is described in Consistency Check. If there is a consistency failure then it will be displayed as shown in that chapter.

If the consistency check passes, the Reuse Instruction execution then processes its script according to the choices of answers and the values of Substitution Variables that were provided in the Build Mode. If the execution does not entirely succeed, then all of its operations are rolled back and none of the changes is made.

If the execution succeeds a pop-up message appears.

The results of the execution should correspond to documentation that was provided with the specific Reuse Instruction.

When the Reuse Instruction performs the operations the answer selections and values of substitution variables are logged in the System Messages Aspect System. It is viewed by clicking to put into focus the Alarm and Event List Aspect, which is another Aspect of the Reuse Instruction along with the ReuseAssistantBuilder. The

log appears as in the example shown in Figure 287.

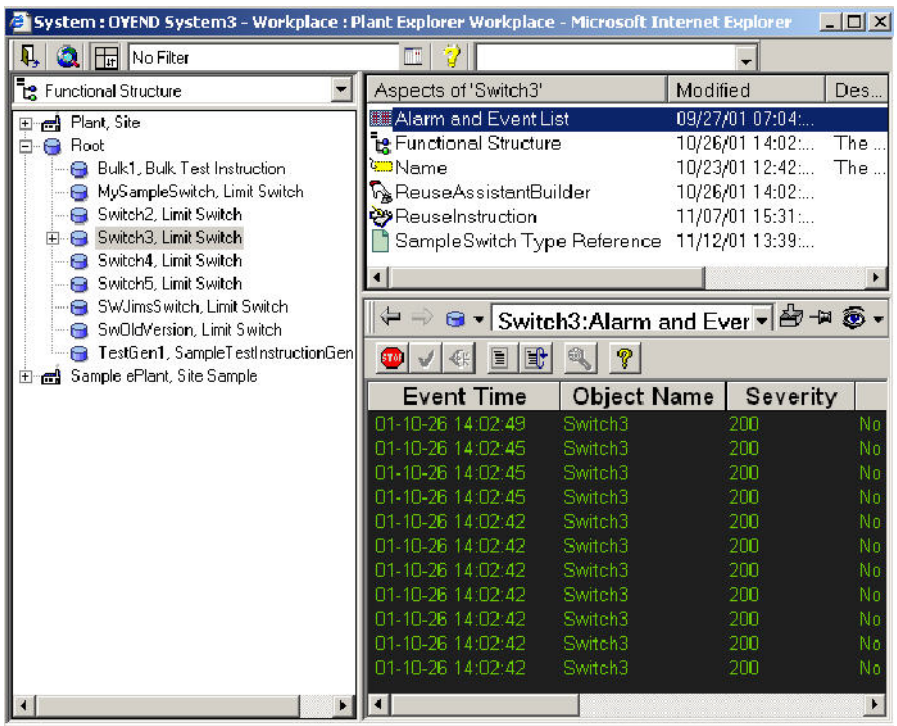


Figure 271. System Message Aspect

If the executed operations return results about the success of these operations they are also logged into this Aspect.

How to work with Reuse Instructions

Consistency Check

The consistency check determines whether all of the Object Types, Aspect Categories, and System Extensions that are called upon by the operations of the Reuse Instruction are available in the system.

Request a consistency check by right clicking on the ReuseAssistantBuilder Aspect and clicking on the “Check Consistency” context menu command as shown in Figure 272.

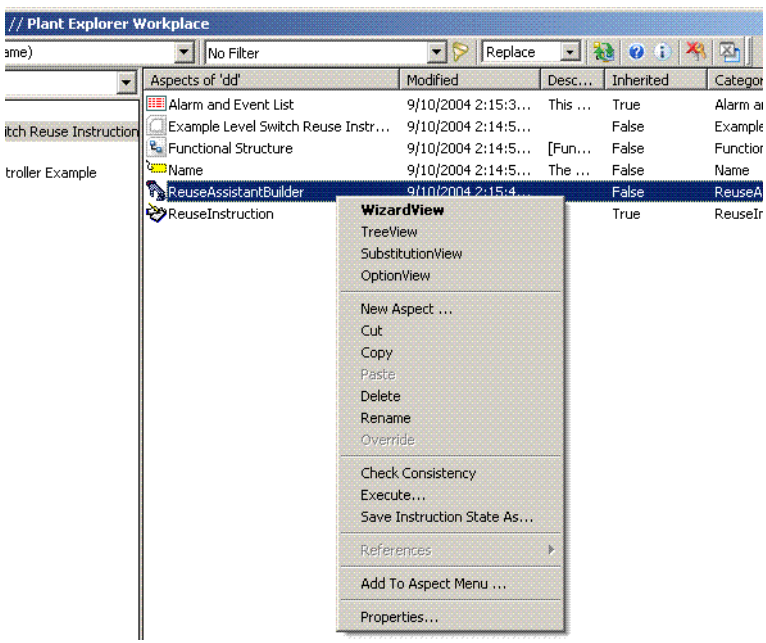


Figure 272. Requesting a Consistency Check

The check is performed over the objects that are directly referenced by the Reuse Instructions. If these objects are available then consistency is guaranteed to all levels because of the platform’s other consistency maintenance.

Failures of the consistency check are listed as shown in the example in Figure 289.

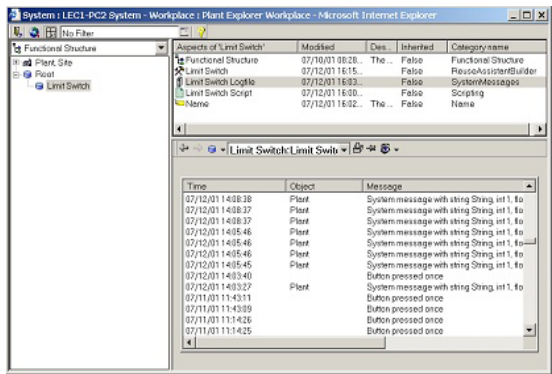


Figure 273. Consistency Check Results

Cut, Copy, and Paste of a Reuse Instruction in the Build Mode

The Aspects of the Reuse Instruction can be cut, copied, and pasted.

It is vital however that the Aspects that are involved in the Reuse Instruction keep the same relationship. The ReuseAssistantBuilder, the Reuse Instruction and the System Message Aspect must be kept together. All three must be copied and pasted together. If one is cut the other two should also be cut. It is recommended that cutting and copying be done to the Aspect Object that contains these three rather than to any of the three individually.

When consistency checking is done it will compare the identities of the three Aspects. There may be additional constraints that the specialized Reuse Instruction bears.

Once the Reuse Instruction has been copied and pasted it is an additional Reuse Instruction. As Answers are chosen and Substitution Variables are given values the other copy is not changed by these actions.

Instruction Comparison

Reuse Instructions can be output in the form of XML text. Any application with the capability to compare text can then be used to show the differences.

To export the Reuse Instructions into a file in XML format right click on the ReuseAssistantBuilder to see the context menu and click on the “Save Instruction State as...” command as shown in Figure 290. The standard file selection dialog pops up to request the filename.

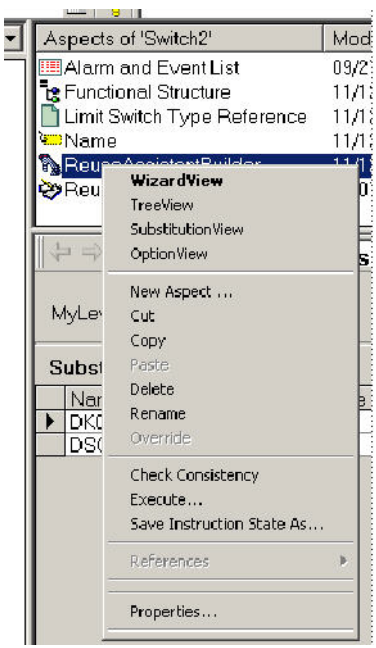


Figure 274. ReuseAssistantBuilder Aspect Context Menu

The ability to read this XML text is not required by the user of the Build Mode for any other purpose. It may be difficult to interpret because it combines the knowledge from the Architect Mode with the arcane XML. Nevertheless the changes may involve names of objects that are indeed recognized by the user of the Build Mode.

Exporting a Reuse Instruction from the Build Mode

A Reuse Instruction can be exported with the System 800xA Import / Export application. Answers can be chosen and Substitution Variables given values before exporting. That way the exported Reuse Instruction is specialized by the user of the Build Mode to meet a more narrow circumstance.

It is important to include in the export all of the objects that need to work together.

Debugging a Script

Debugging a Script is a capability that is intended for the user of the architect although it is used in Build Mode.

The Scripting Engine provides the user with the capability to debug the script. The Scripting Engine Aspect is only created if the Scripting Engine is installed and the user chooses this function in the Options of the Reuse Assistant builder. Selecting this option is described in Working with the Option View.

The “Create Scripting Engine Aspect” option changes the effect of the “Execute” command. When this option has been chosen, the “Execute” command does not execute the Reuse Instruction. Instead a Reuse Assistant dialog pops up, asking “Do you really want to override or create the Scripting Aspect?”. Answering Okay then creates or updates the Scripting Engine Aspect. You will get a display similar to the one when you select this Aspect.

Now you are able to run this script with debugging support like setting of breakpoints and watching of variables. An advanced user can perform the script step-by-step or make intermediate changes of variables. The full functionality of the ABB Scripting Engine is described in Section 9, Script Manager, a display of the Scripting Engine is shown in Figure 291.

Script Manager requires a breakpoint in the first line of the script for proper work with arbitrary other breakpoints.

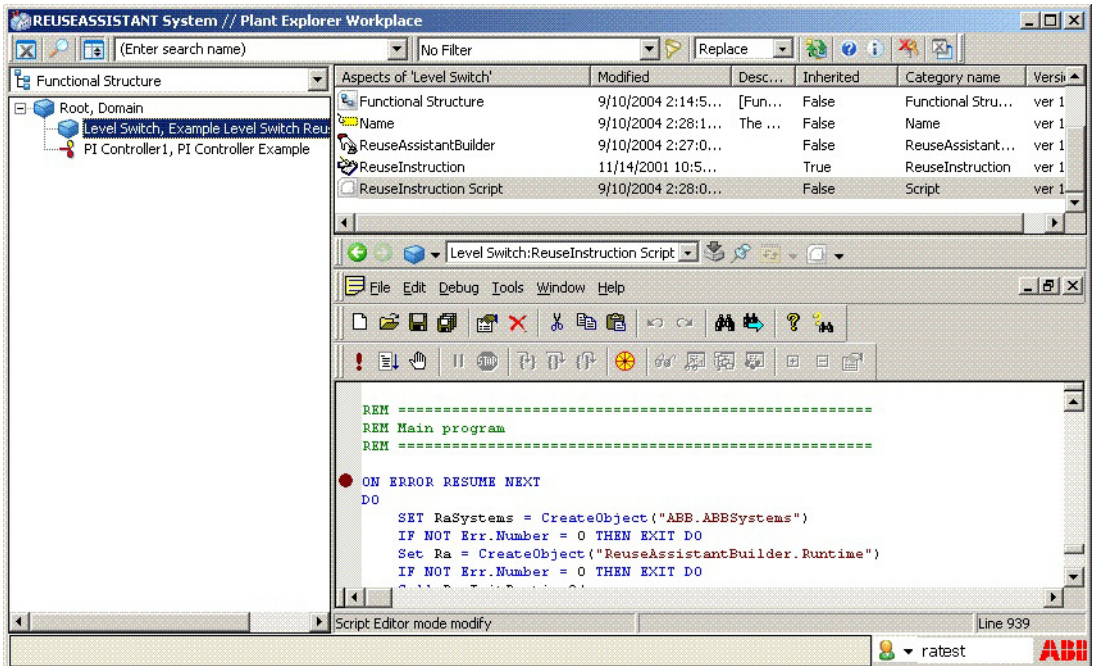


Figure 275. Scripting Engine Display

Executing the Reuse Instruction More then Once

A warning pop-up is presented it the user attempts to execute a Reuse Instruction.

Re-executing the Reuse Instruction does not undo what was previously done in the earlier execution. It may not overwrite everything and it may not write a separate copy of everything. It is dependent upon the specific script and the set of objects in the system when the instruction is executed.

Copying and pasting may be a better means to achieve what the user may intend.

Reuse Assistant - System Functions

This subsection informs about the manifestation of some System Functions within the Reuse Assistant.

Audit Trail

Specific Audit-Trail messages are written at

- (Re-) Generation of a Reuse Instruction
- Begin and end of executing a Reuse Instruction

Unspecific Audit-Trail messages are written when one of the ‘Reuse’ Aspects is modified, as well as when Objects or Aspects are created, modified or deleted.

User Log-Over

You can use the Log-Over function within Reuse Architect and within Reuse Builder. After log-over the current aspect window is closed, the next window you open is then shown according to the new user rights.

Version Handling

Dependencies between existing Objects / Aspects and Reuse Instructions are handled by the Import / Export utility.

Reuse Assistant - Error Messages

The following table lists all messages that may be written into the system message log when a Reuse Instruction is executed.Messages with the numbers 100..499 are error message. Messages with the numbers 500..1000 are information messages.

Table 33. Error Messages

No.	Location	Error Description
100	RaCreateAspect	The function failed while trying to create an aspect of a certain category for a given object.
101	RaModifyProperty	Accessing a given aspect's property is not possible.
102	RaModifyProperty	Changing the value of a given aspect's property is not possible.
103	RaFindChild	he object addressed by a structure cursor has more than one child of the desired name.
104	RaFindChild	The object addressed by a structure cursor has no child of the desired name.
105	RaInsertChildObject	Inserting an object at the root level of the given structure failed.
106	RaInsertChildObject	Inserting an object into the given structure failed.
107	RaOverrideAspect	Overriding a given aspect of a given object failed.
108	RaSetSuperType	Setting the supertype of a given object in this object's type definition failed.
109	RaSetAspect CategoryControl	Failed to define the aspect category create rule (category and min./max values) in the given object's type definition.
110	RaSetAspect CategoryControl	Failed to set the "Propose When Creating" flag of an aspect category create rule in the given object's type definition.

Table 33. Error Messages (Continued)

No.	Location	Error Description
111	RaSetAspect CategoryControl	Failed to set the “Create When Object Is Created” flag of an aspect category create rule in the given object’s type definition.
112	RaChildControl	Failed to modify the child control rule for a given object’s type definition.
113	RaAspectControl	Failed to modify the aspect control flags for a given object’s type definition.
114	general	The call for finding an object did not return exactly one object.
115	general	The structure cursor does not point to a valid location.
116	general	The call of the sub for creating a new generic root object failed.
117	general	The call of the sub for creating a new root object failed.
118	general	The call of the sub for creating a new generic child object failed.
119	general	The call of the sub for creating a new child object failed.
120	general	Object is invalid.
121	general	to call of the sub for inserting a given object as a child of an object given by a structure cursor failed.
122	general	The call of the sub for overriding an aspect failed.
123	general	The call of the sub for supertyping an object failed.
124	general	The call of the sub for setting the aspect category control of an object failed.
125	general	The call of the sub for setting the child control of an object failed.
126	general	The call of the sub for setting the aspect control of an object failed.
127	general	The call of the sub for creating an aspect failed.
128	general	Setting a structure cursor to the descendent of a given structure cursor failed.
129	general	The call of the sub for modifying a property failed.

Table 33. Error Messages (Continued)

No.	Location	Error Description
500	RaCreateAspect	A new aspect has been created for the given object.
501	RaModifyProperty	A property of an aspect has been successfully modified.
502	RaFindChild	A child of a certain name has been found in the given structure.
503	RaInsertChildObject	An object has been placed successfully into a given structure as root object
504	RaInsertChildObject	An object has been placed successfully into a given structure.
505	RaOverrideAspect	An aspect has been successfully overridden.
506	RaSetSuperType	The supertype of a given object in this object's type definition has been successfully set to a new value.
507	RaSetAspect CategoryControl	The aspect category control has been successfully modified.
508	RaChildControl	The child control for a given object has been successfully modified.
509	RaAspectControl	Successfully modified the aspect control flags for a given object's type definition.
510	RaFindChild	Child with the desired name has been found at the given object location.
511	general	The call for creating a new generic root object seceded.
512	general	The call for creating a new root object seceded.
513	general	The call for creating a new generic child object seceded.
514	general	The call for creating a new child object seceded.

Section 9 Script Manager

Script Manager is a System Extension of the Engineering Workplace.

Script Manager is available in two variants Script Manager Basic and Script Manager Professional. Script Manager Basic allows to run scripts developed with Script Manager Professional. Script Manager Basic is contained in the Engineering Workplace, Script Manager Professional is optionally available.

Script Manager Basic Functions

To use Script Manager Basic Functions you have to install the System Extension **Script Manager** before.

Script Manager Basic adds scripting capability to Aspect Objects using VBScript Language. Scripts programmed in VBScript are used to:

- Perform plausibility checks and automate simple planning tasks.
- Provide extended rules and actions to Aspect Objects.
- Implement dependencies between objects and structures.
- Perform extended logging and tracing.
- Implement user and application specific dialogs and menus.
- Perform calculations and arithmetic functions.
- Interface external tools that support VBScript and Windows Scripting Host.
- With Script Manager Basic you can use the entire scope of the Automation Interface of the System 800xA platform.
- Scripts are started either manually or automatically based on structural- or aspect parameter changes. Automatically started scripts can commit or cancel

modifications. Therefore the script of an Aspect Object may decide, whether to allow or reject a modification.

- You can receive OLE-Events in a script. That enables you to build an interactive UI using Visual Basic. For example: We have an Active-X dialog, which fires an OLE-Event “button_ok_pressed“. You can write a script, which receives this event and performs a reaction on it.
- The trigger options allow a more extended definition of the start conditions for scripts.
  - It is possible to trigger Scripts independent from a structure.
  - The Aspect Change Trigger needs no dependence on a special Attribute. You can define the trigger conditions without specifying a dependent OPC attribute.
  - An Object Instantiation Trigger is implemented, which enables you to run a Script automatically if an Object Type is instantiated.
- With **Menu Verbs** you are able to integrate Script procedures in the context menu from an Aspect Object or your Script Aspect. The context menu executes single Script functions.
- **Data Persistence and OPC access** for persistent values support saving persistent values in the script. Other Aspects can access those data via the System 800xA platform property access mechanisms.
- **Additional Script functions** to make your Script development easier

Script Manager Professional Functions

Optional Script Manager Professional allows to develop and debug scripts.

The **Integrated Debugger** allows you to compose and debug Scripts in the same environment.

System Creation

Create a new system.

When creating a new system, a **System Extension** dialog is available in the Configuration Wizard where you can check the system extensions or components which you want included in your system. To get the **Script Manager** functions in your system, click the check box ***Script Manager***

Settings

There are two ways to invoke the Script Manager scripting options:

- Start “Script Trace Window”.
  Select the **File > Global Script Manager Options** command.
- Select a Script Aspect within the Plant Explorer. The “Script Editor” opens.
  Select the **Tools > Global Script Manager Options** command.

800xA settings consist of two parts (see Figure 276):

- Triggering Options (“Trigger“).
  - **Enable automatic script** - enable / disable automatic script execution on structural or Aspect property modifications.
- Logging Options. (“Logging“)
  - **No Logging** - activities are not logged.

Getting Started

Script Manager is composed of the following software parts:

- Script Editor supplies the user interface to write, test and run Visual Basic scripts, change trigger conditions for each script and change global 800xA properties and settings. It also supplies an integrated debugger environment to test and debug Visual Basic scripts.
- Trace Window visualizes outputs and tracing prints of scripts and includes a user interface for printing.

Script Editor Overview

The Script Editor is an aspect system window, integrated in Engineering Workplace. It can either run in the preview area (see Figure 276) or in a separate overlap window (see Figure 277).

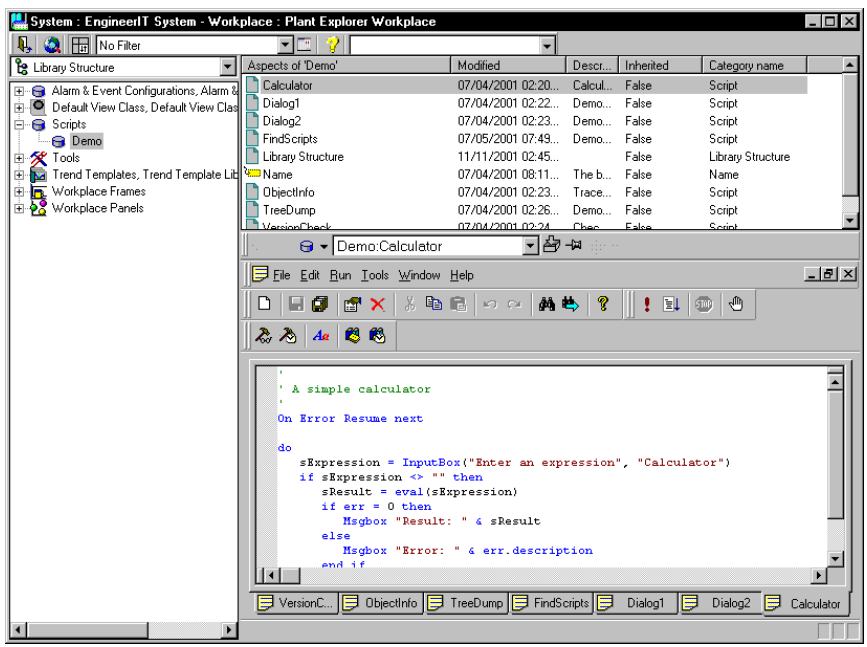


Figure 276. Script Editor in Preview Area

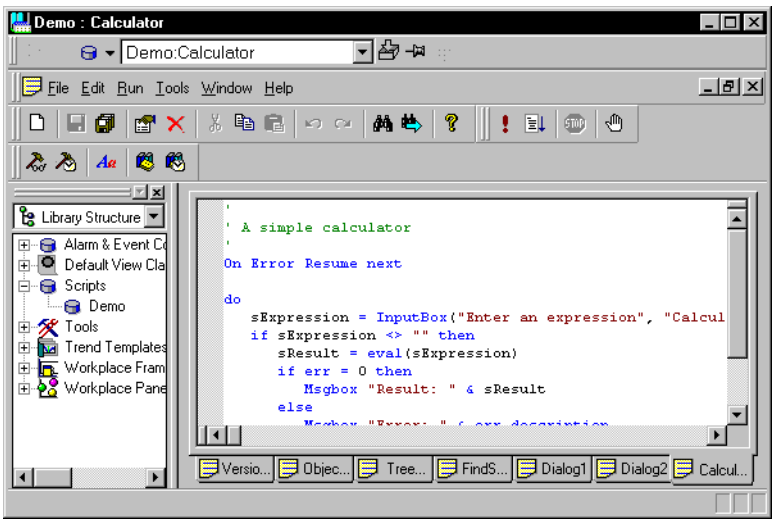


Figure 277. Script Editor in Overlap Window

The Script Editor supports syntax coloring: Basic keywords, strings and comments are shown in different colors.

Script Editor is used for both composing and debugging scripts. Also all script settings are accessible from the Script Editor

The following commands are accessible in the Script Editor via the menu:

Figure 278. Script Editor Menu

File menu

- **New Script**
  Creates a new script.
- **Open**
  Opens an existing script.

- **Rename**
  Renames the current script.
- **Delete**
  Deletes the current script.
- **Save**
  Saves the current script.
- **Save Al**l
  Saves all opened scripts.

Edit menu

- **Undo**
  Reverses the last command, or delete the last characters, you typed.
- **Redo**
  Reverses the last Undo command.
- **Cut**
  Removes the selection from the active document (script text) and place it on the clipboard.
- **Copy**
  Copies the selection from active document (script text) to the clipboard.
- **Paste**
  Inserts the contents of the clipboard at the insertion point and replace any selection. This command is available only if you have cut or copied any text.
- **Select All**
  Selects the entire text in the active window.
- **Bookmark**
  Toggles a bookmark for the current line on and off.
- **Next Bookmark**
  Moves to the line containing the next bookmark.
- **Previous Bookmark**
  Moves to the line containing the previous bookmark.

- **Delete All Bookmarks**
  Clears all bookmarks in the window.
- **Increase Indent**
  Indent the selected text right one tab stop.
- **Decrease Indent**
  Outdent the selected text left one tab stop.
- **Find**
  Finds the specified text.
- **Find Next**
  Finds the next occurrence of the specified text.
- **Replace**
  Replaces a specified text by a new one.

Debug menu

- Attach
  Invokes a dialog, which shows all running scripts. You can connect the debugger to a running script or terminate it.
- **Run Script**
  Runs the currently active script without debugger. Breakpoints are disabled.
- **Run Debug**
  Starts a debugger session for the currently active script.
- **Break**
  Breaks the currently active script running in debug mode.
  The current script statement must be finished, before the break hits. Therefore the break might not occur immediately, in especially when breaking in a MsgBox statement, you have to close the related message box first. The execution of a script is suspended. You can resume it again in the debugger.
- **Stop**
  Stops the currently running script.
  The current script statement must be finished, before the script is stopped. Therefore the script execution might not stop immediately, in especially when

stopping in a MsgBox statement, you have to close the relates message box first.

- **Step into**
  Available in debug mode only.
  Steps into the next statement.
- Step over
  Available in debug mode only.
  Steps over the next statement.
- Step out
  Available in debug mode only.
  Steps out of the current statement.
- **Toggle breakpoint**
  Toggles a Breakpoint at the current line on and off.
- Remove all breakpoints
  Clear all breakpoints in script.

Debug > Windows

- Quick Watch
  Available in debug mode only.
  Shows Quick Watch dialog window. The Quick Watch dialog provides an interface to evaluate an expression.
- **Watch Window**
  Available in debug mode only.
  Toggles the watch window on and off.
- **Variables Window**
  Available in debug mode only.
  Toggles the variables window on and off.
- **Status Messages**
  Available in debug and edit mode.
  Toggles the status window on and off.

- Callstack
  Available in debug mode only.
  Toggles the callstack window on and off.
- **Expand Variables**
  Available in debug mode only.
  Expands the object level in the variables window.
- **Collapse Variables**
  Available in debug mode only.
  Collapse the object level in the variables window.

Tools menu

- **Script Options**
  Invokes the **Script Options Dialog** with the **Trigger Conditions** and the **Menu Verbs** section.
  The **Trigger Conditions** dialog displays and changes the trigger conditions of the current script.
  The **Menu Verbs** dialog associates script functions to Object or Aspect context menus.
- **Global Script Manager Options**
  Invokes the Scripting Options dialog. The Scripting Options dialog displays and changes the scripting options for **800xA**. For a short description of these options refer to Settings**.**
- **Set Font**
  Invokes the **Font and Color Settings** dialog. The Font and Color Settings dialog displays and changes the font properties and syntax highlighting in Script Editor.
- Customize Toolbars
  Invokes the Customize dialog. The Customize Dialog configures the toolbars displayed and the command buttons on the toolbars.
- **Type Libraries Browser**
  Invokes the **Type Libraries Browser** window. The Type Library Browser displays the content of the connected Type Libraries. A Type Library can be connected or disconnected by use of the Type Libraries Browser

- **Type Libraries Browser Options**
  Invokes the **Type Libraries Browser Options** dialog. The Type Libraries Browser Options dialog provides an interface to connect or disconnect various type libraries, installed on your system.

Window menu

- **Workbook Mode**
  Toggles the Workbook Mode on and off.
- **New Window**
  Opens another window for the current script.
- **Cascade**
  Arranges the windows as overlapping tiles.
- **Tile Horizontally**
  Arranges the windows as horizontal, non-overlapping tiles.
- **Tile Vertically**
  Arranges the windows as vertical, non-overlapping tiles.
- **Arrange Icons**
  Arranges the icons of all minimized windows as horizontal, non-overlapping tiles.

Help menu

- Contents
  Displays the contents of the online help.
- Search
  Invokes the search for a specific help topic.
- **Index**
  Displays the online help index.
- **About** Script Editor
  Displays the program information and copyright.

Script Editor Toolbars

- **Standard**
  Includes all commands from the Script Editor **File** and parts from the **Edit** menu.
- **Bookmarks**
  Includes commands from the Script Editor Edit menu.
- **Debug**
  Includes commands from the Script Editor Debug menu.
- **Window**
  Includes all commands from the Script Editor Window menu.
- **Tools**
  Includes all commands from the Script Editor Tools menu.

Script Editor Toolbar Customizing Menu

To invoke the Toolbars customizing menu, right-click the mouse on the grey area of any toolbar or menu.

Figure 279. Toolbar Customizing

- **Structure**
  Toggle "**Structure Window** on and off.
- **Standard**
  Toggle **Standard Toolbar** on and off
- **Bookmarks**
  Toggle **Bookmarks Toolbar** on and off

- **Debug**
  Toggle Debug Toolbar on and off
- **Window**
  Toggle Window Toolbar on and off
- **Tools**
  Toggle Tools Toolbar on and off.
- **Debug Status Window**
  Toggle the Status Window on and off.

Script Editor Toolbar Customizing Dialog

To invoke the Toolbar Customizing Dialog, select **Tools > Customize Toolbars** from the Script Editor menu. The **Customize** Dialog with two property pages appears. On the **Toolbars** page, you define the toolbars to be displayed.

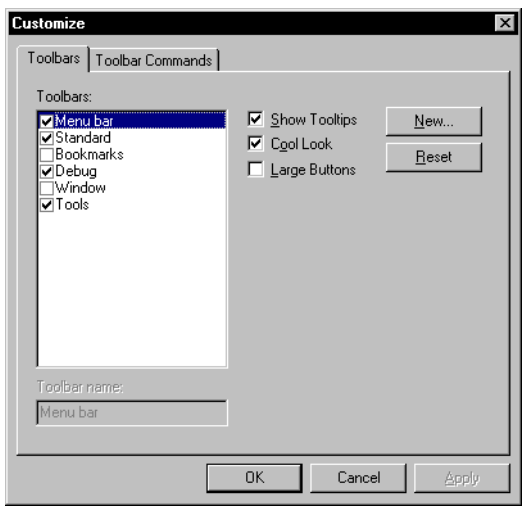


Figure 280. Customize Toolbars Property Dialog

Using the **Toolbar Commands** page, you can customize individual commands on your toolbars using drag and drop.

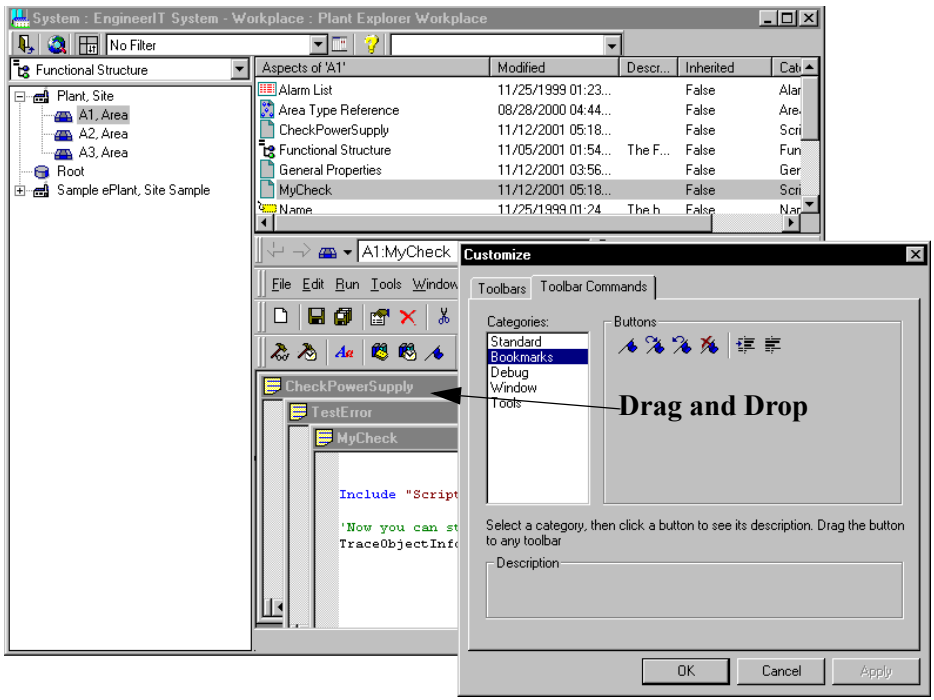


Figure 281. Customize Toolbar Commands

Trace Window

The Trace Window is an MDI style standalone application, started when the first trace function call is executed. It displays the trace messages.

This program is not integrated into the Engineering Workplace and its main window can be freely resized and positioned at the screen. For each script a child window is created to show its trace messages. The scripts name and its status (running or terminated) are shown in the title bar. An extra pane maintains a table of the most recent executed scripts with additional information:

- Run state (running or terminated).

- Script and object names.
- Type of trigger and the trigger condition, that initiated the script.
- Execution time (start and duration).
- Error codes.
- Additional information provided by the script engine.

The user can close a scripts output window only when it is terminated. The program cannot be terminated while a script is running in debug mode.

Standard functionality of an MDI user interface like tiling and cascading windows is provided. Additionally a user can

- Copy portions of text from a child window.
- Save or print the contents of window.
- Set the maximum number of child windows displayed at a time.
- Select a trace output by double-clicking a row in the list window.

- Sort the list window by any column.

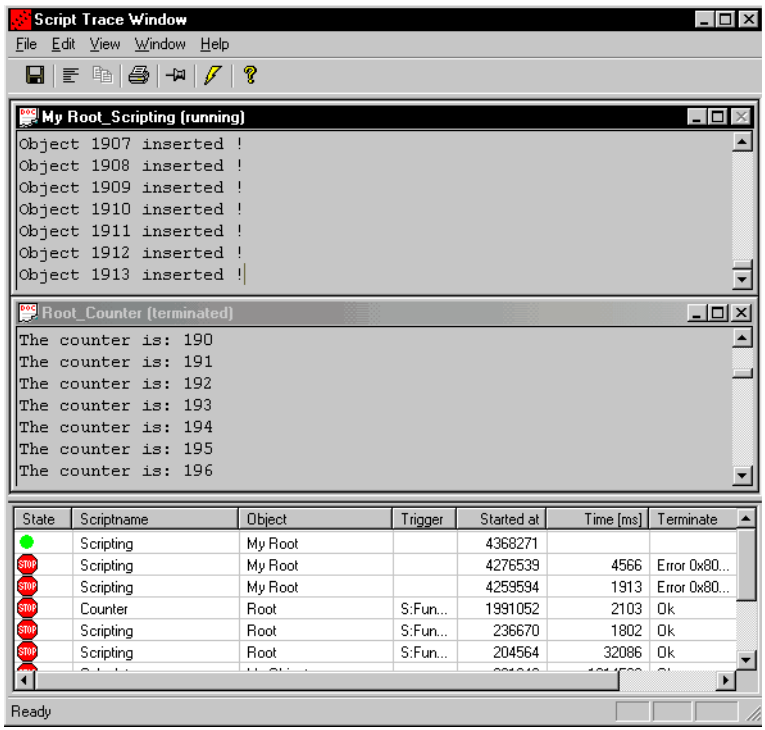


Figure 282. Trace Window Application

The following commands are accessible in the Trace Window via the menu:

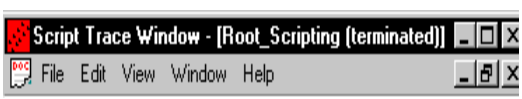


Figure 283. Trace Window Menu

File menu

- **Close**
  Close the current window.

- **Save As**
  Save current window's content in text file.
- **Print**
  Print current window's content.
- **Print Preview**
  Shows how a text will look when you print it.
- **Print Setup**
  Shows a system Print Setup dialog window. This dialog provides an interface to change current printer, printer properties and paper layout.
- **Options**
  Shows "Options" dialog window. This dialog provide an interface to change the parameters of Trace Window application, such as: Maximum number of log windows etc.
- **Scripting Options**
  Shows the **Scripting Options** dialog window. The Scripting Options dialog window provides the interface to set and change the Global Scripting Options for **800xA**. For a short description of these functions please refer to Settings.

Edit menu

- **Select All**
  Selects all text in the script.View
- **Copy**
  Copies the selection from active document (script text) to the clipboard.

Window menu

- **Cascade**
  Arranges the windows as overlapping tiles.
- **Tile**
  Arranges the windows as horizontal, non-overlapping tiles.
- **Arrange Icons**
  Arranges the icons of all minimized windows as horizontal, non-overlapping tiles.

- **Close All**
  Close all script trace windows.

Help menu

- **About** Trace Window
  Displays the program information and copyright.

Application Start-up

The **800xA** is accessed via the Engineering Workplace by opening an aspect representing a script.

Opening a script is performed in the Engineering Workplace's *Aspect List Area*.

Working with Script Manager

The Script Editor is the application to compose, test and configure Visual Basic scripts in the System 800xA platform context. You compose Scripts within the Editor. You test and debug your Scripts with the integrated debugger and you configure the Script execution for your operating environment. For example your Operator Workplace.

To run scripts by trigger or as an Aspect or Menu Verb does not require the Script Editor. Therefore you do not need the Script Editor in your operating environment.

Script Editor Settings

To change font settings and syntax highlighting options select **Tools >Set Font** command in “Script Editor“. This command will open the “Font and Color Settings” dialog.

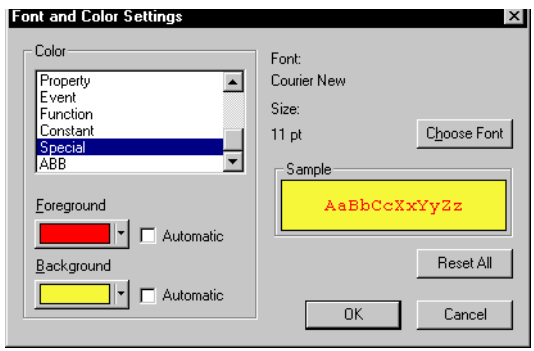


Figure 284. Font and Color Settings Dialog

This dialog enables you to change font properties and syntax highlighting properties (section “**Color**”) for each token group of VBScript.

Create a New Script

A new **“Script Aspect**” is created either in the Script Editor or in the Engineering Workplace:

- To create a new script in the Script Editor, invoke the **File > New Script** command. The new “**Script Aspect**”is created for the Aspect Object, the Script

Editor currently is related to.

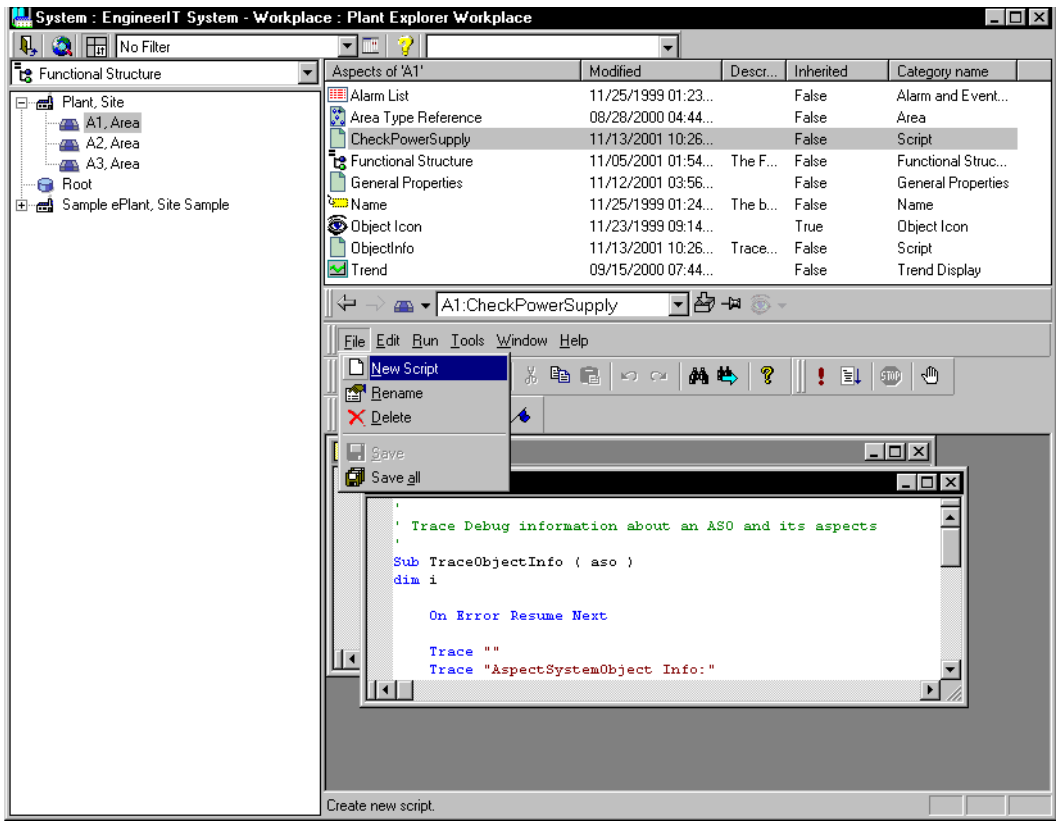


Figure 285. Create a New Script in Script Editor

- The name of the new script is “**Untitled**”. You can rename the script to a more meaningful name using the **File > Rename** command.
- A new “**Script Aspect**” is created in two steps in the Engineering Workplace:
  a. Right click on the Aspect Object in the “**Object Tree**” (see Figure 302) and choose the “**New Aspect**” command from the popup menu.

b. Select the **Script** Aspect from the list of Aspect Types in the "**New Aspect**" dialog. Enter the name for the new script in the field "**Name**".

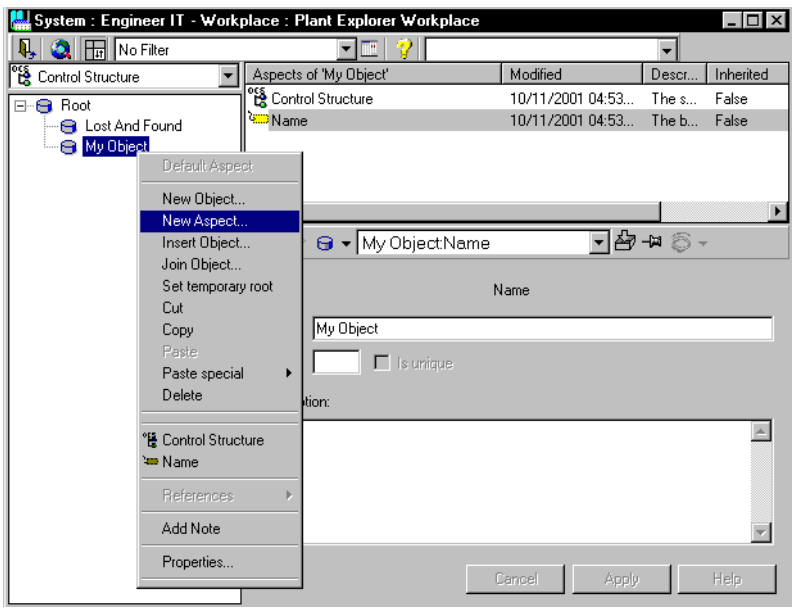


Figure 286. Create a New Script Aspect

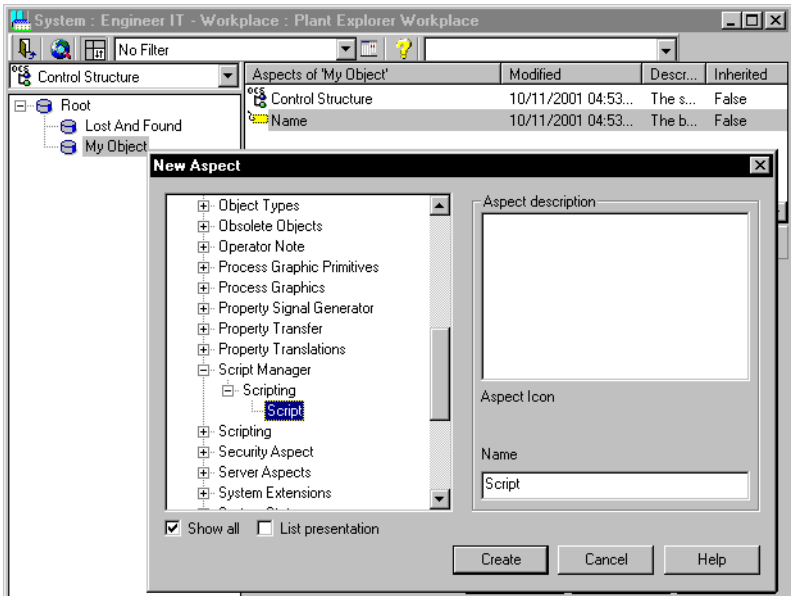


Figure 287. Select the Script Aspect From the List

Rename Script

To rename a script, select the **File >Rename** command in Script Editor. This pops up the "**Rename Script**" dialog window.

Figure 288. cript Rename Dialog

Enter the new script name and click "OK".

Delete Script

To delete a script, select the **File >Delete** command in the Script Editor. A warning message requires you to confirm the deletion of the script.

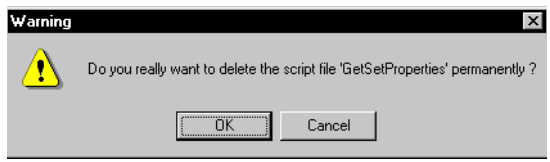


Figure 289. Script Delete Warning

Click “OK” to permanently delete the script.

Open a Script

The **Script Open** dialog opens existing Script Aspects, not only from the current object, but from all Aspect Objects within the entire system.

The navigation is based on the **File Open** dialog known from other Windows applications. To open a Script select **File** > **Open**. In **Look in**, click the Aspect Object that contains the script to open.

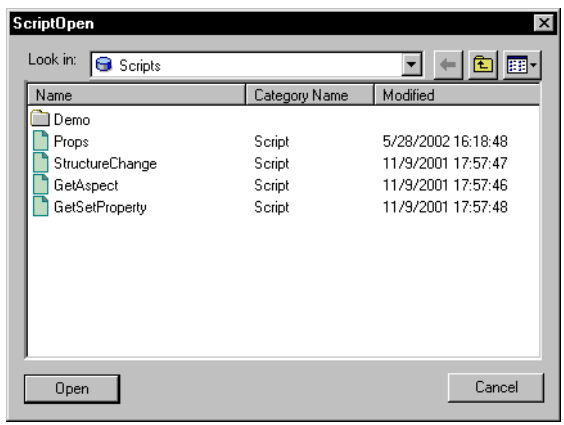


Figure 290. Script Open Dialog

Run Scripts Through Script Editor

You can start execution of a Script in two different modes:

1. **Run Script**
   Select the **Debug > Run Script** command in the Script Editor. The currently selected script starts without debugger.
2. **Debug Script**
   Select the **Debug > Debug Script** command in the Script Editor. The currently selected script starts in with debugging functions enabled. The Script will run until the script ends or a breakpoint hits. To break a running script select **Debug > Break**. You may not modify Scripts during debug sessions, since modifying the code of a running script is not allowed.

If a script runs through Run Script or Debug Script command in Script Editor you can't close the workplace or the Script Window. Therefore wait until the script has finished, or stop it with the Stop command.

- Don't forget to close Message Boxes that belong to your running Script.
- Look if a Message Box is hidden by another window.
- Look for other Dialogs your Script could have activated. Those Dialogs have to be closed.
- In some cases the script execution may be blocked. Close the workplace using the Task Manager.

Breakpoints

To set a breakpoint, select the **Debug > Breakpoint** command in the Script Editor. This command toggles a breakpoint at the current line. The breakpoint is evaluated during the debug session. Breakpoints are indicated as dark red dot.

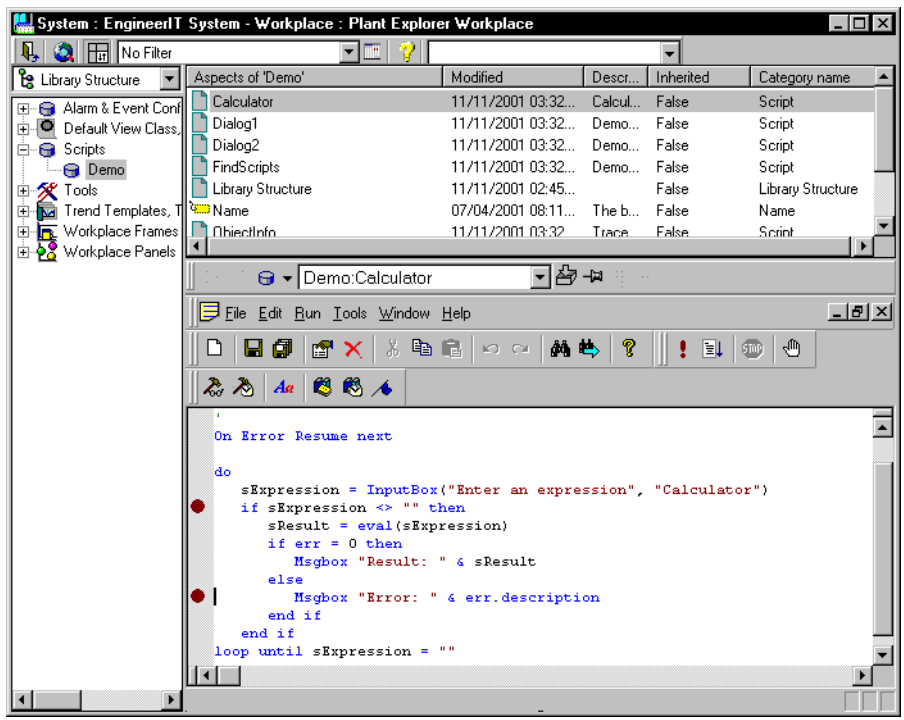


Figure 291. Breakpoints in the Script Editor

Breakpoint set in an empty line in Editor Mode: The breakpoints will not be evaluated at run time.

Breakpoint set in an empty line in Debug Mode: The script breaks at the next executable statement.

Breakpoint set in comment lines: In Editor mode and Debug mode the behavior is the same, at run time the script breaks at the next executable statement.

Using Bookmarks

The Script Editor enables you to set bookmarks in scripts and quickly navigate between these bookmarks.

The bookmarks are indicated as a light blue marker at the left margin of the script text. You can use the following commands to navigate along bookmarks

- **Edit > Bookmark** toggles the bookmark at the current line.
- **Edit > Next Bookmark** sets the text cursor position at the beginning of the next line, marked with a bookmark.
- **Edit > Previous Bookmark** sets text cursor position at the beginning of the previous line, marked with a bookmark.
- **Edit > Delete all Bookmarks** deletes all bookmarks in the current script.

Fast Navigation to Included Scripts.

To navigate to included scripts, click with the right mouse button on the line with the keyword **include** (see also Include Other Scripts). In the popup menu an entry

Open Script with script name appears. This command opens the included script in the Script Editor.

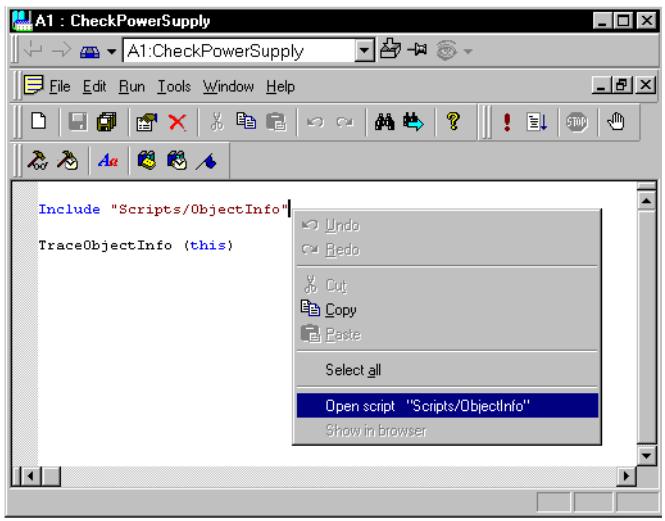


Figure 292. Popup Menu "Open Script"

Using the Type Libraries Browser

The **Type Libraries Browser** supports you using COM objects within a script. Type libraries describe which functions and properties a COM object supports and how to access them. However, they are difficult to read or require external tools (e.g. Visual Basic or a type library Viewer). To ease the usage of COM objects within a script, the Script Editor contains the **Type Libraries Browser**. To work with the **Type Libraries Browser**, you first have to configure, which type libraries are available for browsing.

To announce a type library to your Script Editor, select **Tools > TypeLibraries Browser Options**. The **Connected Type Libraries** dialog pops up.

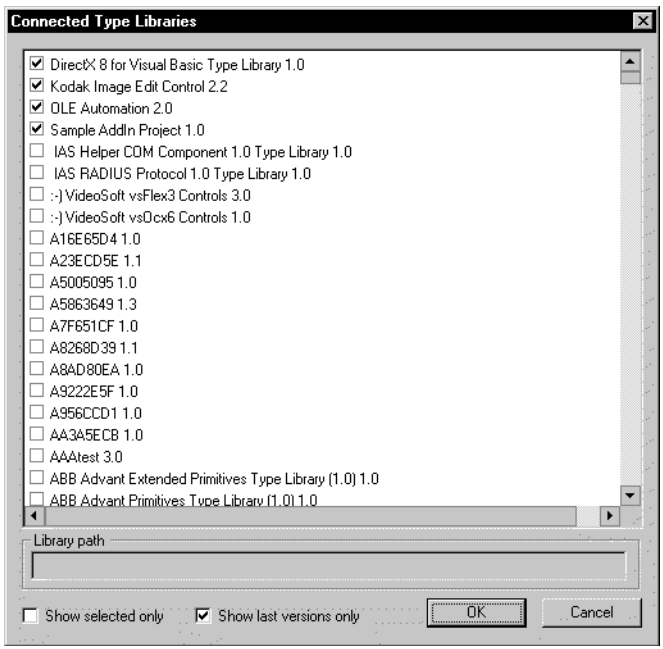


Figure 293. Connected Type Libraries Dialog Window

A list of all type libraries registered on your system is displayed. You can restrict the list using the following options:

- **Show selected only**.
  Only the selected type libraries will be shown. You get an empty list, if no type library is selected.
- **Show last version only**.
  Only the latest version of a type library is shown. Older versions are hidden.

To start the **Type Libraries Browser**, select **Tools > TypeLibraries Browser** in the Script Editor.

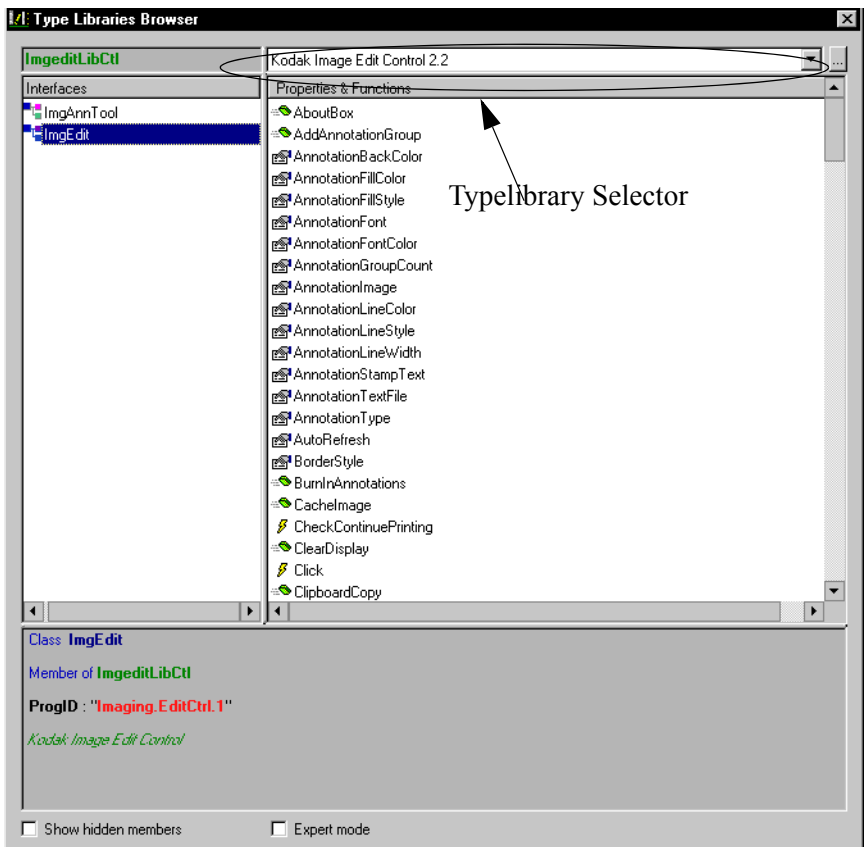


Figure 294. Type Libraries Browser Window

You define the type library to browse by selecting one of the announced type libraries in the selector at the upper right corner. The browse button invokes the "**Connected Type Libraries**" dialog window.

On the left side, the interfaces of the type library are listed. If you deselect the "**Expert mode**" only classes respectively interfaces are shown, that are accessible

via VBScript. Selecting the "**Expert Mode**" shows all classes and interfaces of the type library.

The right side shows the functions and properties of the interface or class selected on the left side.

Every item has an icon as described in the following Table:

Table 34. Type Libraries Browser Icons

Icon	Description	Remarks
	Coclass	
	Interface	
	Source interface	shown in expert mode only
	Enumerator	
	Module	
	Type	
	Const	
	Default property	
	Property	
	Event	
	Method	
U	Unit	shown in expert mode only
A	Alias	shown in expert mode only

Additional information about the currently selected element and hints for its usage are shown at the bottom of the dialog.

Debugging Scripts

You debug Scripts within the same environment, you edit them. Editing and debugging are different modes which is indicated by toolbars and docking windows.

The basic process of debugging scripts consists of the following tasks:

- Start debugging the script you are currently editing in your Script Editor.
- Start debugging by attaching the debugger to a script that is already running.
- Stop script execution by issuing a break command. You can also set a *breakpoint* in the script where the debugger will stop automatically. When you stop a procedure, its source is displayed.
- Inspect the state of the script by examining the values of variables or properties and the list of running procedures.
- Control the execution of individual statements or procedures and watch the effect by watching the values of variables or properties.
- Skip over or walk through procedures called by the current procedure.

Attach to a Running Script

Select the **Debug >Attach** command. The **Running Scripts** dialog displays a list of all currently running scripts in your Engineering Platform. You can select a script by name. The numbers in brackets indicate the thread ID of the thread, in which the script runs. It is additional information to differentiate two scripts with the same name. The script name within the list is composed from the Aspect Objects name and the Script Aspect name according to the following rule:

<Aspect Object Name>::<Script Aspect Name> (<Thread ID>)

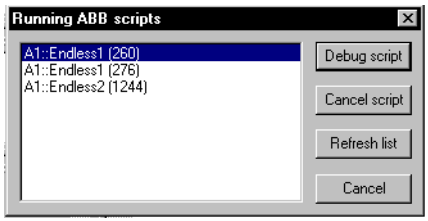


Figure 295. Running Script Dialog

The button "**Debug Script**" will attach the Script Debugger to the selected script and break it at the next statement.The button "**Cancel Script**" will cancel the selected script at the next statement.

"Next" means the current statement has to be finished.

Scripts executed via trigger conditions or menu verb invocation seem to suspend the Engineering Workplace. However, you can also attach a debugger to such a script by starting a new Engineering Workplace application, open an arbitrary Script Aspect in the Script Editor and attach the debugger.

If you open a second workplace to debug a running script, take care you run both workplaces on the same project / system.

How to Break a Running Script

To suspend (break) a running script use one of the following methods:

1. Break a Script running in debug mode
   In case of the script is started in **Debug Mode**, use the menu command **Debug > Break Script** to break the script.
2. Break a Scrip, running without debugger.
   Use the **Debug> Attach** command to get control of the Script.

The script execution is suspended at the end of the current statement. Some VBScript statements require a user interaction to conclude (e.g. MsgBox). If breaking a running script seems not to work properly, check for hidden message boxes created by your script code.

Using Variables Window

The variables window is used to examine the values of variables or properties. All variables in current context are displayed in this window.

The variable window is available only in debug mode.

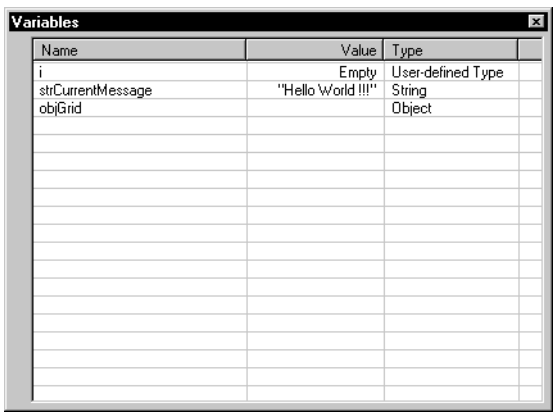


Figure 296. Variables Window

The Script Debugger enables you to expand values of object variables up to a depth of three levels.

There are two buttons on the toolbar, which are available also in the menu **Debug > Windows**:

Table 35. Toolbar Buttons

⊞	Expand object's members for one level
⊟	Collapse expanded variables one level

The result of expansion is shown in Figure 297.

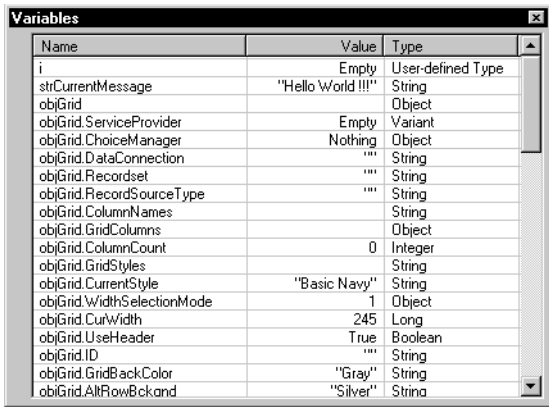


Figure 297. Expanded Object in Variables Window

Using Variables Window with Expanded Variables may block Script Editor and workplace.

Using Watch Window and Quick Watch Window

The **Quick Watch Window** is used to examine the values of variables, properties and expression. For details see Quick Watch Window.

The **Watch Window** is used for permanent examination of the values of variables, properties and expressions after each operation. For details see Watch Window.

Quick Watch Window

The **Quick Watch Window** is used to examine the values of variables, properties and expression.

The Quick Watch window is available only in debug mode.

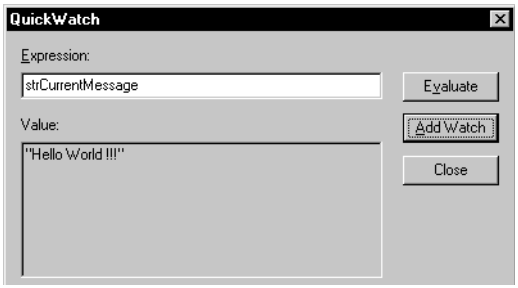


Figure 298. Quick Watch Dialog

Evaluate expression

To evaluate the expression in the text field, select “Evaluate”. The expressions result is shown in the Value field

Add Watch

You can add the expression to the **Watch Window** using the **Add Watch** command. See Watch Window for details.

Watch Window

The **Watch Window** is used for permanent examination of the values of variables, properties and expressions after each operation. The Script Debugger will evaluate all expression after each step.The Watch Window it’s only in the debug mode available.

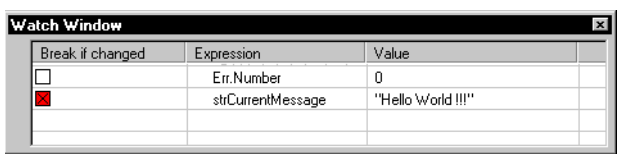


Figure 299. The Watch Window

Add Variables to the Watch Window

You add variables to the watch window either through the

- **Quick Watch** dialog**.**
  For details see Quick Watch Window.

or via

- Drag and Drop
  Select the desired variable in the **Script Editor** and drag it with the mouse to the **Watch Window** where you drop it.

Delete Variables in the watch window

Select the variable you wish to delete. And invoke the **DEL** key.

Break if Variable Changed

If the check box in column "**Break if Changed**" is checked, the Script Debugger will break the running script at the statement, that caused a modification of the expression.

Settings for Automatic Script Execution

Scripts can be automatically executed by:

- **Trigger events** when an Aspect is modified or an Aspect Object is placed in a structure.
- The user selecting an Aspect Object or Aspect context menu item. This will invoke a Function or Sub of a Script.

The Script Options Dialog

Automatic script execution is configured in the **Script Options** dialog, which is invoked in the Script Editor via **Tools > Script Options**.

The **Script Option** dialog comprises of two sections:

- **Trigger Conditions Dialog** Tab, see Trigger Conditions.
- **Menu Verb Dialog** Tab, see Menu Verb Settings.

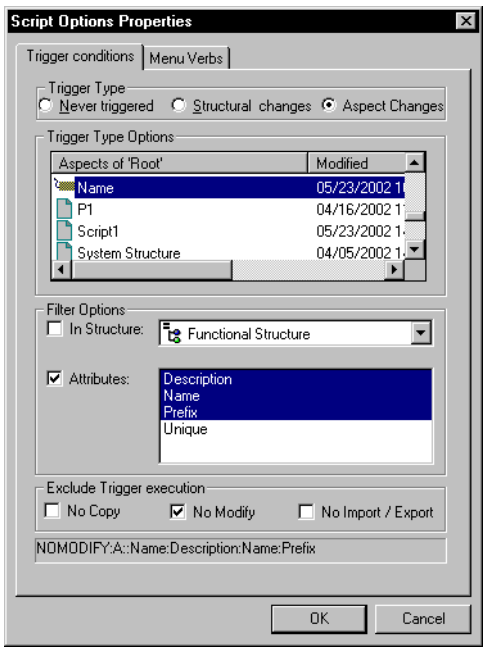


Figure 300. Script Options Dialog

Trigger Conditions

A trigger condition defines the event which leads to an implicit (automatic) activation of a script. To configure the activation events, select the **Trigger Conditions** section.

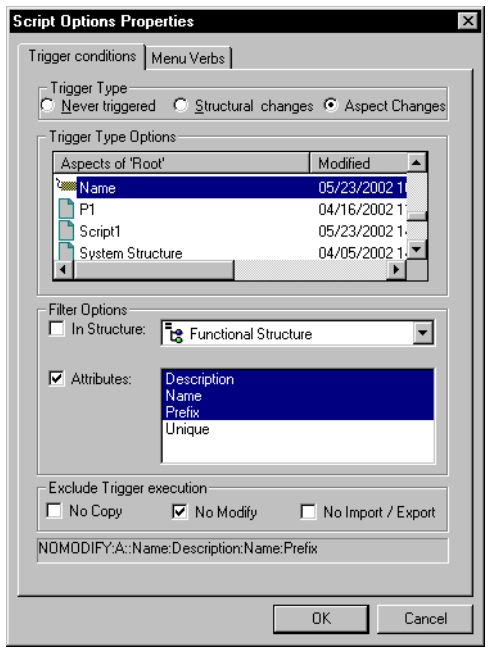


Figure 301. Trigger Conditions Dialog

Trigger types

There are three basic classes of trigger types:

- **Never triggered**
  This option disables automatic execution of a script.
- **Aspect** triggers, which fire on modifications on Aspect level. An Aspect of the Aspect Object, which owns our Script Aspect changes and leads to the execution of the script. Select the Aspect to observe for modifications in the **Trigger Type Option** section.
- **Structural** triggers, which fire on structure modifications on Aspect Object level. The Aspect Object, which owns our Script Aspect is placed or moved in a structure or children of this Aspect Object are placed or moved. Select the

structure operation to observe in the **Trigger Type Option** section. The structure operation can be any combination of the following actions:

– **Child Insert**
The script is executed, when a child is inserted beneath the Aspect Object, which owns the Script Aspect.

– **Child Move**
The script is executed, when a child of the Aspect Object, which owns the Script Aspect, is moved.

– **Child Remove**
The script is executed, when a child is removed beneath the Aspect Object, which owns the Script Aspect.

– **Parent Change**
The script is executed, when the Aspect Object, which owns the Script Aspect, is moved or placed in a structure.

– **Instantiate**
The script is executed, when the Aspect Object Type, which owns the Script Aspect, is instantiated in a structure.

Filter Options

In order to restrict the script execution to a specific structure or a set of Aspect Properties you can specify filter options.

Exclude trigger execution

Sometimes it is necessary to restrict the execution of a script to specific actions. Therefore you can specify exclusion conditions:

- **No Copy**
The script execution is suppressed during Copy & Paste actions.
- **No Modify**
The script execution is suppressed when modifying an Aspect.

- **No Import / Export**
  The script execution is suppressed during Import / Export or Backup / Restore of an Aspect Object.

Triggered scripts run within the transaction context of the System 800xA platform. Therefore, you should restrict the scripts regarding the execution time. Scripts causing time-outs within the transactions may lead to an undesired behavior of the System 800xA platform. In especially, you should avoid message boxes in triggered scripts.

Trigger option Exclude trigger execution, trigger conditions are set to:

- Structural changes
- Parent change
- Exclude Trigger Execution No copy
- The trigger fires if the parent changes even if the Aspect Object owning the script is copied to a new parent.

Menu Verb Settings

The context menu is the pop-up menu displayed by a right mouse click on an Aspect or Aspect Object in the Engineering Workplace. You can configure VBScript functions or subroutines to appear in the context menus and get invoked on user selection. Two different types of menu verbs exist:

- **Object Verbs:**
  The script function will be displayed in the pop up menu of the Aspect Object, which owns the Script aspect.
- **Aspect Verbs:**
  The script function will be displayed in the pop up menu of the Script Aspect.

To configure the menu verbs, select the **Menu Verbs** section.

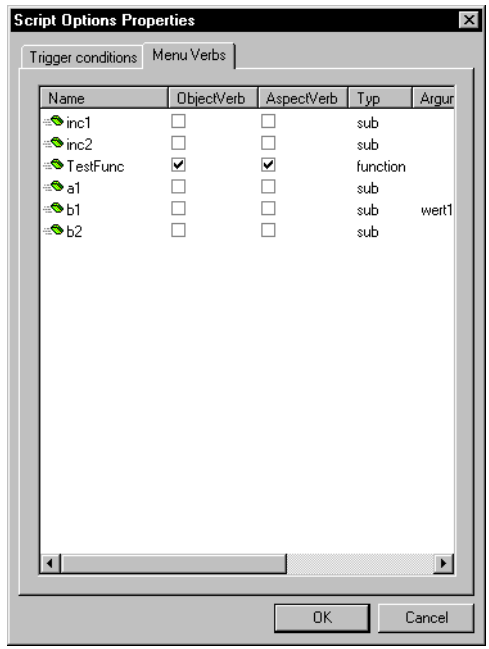


Figure 302. The Menu Verb Dialog

List of Functions

All functions and subroutines of a script functions are listed in the **Menu Verbs** section.

The following information is displayed for every function:

- **Name**
  The function name as given in the source code of the script.
- **Object Verb**
  The indication of the function being activated as an Object Verb or not.
- **Aspect Verb**
  The indication of the function being activated as an Aspect Verb or not.

- **Type**
  The function type (function or subroutine).
- Argument
  The function argument list of the function. You may not configure functions with arguments as menu verbs.
- **Description.**
  In the last column a function description is optionally displayed. The function description is an apostrophe (') separated comment in the first function declaration line in the script source text.

The Rules for Menu Verbs:

- Every function or sub **without** any arguments it's a candidate for a Menu Verb.
- Avoid any **global code** in your script, since it is executed every time your Menu Verb function is invoked.

Working with Trace Window

Configure the Trace Window

Select **File > Options** to invoke the Trace Window options dialog.

Figure 303. Trace Window Options Dialog

You can adjust the following settings:

- **Maximum number of log windows**
  Gives the maximum number of concurrently traced scripts. Each new script will clear one of the old windows, if this number is reached.

- **Maximize child windows**
  If this control is checked, all trace windows are maximized after their creation.

Close Connection to the Running Script

You can close Trace Window only, if the script is completely executed or (in other words) no script is connected to the Trace Window application. You can break the connection to the running script with the **Tools > Break Connection** command. The connection to the script will be broken and you can close the Trace Window application.

Scripting Guide

This chapter describes the usage of Script Manager. It is meant for programmers who need the scripting capability inside the System 800xA platform, but do neither know VBScript nor the Aspect Automation Model.

Extensions to Standard VBScript

The complete VBScript language is supported by the Script Manager, however there are some additions and a special restriction.

Name Restrictions

Function/Sub names must be unique:
In VBScript you may define two or more functions with the same name - at script execution usually the last one will be used. This is considered as bad and dangerous and therefore not allowed. If identical function names are found you will get an error message like "Function/Sub 'FooBar' already defined in Script Test" and the script will not be executed.

Accessing Aspect Objects

Generally a script is owned by an Aspect Object (AO). The **This** keyword provides access to this object. It is of the VBScript data type Object.

Example: `Msgbox "This Name=" & This.Name`

Within the Aspect Automation Model **This** represents an Aspect Object, so you can get the properties by using the dot notation.

```
Option Explicit

Dim i, Count , Aspects

Count = This.Aspects.Count-1
Set Aspects = This.Aspects

for i = 0 to Count
   Trace CStr(i) & " " & Aspects(i).Name
   Trace "  Date: " & Aspects(i).CreationTime
next
```

Creation and modification time: If you use the Special Script Parameter "this" to access your own Object, it returns the modification and creation time reflecting Greenwich Mean Time without considering the current time zone settings. Add the difference to your time zone according to the following script.

```
Dim ws, intTZOffset, TZOffset, HexVal, ModiTime

'Read time zone offset hex value from Registry.
Set ws = CreateObject("WScript.Shell")
intTZOffset = ws.RegRead("HKLM\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\ActiveTimeBias")

'Convert Offest to our Date/Time format.
strTZOffset=abs(intTZOffset/60) & ":00:00 AM"
TZOffset = TimeValue(strTZOffset)

'Add the Offset
if intTZOffset > 0 then
  ModiTime  = FormatDateTime(this.ModificationTime - TZOffset)
  CreatTime = FormatDateTime(This.CreationTime - TZOffset)
else
  ModiTime  = FormatDateTime(this.ModificationTime + TZOffset)
  CreatTime = FormatDateTime(This.CreationTime + TZOffset)
end if

'Output
MsgBox ("ModificationTime : " & ModiTime)
MsgBox ("CreationTime : " & CreatTime)
```

Include Other Scripts

For better maintainability and library handling an **Include** directive was added. This directive enables you to include other scripts from the same Aspect Object or the Library Structure.

The statement works like the **#include** in the C language. The content of the specified script is inserted at the current location. Recursive inclusions are detected by the script engine and reported as an error.

Suppose you have a script procedure named "Check_power_supply" which is used in 5 scripts of an object. Instead of storing and maintaining all these copies, the procedure is stored in its own script named CheckPowerSupply. This script is then included by other scripts.

```
'Sample for using the include statement
Include "CheckPowerSupply"
```

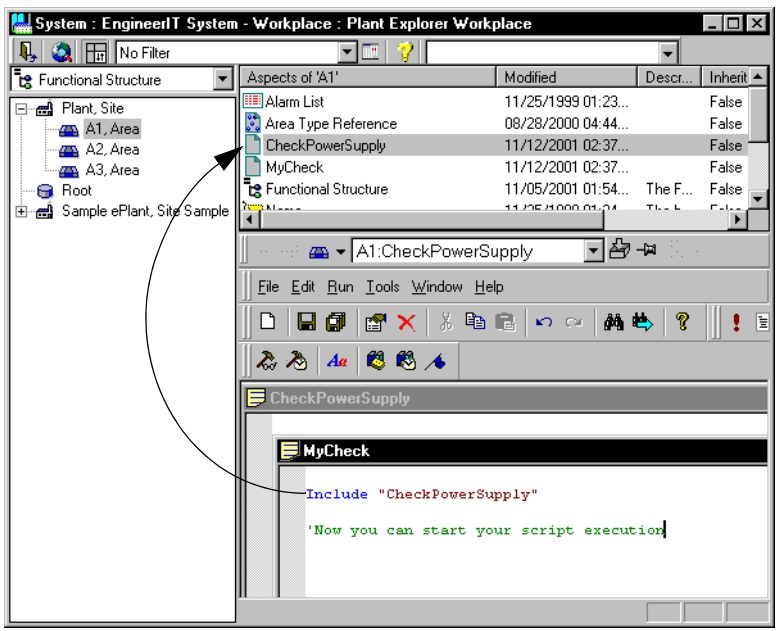


Figure 304. Example for Including a Script

Obviously there are routines, which are useful for a great number of scripts. They can be stored in a dedicated object in the library structure (named for example “Scripts”) and accessed using the following include directive

```
'Sample for including a script from the library structure
Include "Scripts/Demo/ObjectInfo"
```

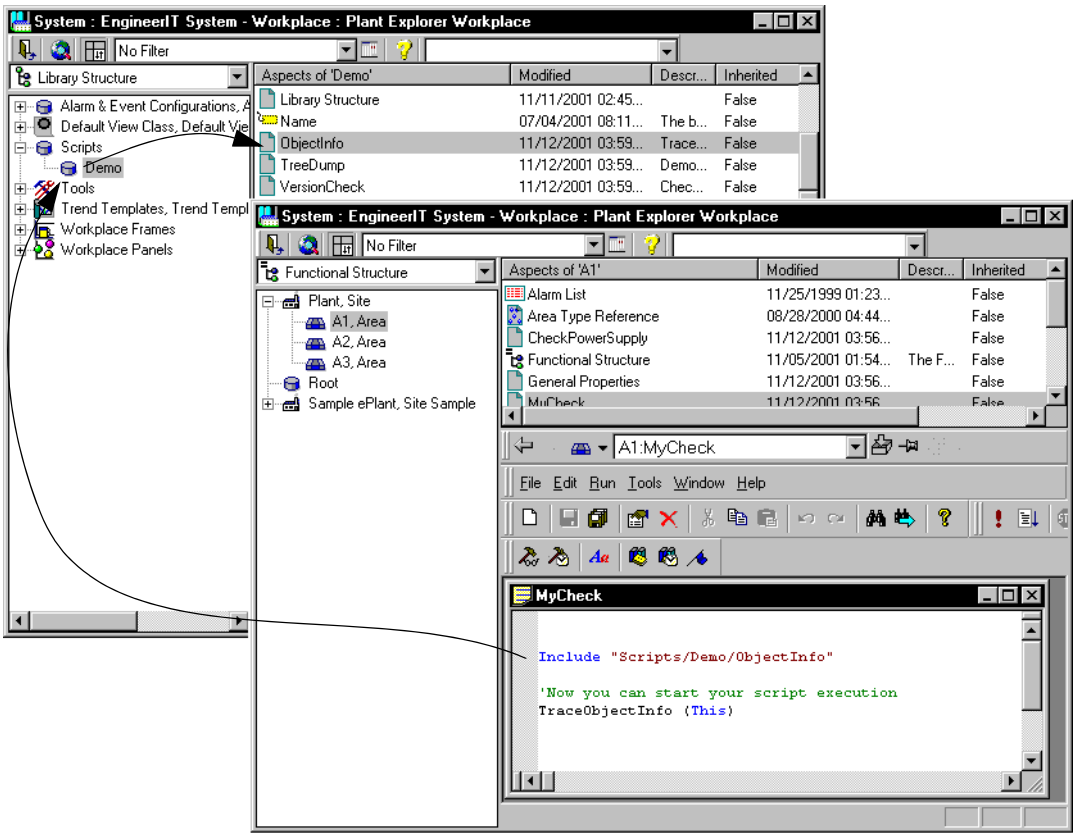


Figure 305. Include Script From Library Structure

If the include directive contains a path seperator “/”, the script is included from the library structure.

Additional Functions Overview

Additional script functions are shown in the following Table:

Table 36. Additional Script Functions

Function	Description
Trace	Show message in a Trace Window
TraceOn	Enables tracing of a script independent from global settings (see Settings)
ScriptName	Returns the name of the current executed script
SetScriptParameter	Passes a named parameter to a script
GetScriptParameter	Retrieves a named parameter set by a script
ConnectObject	Enables event handling for COM objects in a script
DisconnectObject	Disables event handling for COM objects in a script
RunScript	Runs any script and allows to pass values to the called Script
GetPropValue	Reads a Property Value
GetOldPropValue	Gives you access to the old property value from the attribute in a Script which is called via an Aspect Trigger
ADSync	Synchronize the Aspect Directory Server
SNSSync	The Sync method synchronizes the Structure and Name server with the aspect directory
Sleep	Suspends the Script execution
WriteData	Save Value within your Script and publish them as OPC
ReadData	ReadValue within your Script
RemoveData	DeleteValue entry

Additional Functions Description

- **Trace** (*strMessage as String*)
  this function is used to display a message in separate windows provided by the Trace Window application.

To use tracing, you first have to enable it in the Global Script Manager Options Dialog.

- **TraceOn** - You may supersede the global script settings for tracing by calling the TraceOn function within a script. Only the actual script is affected, other scripts continue without tracing. For example, error handlers are a good place to use TraceOn. So you get a nice error log, without the runtime disadvantages of tracing everything:

```
Example:
  if err <> 0 then
       TraceOn
       Trace "Error " & err.description
  end if
```

- **ScriptName** - Returns the script name.
  This is also useful at tracing messages of several scripts.

```
Example:
  if err <> 0 then
       TraceOn
       Trace "Error in script " & ScriptName
       Trace "Error Description " & err.description
  end if
```

- **SetScriptParameter** (*sParameterName, Value*)
  Passes a named parameter to a script (for example see Special Parameters for Structure Triggers)
- **GetScriptParameter** (*sParameterName, Value*)
  Retrieves a named parameter set by a script (for example see Special Parameters for Structure Triggers).
- **ConnectObject** (*Object, sPrefix*)
  Enables event handling for COM objects in a script.
  Object - initialized COM object with OLE Source Interface
  sPrefix - Prefix for the event handle subroutine.

(for additional info and example see Event Handling for COM Objects).

- **DisconnectObject** (*Object*)
  Disables event handling for COM objects in a script.
  Object - initialized COM object with OLE Source Interface.
  (for additional info and example see Event Handling for COM Objects).

- **RunScript**(*Script Aspect, Value**1**...Value**n***)
  Runs the script identified by the Aspect. You can pass optional parameter values to the called script, which can access these parameter values via GetScriptParameter, where the Parameter Name is Para**1** to Para**n** for the Value 1 to Value n.

- **GetPropValue (***Aspect, PropertyName***) returns** *Value*
  Reads an Aspect Property. The Property has to be identified by its Name The property value will be returned.

```
Example:
Dim MyValue
For each Aspect in this.Aspects 'Go through all Aspects
  if Aspect.Name = "Name" then  ' Find the Name Aspect
    MyValue = GetPropValue (Aspect , "Prefix")
    MsgBox "The Prefix is :" & MyValue
  end if
next
```

- **GetOldPropValue**(*Aspect, PropertyName*) **returns** *Value*
  Reads the old value of an Aspect Property. The Property has to be identified by its Name. The old property value will be returned. This function works only if script execution was caused by an attribute trigger. The old value is the property value before the modification occurred, which caused the script activation. When you call the function outside a trigger, you will get the current property value.

- **ADSync()**
  This function synchronizes the AD server with the object manager. This implies that all trackers that are waiting to be fired are handled. For more details see Sigma SDK.

- **SNSSync()**
  This function synchronizes the Structure and Name server with the aspect

directory. This method should be used after modifications have been performed and you want to ensure that the name server is aware of the changes. For more details see Sigma SDK.

- **Sleep(***time in milliseconds***)**
  The Sleep function suspends the execution of the current script for the specified time in milliseconds.
- **WriteData**(*ValueName, Value*)
  This function writes persistent data. If a data entry with the given name already exists, the associated value is overridden. Otherwise a new data entry is created for the given name.
- **ReadData**(*ValueName*) returns *Value*
  This function reads persistent data.
- **RemoveData**(*ValueName*)
  This function removes the data entry for an existing script property.

Special Parameters for Structure Triggers

When a script is triggered by a structural change, the script engine initializes some special parameters (refer Table 37) to inform the script about the type of change. As a result of this the script may set a return value to deny the changes.

Table 37. Special Script Parameters

Name	Type	Set by
Reason	VT_BSTR	800xA
OldParent	VT_DISPATCH	800xA
Child	VT_DISPATCH	800xA
Return	VT_BOOL	Script

Reason indicates the type of change going on and is always defined. It may be one of the values **CR**, **CI**, **CM**, **IN** or **PC** (described in Table 38).

Table 38. “Reason” Parameter Description

Value	Type Of Change	Applies to
CR	Child Remove	Parent nodes
CI	Child Insert	Parent nodes
CM	Child Move	Parent nodes
PC	Parent Change	Child nodes
IN	Instantiation	

Child
The child object, that is about to be moved. This parameter is undefined for a child node, whose parent is changing.

OldParent
Defined for child nodes that are moved or deleted.

Return
A script may set the Boolean parameter Return to FALSE in order to cancel the structural change.

The **OldParent** and **Child** parameters are not defined for all nodes under all conditions. Before accessing variables set by **GetScriptParameter**, use the VBScript functions **IsEmpty** or **IsObject** to check for valid objects.

Suppose a child node C is moved within a structure from parent A (see Figure 322) to B (see Figure 323) and each node has a script triggered by any structural change.

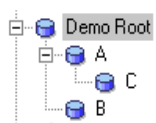


Figure 306. Before Change of Structure

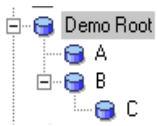


Figure 307. After Change of Structure

sIf none of the scripts cancels the move, they are called in this order with the indicated parameters as describe in following Table:

Table 39. Script Parameters

Node	Reason	OldParent	Child
A	CR	Empty	C
B	CI	Empty	C
C	PC	A	Empty

Example script "**StructureChange**"(see Table 40):

```
'
' Display script parameters for structural changes
' in a message box.
' If bAskForCompletion is set, ask weather to
' commit the changes.
Sub StructureChangeMsg(bAskForCompletion)

Dim OldParent, Child, Reason, sMsg, answer

    ' Get Script Parameters
  GetScriptParameter "OldParent" , OldParent
  GetScriptParameter "Child"     , Child
  GetScriptParameter "Reason"    , Reason
    ' Build the message
  sMsg = This.Name & ": A structural change is going on!" &
vbCrLf & vbCrLf
  sMsg = sMsg & "Structure Trigger Parameters:" & vbCrLf
  sMsg = sMsg & vbCrLf & "  Reason:     " & vbTab & Reason
```

```
   if IsObject(OldParent) then
     sMsg = sMsg & vbCrLf & "  OldParent:  " & vbTab &
OldParent.Name
   end if
   if IsObject(Child) then
     sMsg = sMsg & vbCrLf & "  Child:      " & vbTab &
Child.Name
   end if

   if bAskForCompletion = TRUE then
     sMsg = sMsg & vbCrLf & vbCrLf & "Commit changes ?"
     answer = Msgbox (sMsg, vbYesNo + vbQuestion)
     if answer = vbNo then SetScriptParameter "Return", FALSE
   else
     Msgbox sMsg
   end if

end sub
```

For Trigger Type Child Move fires the trigger fires only at the first time a given object has been moved The next time the same object will be moved the trigger does not fire.

Event Handling for COM Objects

One of the most powerful scripting functions is **CreateObject**. You may instantiate any COM object (that supports IDispatch, naturally) and use its services. But CreateObject lacks one important thing: a script can never react on events fired by the created component. This is where the functions **ConnectObject** and **DisconnectObject** come in. **ConnectObject** let a script handle the events of a component, **DisconnectObject** disables event handling by the script.

In order to receive OLE events in the script, the COM component must be declared as event source by calling the function **ConnectObject (***Object, sPrefix***)**.

The **ConnectObject** function has a two parameters:

- **Object** - initialized COM object with OLE Source Interface.
- **sPrefix** - Prefix for the event handle subroutine.
  The name of the event handling subroutine is comprised of two parts:

– the first part - is the **PREFIX**

– the second part - the name of the event in the source component.

Example:

We have a COM component **“ScriptDlg1.Dialog2”**. This component fires the public event:

Event OnClickButton (Index As Integer).

The parameter **Index** passes a button ID to the event handler.

The event handler for this event is:

```
Sub Event_OnClickButton( Index )
.....
.....
End Sub
```

The part “**Event_**“ is the prefix and **“OnClickButton”** is the the name of the event in the source component.

To receive OLE events from **ScriptDlg1.Dialog2** component in your script, use the following pattern:

```
...
...
Sub Event_OnClickButton( Index )
...
...

end sub

Dim dlg

On Error Resume next

Set dlg = CreateObject("ScriptDlg1.Dialog2")
call ConnectObject(dlg, "Event_")
...
...
```

In some cases you will need to turn off the reception of OLE events. To disconnect an OLE - event source from the script, call

DisconnectObject (Object).

After the call to this function, the event handler will not be called any longer.

In our example this look as follows:

```
...
...
call DisconnectObject ( dlg )
...
...
```

DisconnectObject does not destroy the object. Only event handling is disabled.

User can reconnect to the object using the **ConnectObject(...)** function and your script will receive OLE events from the COM object.

See also the demo scripts **Dialog**2 for more advanced example. (How to find this script, see Table 40).

Data Persistence and Aspect Parameter Support

It is possible to store data values within your script to a persistent data storage and to read back the values from the data storage again.

All Values saved with WriteData are saved in a Script Property Bag. Values in the Script Property Bag are published as Aspect Parameter values and therefore can be read and subscribed on by other applications which use the System 800xA platform subscription mechanisms. Within your Script you read the values with Read Data.

800xA Library

After installation of the 800xA System Extension, you will find the following scripts in the Library Structure:

Table 40. Script Library

Path	Script name	Description
Scripts\Demo	ObjectInfo	Trace Debug information about an ASO and its aspects
Scripts\Demo	TreeDump	Demonstrates the usage of the StructureCursor
Scripts\Demo	FindScripts	Demo: Search a structure for scripts
Scripts\Demo	Dialog1	Demonstrates a simple user interface
Scripts\Demo	Dialog2	Demonstrates the usage of ConnectObject
Scripts\Demo	Calculator	Calculator Demo

System 800xA Platform Related Functions

Script Manager is an aspect system in System 800xA and supports therefore all platform functions like Electronic Signature, user Log-Over, Backup / Restore, Import / Export. The most functions are covered by the platform, however Script Manager provides some extensions as described below.

Permissions and User Roles

The user of the Script Manager Professional is an application engineer. The application engineer has good knowledge about working with the Aspect Object Framework and Engineer IT. Basic programming skills are required.
The user of the Script Manager Basic can be a system engineer, an operator, or a maintenance employee.

Authority

Script Manager supports the platform's authority concept of the platform by checking

- Permissions granted to a user.
- Roles connected with this user.

This is done against the permissions required for actions in Script Manager and roles required for user interface availability.

To create, edit or debug a Script you need the application engineer role.

The Script execution by a trigger event is not restricted by user role or Process Portal permissions.

Backup/Restore and Export/Import

Script aspects of the Script Manager can have dependencies by including script code which are stored within other Script aspects. When exporting a script aspect these dependencies will be reported. However, Script Manager aspects having dependencies have to be manually exported.

Also the Script property bags will be added to the dependency list. This means if a Script is exported, all dependent Scripts and Script property bags will be exported too.

Audit trail

The Script Manager supports Audit trail functions for the following events:

- Changes in the Script code.
- Changes of Script Manager persistent properties.
- Changes of Trigger Types.

Error Messages

If any errors occur, error messages are displayed. The most frequent errors when working with scripts will be syntax errors in VBScript. These errors are reported by a dialog box.

Figure 308. Error Message Reporting Syntax Error

If you are working in the Script Editor the errorneous line is marked with a red bookmark.

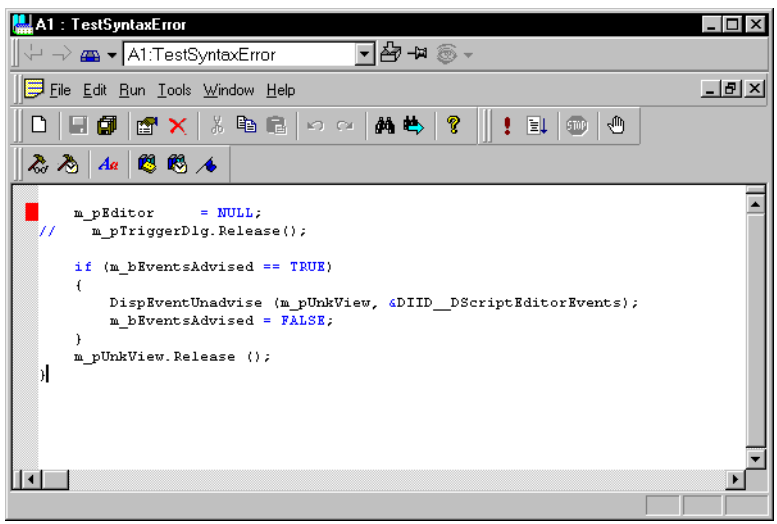


Figure 309. Error Mark in Script Editor

Appendix A Engineering Templates

Engineering Templates are positioned as an aid for engineers supporting various engineering activities like getting cross-reference lists, creating variables for applications or programs, etc.

There are two classes of Application Templates:

Bulk Application Templates - Templates, which could be started from Folder “Engineering Templates”. These templates need not to be integrated as aspects but can be used to handle data stored in different aspect systems.

Example: The cross-reference list supports a report of variables and their usage in the Control Builder M.

Document Manager Templates – These templates do only function in the scope of an Aspect Object and hence need to be handled as Document Manager Aspects.

Example: The TagSheet presents data, taken from different aspects of the same object, in a certain format.

Working with Engineering Templates

The **Bulk Application Templates** are located in a desktop folder ‘**Engineering Templates**’. User can open this folder by either of the following methods:

- Double-click and open the desktop folder ‘**Engineering Templates**’.
- Right-click on any of the aspect objects and select **Advanced > Engineering Templates**.

The following templates are available within the Engineering Templates folder:

- BDM_DiagramRef_Var_Adv.xlsx.
- BDM_DiagramRef_Var_Basic.xlsx.
- BDM_for_Function_Diagrams.xlsx.
- TrendConfig.xlsx.

- LogConfig.xlsx.
- Label_Report.xlsx.

For more information, double-click and open the '**_Documentation**' folder available in the desktop-folder Engineering Templates to find the descriptions for the different Engineering Templates.

Refer to Existing Predefined Templatesto learn about the **Document Manager Templates**.

Data in all columns (for example, Name and Description) of BDM templates like **TrendConfig.xlsx** and **LogConfig.xlsx** are always updated in Aspect System irrespective of the fact that the value of the columns is changed or not.

Appendix B Property References

A Property Reference is a dynamic link to a property of an aspect of any object.

A reference string, is a string that contains all information necessary to identify and access one single (aspect) property in an Aspect System

You can use Property References e.g. in Excel workbooks and Word documents.

This appendix describes the different types of **property references** and how to work with the Reference Control to insert the references.

How to call the Reference Control is described in Section Bulk Data Manager and Document Manager

Working With References

You can define a reference string to retrieve and set (depending on Aspect System) the value for a property of an aspect.

Each property of all configured aspects can be accessed with this reference, provided the aspect has published its data, i.e. the properties are externally accessible.

Example:
To get the description for circulation pump EU201, define a reference string for the object “EU201”, aspect “Name” and property “Description” set the start object to “EU201” then the reference string will be “.:Name:Description”

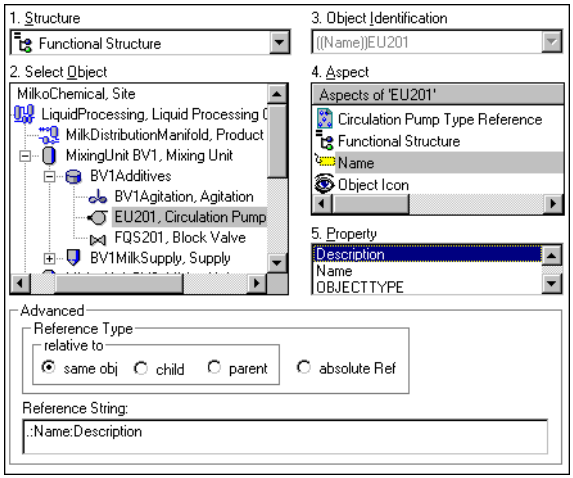


Figure 310. Reference string shown in the Insert Property References dialog

Use small fonts settings on your computer if you work with Property Reference dialog, otherwise this dialog will not be properly formatted.

Types of References

There are three different types of references:

- Relative reference to the same object
- Relative reference to a child or parent object
- Absolute reference to any object

The reference type depends on the relation between the object you refer to (i.e. the object where you started to browse), and your current object.
You can see the type of reference string, the **Reference Type**, in the **Advanced** group.
It will be automatically set when browsing.

The reference type examples below will show you the different types.

Relative Reference to the Same Object

If your current object is the same object where you started to browse, the reference type is “same obj”.
As you are referring to the same object, the reference string contains only the aspect name and property name.

Example:
You want to get the description of the current object “EU201”:
the object “EU201” is already selected in the **Select Object** list box. You select the **Aspect** “Name” and **Property** “Description”.
The reference string is: “.:Name:Description”.

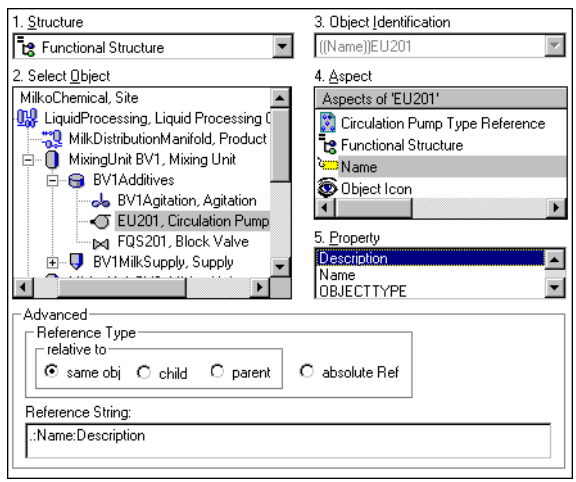


Figure 311. Relative Reference to the Same Object

Relative Reference to a Child or Parent Object

Relative Reference to a Child Object:

If your current object is below the object where you started (i.e. a son, or grandson, and so on) the reference type is “child”.
The reference string contains the structure, keyword “down”, object identification, aspect name and property name.

Example:
You want to get the description of the object “EU201” in the Functional Structure. The start object you refer to here was e.g. “MixingUnit BV1”. The current object “EU201” is a child of “MixingUnit BV1”.
You select the “Functional Structure” in the **Structure** drop down list box, then the object “EU201” in the **Select Object** list box. You select the **Object Identification** “((Name))EU201”, **Aspect** “Name” and **Property** “Description”.
The resulting reference string is:
“.(Functional Structure)(down)((Name))EU201:Name:Description”

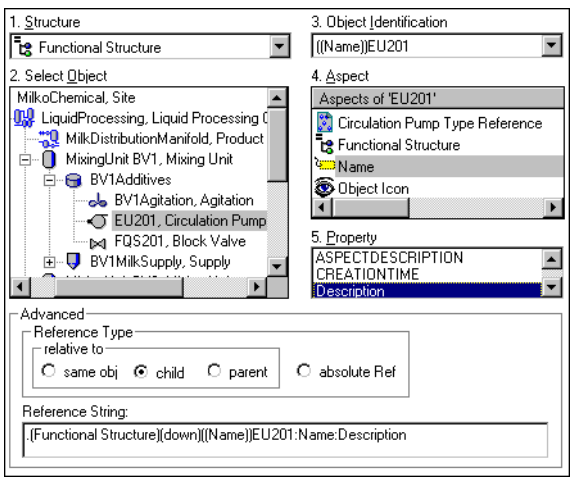


Figure 312. Relative Reference to a Child Object

Relative Reference to a Parent Object:

If your current object is above the object where you started (i.e. a parent, or grandparent, and so on) the reference type is “parent”.
The reference string contains the structure, keyword “up”, object identification, aspect name and property name.

Example:
You want to get the description of the object “MixingUnit BV1” in the Functional Structure. The start object you refer to here was e.g. “EU201”. The current object “MixingUnit BV1” is a parent of “EU201”.

You select the “Functional Structure” in the **Structure** drop down list box, then the object “MixingUnit BV1” in the **Select Object** list box. You select the **Object Identification** “((Name))MixingUnit BV1”, **Aspect** “Name” and **Property** “Description”.
The reference string is:
“.(Functional Structure)(up)((Name))MixingUnit BV1:Name:Description”

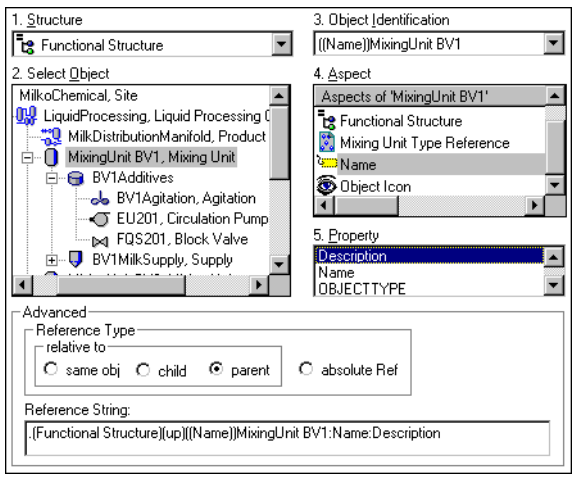


Figure 313. Relative Reference to a Parent Object

Absolute Reference to any Object

If your current object is not relative to a child, parent or the same object, it is of type “absolute reference”.
The reference string contains the object identification, aspect name and property name.

Example:
You want to get the description of the object “MilkReception”. The start object was e.g. “EU201”. The object “MilkReception” is not relative to “EU201”, so the type of reference is absolute.
You select the object “MilkReception” in the **Select Object** list box. Then you select the **Object Identification** “((Name))MilkReception”, **Aspect** “Name” and **Property** “Description”.

The reference string is:
“((Name))MilkReception:Name:Description”

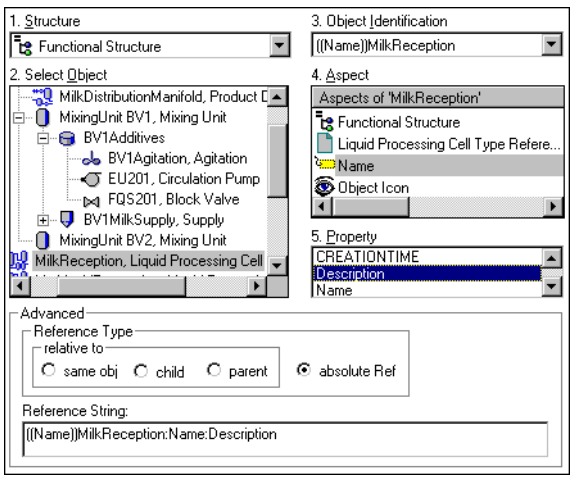


Figure 314. Absolute Reference

Methods to Enter a Reference

You can choose between two different methods to enter the reference string:

- browse through the structures and objects:
  You browse through the structures and objects and select the **Structure**, **Object**, **Object Identification**, **Aspect** and **Property** (see Figure 310). Depending on your selection and the object you started to browse, the corresponding **Reference Type** will be set and the **Reference String** will be built. The **Reference Type** and **Reference String** can be seen in the **Advanced** group, if it is expanded.
- insert the reference string manually:
  You insert the reference directly in the **Reference String** edit box.
  You can use this function when you can’t browse, e.g. because an object and/or aspect does not exist yet.

Object Identification

Within the type of reference string there are subtypes possible.
You can choose a subtype within the **Object Identification** list box.
The Object Identification is not used when you have a reference to the same object.

The following subtypes are available:

- reference by name
  This is the basic property name of the object. As the object may have several names, you can refer to a specific one with the help of its name category, e.g. “Name”, “Relative Name”, “Location ID” or “Customer ID”.
  An object may have for example a Customer ID “Turbine A” and a Location ID “Turbine A 2nd floor”.
  The complete Object Identification holds the name category and the actual name.
  Example:
  ((Name))EU201
  ((Relative Name))Pump2
  ((Customer ID))Turbine A
- reference by designation
  - relative designation:
    The Object Identification holds the name of the designation aspect and the designation.
    Example:
    ((Functional Designation))CP27
  - absolute reference designation:
    The Object Identification holds the keyword “ARD”, and the absolute reference designation for this object (see also section Absolute Reference Designations). Example:
    (ARD)=A1.B2.C2

Thus, when the Object Identification is used you can distinguish the following kinds of references:

- Relative reference to a child or parent object, by name.
- Relative reference to a child or parent object, by designation

- Absolute reference to any object, by name
- Absolute reference to any object, by designation

The examples below show the usage of the subtypes.

Example1:
Reference by name, category name “Name”:
For this type of reference you select the object identification “((Name))EU101”.
See above, Figure 312.

Example 2:
Reference by name, category name “Relative Name”:
This type is especially useful when you build object types or reusable libraries. Let’s assume you want to build an object type “OT LiquidProcessing” which includes the objects Obj1, Obj4 and Obj5. You want to add a reference from Obj1 to its child, Obj4, which should be still valid after instantiating the object type.
You can’t add a reference via name, because the name “Obj1” will become “MyObj4” after instantiating, so the reference would be invalid then.
Therefore you choose a reference via relative name (see Figure 315). The relative name, here RO4, is still the same after instantiating (see Figure 316).
For this type of reference you select the object identification “((Relative Name))RO4”.

NOTE: There will be no entry in Object Identification, if the relative name is empty.

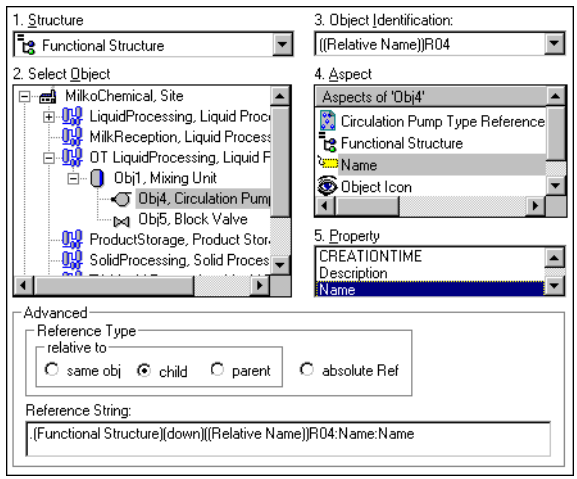


Figure 315. Reference by Name, Category Relative Name, Object Type

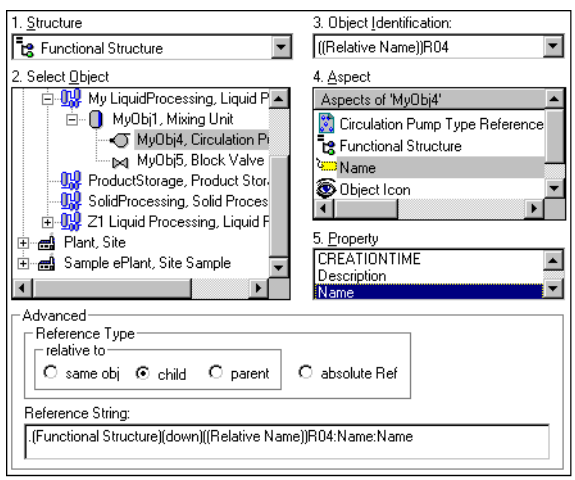


Figure 316. Reference by Name, Category Relative Name, Object Instantiated

Example 3:
Reference by relative designation:
Let's assume you want to add a reference from object Obj1 to its child object Obj4 by relative designation.
You can do this by referring from Obj1 to the name "Obj4" of object Obj4 via the Functional Designation "CP27" of Obj4.
The value for the reference string is "Obj4".
For this type of reference you select the object identification "((Functional Designation))CP27".

NOTE: There will be no entry in Object Identification, if the designation is empty.

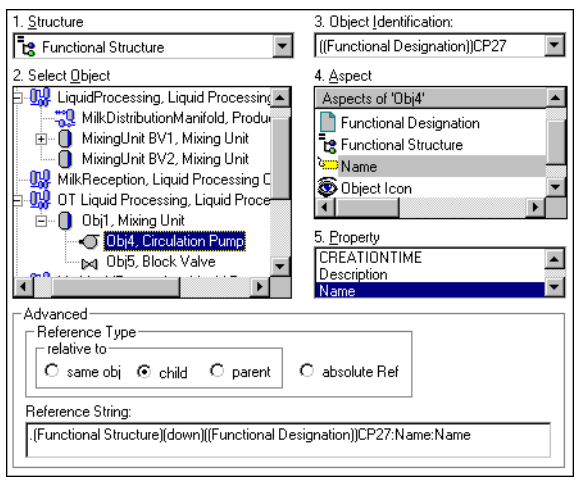


Figure 317. Reference by Relative Designation

Example 4:
Reference by absolute reference designation:
Let's assume you want to add a reference from object "Milk Reception" to object "BV1 Milk Supply".
You can do this by using an absolute reference, because "BV1 Milk Supply" is not above or below "Milk Reception".
For this type of reference you select the object identification "(ARD)=A1.B2.C2".

NOTE: There will be no entry in Object Identification, if the ARD is empty.

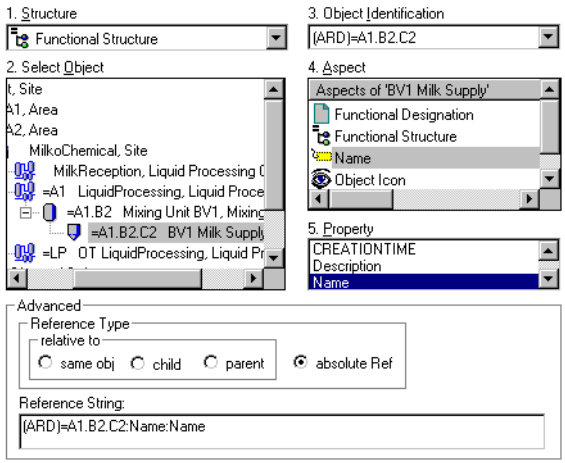


Figure 318. Reference by Absolute Reference Designation

Absolute Reference Designations

Two types of reference designations can be distinguished:

- relative:
  A relative reference is relative to the related parent or child object. It can be assigned to a certain object.
- absolute:
  Instead of using a relative reference by name to another parent or child object (see section), you can also use an absolute reference designation.
  The absolute reference designation of an object is built by concatenating the absolute reference designation of the parent object with all relative reference designations down to your current object.

Example 1:
absolute reference designation of parent is: =A1.A2
relative reference of board is: =A3

relative reference of signal is: =B2
absolute reference designation is: =A1.A2.A3.B2, if a dot is used as separator.

Example2:
an object “Signal A” has a relative reference “A2”
its parent object has an absolute reference “+A1”
absolute reference of Signal A is “+A1.A2”, if a dot is used as separator.

Building Rule for Absolute Reference Designations

The way the concatenation is performed, can be controlled by building rules. Separators may be inserted in between the designations for a better visualization.

Each relative reference designation may have a prefix. The prefix specifies the type of the reference designation. In the absolute reference of the parent object in the above mentioned example, “=” and “+” is the prefix. The prefix may also be used as separator. Whether a separator is used or not and the type of separator, is controlled via the building rules.

Available building rules are:

- PREFIX
- DOT
- PREFIX_TO_DOT
- NONE

Below is an example for the building rules. The shown objects have reference designations A1, A2, A3 and B2. The first object has a “=” as prefix, the next “=”, the next “+”, and the last one “#” as prefix.

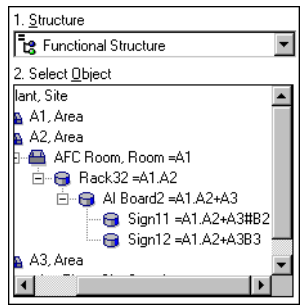


Figure 319. Example of Reference Designation

If you select the **PREFIX** rule, the prefix is used as separator between relative reference designations. When no prefix is defined for a certain relative reference designation, no separator will be inserted.
Example:
PREFIX: =A1=A2+A3#B2
=A1=A2+A3B3

If you select **DOT** rule, the dot is always used as separator between relative reference designations, except at positions, where the type of the designation changes. Here the prefix is used as separator.
Example:
DOT: =A1.A2+A3#B2
=A1.A2+A3.B3

The **PREFIX_TO_DOT** rule has the following meaning: For relative reference designations with defined prefix, a dot will be shown instead of the prefix, unless for cases, where the type changes. In this case the prefix will be shown, as usual.
Example:
PREFIX_TO_DOT: =A1.A2+A3#B2
=A1.A2+A3B3

The **NONE** rule has the following meaning: relative reference designations are concatenated to build the absolute reference, without using a separator in between. But when the type of the reference designation changes, the prefix is used as separator.

Example:

NONE:	=A1A2+A3#B2
	=A1A2+A3B3

How to Display Absolute Reference Designations

To display Absolute Reference Designations (ARD) you have to specify it in the Name Composer in Plant Explorer. Do the following steps in Plant Explorer:

1. Select the **User Structure**
2. Select a user in the user group for which you want to display the ARD
3. In the **Aspects** list select **Workplace Profile Values**
4. In the **Names:** list select **Plant Explorer Settings**
5. In the **Datasource** group click on radio button **Local**
6. Click on the **Name composer** tab
7. In the **Name categories** list select **ARD** and click the **Insert** button
8. Click the **Apply** button
9. Stop the system and start it up again to take the changes effect

How to Select a Building Rule (ARD Rule)

Do the following steps in Plant Explorer:

1. Select the **Service Structure**
2. Select AfwSNServer service
3. In the **Aspects** list select **ARD rules**
4. In the **Property Configuration** tab select a structure in the **Properties:** list.
5. In the **Value** group click the radio button **String**.
6. Enter the building rule, e.g. PREFIX_TO_DOT.

Prefix

The prefix determines the type of a reference designation. Some prefixes are standardized:

- = Function reference designation
- + Location reference designation
- - Product reference designation
- == Control structure reference designation
- & Document reference designation

Besides these predefined prefixes, you are free to use your own prefixes.

The prefix will be used to concatenate relative reference designations.

Features

The following features are available when you browse to enter the reference string:

- When you change the **Structure,** and the selected object exists also in the new structure, the object browser will navigate in the **Select Object** list box to the object which was selected in the previous structure.
  If this object does not exist in the new structure, it will start from the root object in the new structure.
- The selected object in the **Select Object** list box is highlighted.
- A default object identification in the **Object Identification** list box is selected, depending on the **Reference Type** and the available aspect types (see section).
- In the **Object Identification** list box all object names that have aspects of type “Basic Property Name” are inserted.
- When several object identifications exist in the list, the system selects the default with the following priority (in descending priority):
  - For **Reference Type** child/parent:
    a. Object identification with aspect of category name Relative Name, e.g. ((Relative Name))P3EU201

b. Object identification with Designation category name, e.g. ((Functional Designation))CP17

c. Object identification with aspect of category name Name, e.g. ((Name))EU201

– For **Reference Type** absolute:

a. Object identification with aspect of category name Name, e.g.((Name))EU201

b. Object identification with absolute reference designation (ARD) for an object in a certain structure, e.g. (ARD)=A3.M2

- In the **Property** list the following properties are added, in addition to the properties the aspect has published:
  - ASPECTDESCRIPTION and CREATIONTIME (always inserted)
  - OBJECTTYPE, only for an object name with an aspect of category name “Name”
  - ABS and ARD_DEFAULT, only for structure aspects, like e.g. Functional Structure.
    ARD_DEFAULT refers to the default ARD rule which is configured in the Plant Explorer. See “How to Select a Building Rule (ARD Rule)” above.
    ABS is used to get absolute reference designations without prefix and dot. This is used e.g. to do not offend against DB element naming.

User Interface Components

- Structure

  Drop down list which provides all available structures.
  Select a structure in which your object is located.

- Select Object

  List box containing the objects in the selected structure.
  Select the object for which you want to add a reference.

- Object Identification

  Drop down list box containing the object identification.

An object identification can be

– the name of an object with aspect of type “Basic Property Name”.
It consists of a category name and the name of the object, e.g.
((Name))EU201 for category name “Name” and object name “EU201”.

Example:
((Name))EU201 or ((Relative Name))P3EU201)

– a relative reference designation

Example:
((Functional Designation))B17

– an absolute reference designation for an object in a certain structure.

Example:
(ARD)=A1.B2.C2

NOTE: The list box is dimmed when it contains only one entry or the reference type is “same obj”.

- Aspect

  List box containing all the aspects of the selected object.

  Select the aspect for which you want to add a reference.

- Property

  List box containing all the properties of the selected aspect.
  A property holds descriptive data of an aspect.

  Select the property for which you want to add a reference.

- **Advanced** group

  Here you will see the **Reference Type** and **Reference String** the system has selected when you are browsing.

  You can also change the Reference String and Reference Type manually, when you want to insert a reference manually.

  The Advanced group may be compressed by default, as it is normally not needed to change the Reference String and Reference Type manually.
  In this case you can display it by clicking on the **Advanced** button.

- Reference Type

  The reference string is built up depending on the type of the reference, see section .
  You can see the reference type in the **Advanced** group.
  The reference type depends on the relation between the object where you started to browse, and your current object.
  It will be automatically set when browsing.

- Reference String

  The Reference String contains all information necessary to identify and access one single property in an Aspect System.
  Each property of all configured aspects can be accessed with this reference.
  You build the reference string by browsing. Alternatively you can also enter the string manually, if the object or aspect does not exist yet.

Reference String Syntax

When you insert a reference string manually the following syntax has to be used:

- Reference type “same obj”:
  .:<aspect name>:<property name>
  Example: .:Name:Description
- Reference type “child”:
  .(<structure name>)(down)((<category>))<object name>:<aspect name>:<property name>
  Example: .(Functional Structure)(down)((Name))EU201:Name:Description
- Reference type “parent”:
  .(<structure name>)(up)((<category>))<object name>:<aspect name>:<property name>
  Example:
  .(Functional Structure)(up)((Name))MixingUnit BV1:Name:Description
- Reference type “absolute Ref”:
  ((<category>))<object name>:<aspect name>:<property name>
  Example: ((Name))MilkReception:Name:Description

Advanced Reference Handling

Expert users can work with advanced reference handling to get e.g. the property value of a related object in an other structure.

Example:
You want to build a reusable library. In this library you want to get for the signal “AI Signal1” in the Functional Structure the location information of its AI board, “AI Board1”. The “AI Board1” is not available in the Functional Structure.
See Figure 320, Figure 321 and Figure 322.

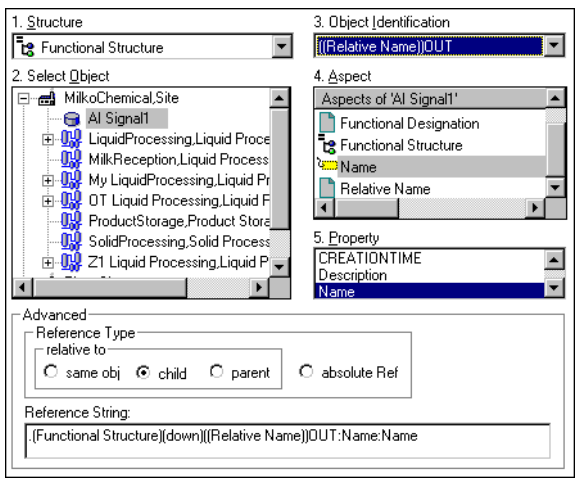


Figure 320. AI Signal1 in Functional Structure

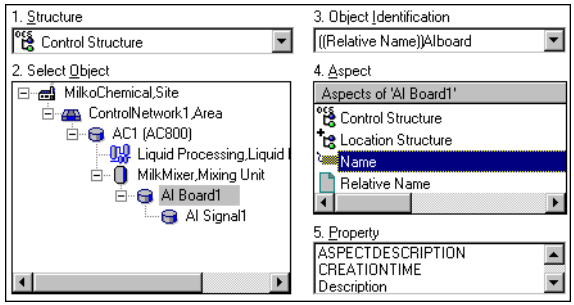


Figure 321. AI Signal1 in Control Structure

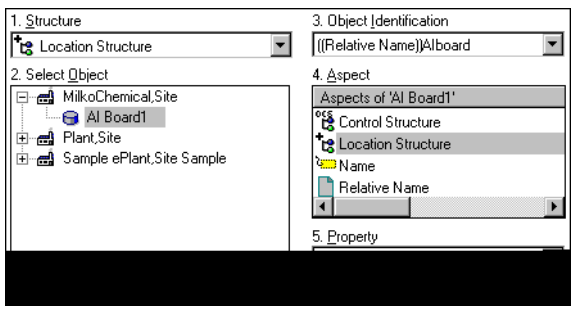


Figure 322. AI Board1 in Location Structure

You have to go from the “AI Signal1” in the Functional Structure to “AI Signal1” in the Control Structure. In the Control Structure you navigate to the parent, “AI Board1”.
Then you get the information of **Aspect** Location Structure and **Property** ARD_DEFAULT.

Use the relative name, because it will still exist after instantiating of the library.

How you build the reference string:
When browsing, you can only build the reference string within the Functional Structure. Therefore you enter the reference string manually in the Reference String edit box:

.(Functional Structure)(down)((Relative Name))OUT.(Control Structure)(up)((Relative Name))AIboard:Location Structure:ARD_DEFAULT

The resulting value is the location of AI Board1, e.g. Area1.Room12.Rack15.Pos3.

Appendix C Document Aspect Properties

The table below lists all administrative properties for document aspects stored in the Document Manager database in alphabetical order.

Table 41. Administrative Properties for Document Aspects

Property	Description	Type	Length
ApprDate	approval date	Date	
ApprDept	approval department	Text	30
ApprName	approval name	Text	30
BasedOn	based on	Text	50
CheckedInBy	system name of user who last checked in the document	Text	50
CheckedOut	indication if this document aspect is checked out	Bool	
CheckedOutBy	system name of user who last checked out the document	Text	50
CheckinDate	date of last check-in	Date	
CheckinDescription	user-defined checkin-description	Text	150
CheckOutDate	date of last check-out	Date	
CheckOutMachine	name of machine to which the document was checked out	Text	25
CheckOutPath	path to which the document was checked out	Text	300
ChkDate	check date	Date	
ChkDept	check department	Text	30

Table 41. Administrative Properties for Document Aspects (Continued)

Property	Description	Type	Length
ChkName	check name	Text	30
Comment	comment to document	Text	255
DefaultFile	default file for folder aspects	Text	100
DocDCCName	name part of document kind classification code	Text	30
DocDCCNo	number part of document kind classification code	Text	30
DocId	document identity	Text	30
ExternalFilePath	internally used for BDM Import	Text	400
FileFilter	not yet used	Text	150
FilePath	complete file path or only folder path	Text	300
FileReference	indication if this document aspect is a reference	Bool	
FreeDisp	free disposition, also called FreeDis	Text	50
Keyword	keywords to document	Text	100
Lang	document language	Text	10
LastEditDate	last edit date	Date	
LastEditName	last editor name	Text	30
NextSheet	number of next sheet (for drawings)	Text	10
NoOfSheets	number of sheets	Text	10
Owner	owner of document	Text	50
PrepDate	preparation date	Date	

Table 41. Administrative Properties for Document Aspects (Continued)

Property	Description	Type	Length
PrepDept	preparation department	Text	30
PrepName	preparation name	Text	30
Replaces	replaces	Text	50
ReplBy	replaced by	Text	50
RevisionIndex	revision index	Text	10
SheetNo	number of current sheet (fro drawings)	Text	10
Standard	document standard	Text	30
StartDate	start date of project	Text	30
Status	status of document	Text	30
Subject	subject of document	Text	100
Title1..Title4	document titles	Text	100

Appendix D Parameter Categories Examples

This appendix supplies some examples to understand the different kinds of Parameter Aspect Categories.

It shows examples for expressions used in Parameter Aspect Category Definition.

Example 1: Table-like Categories

StationParameter

A table-like Parameter Aspect Category named 'StationParameter' with the properties ObjectName, AspectName, Bus, Station, Type

can be displayed like a one-dimensional table:

Object Name	AspectName	Bus	Station	Type
Part1.3	StationParameter3			
Part1.2	StationParameter2	0	0	
Part1.1	StationParameter1	0	0	
Part2	StationParameter20			

Figure 323. Example of a Table-like Category

Example 2: Structured Categories

In some cases a category needs to be defined in a way to store structured information.

View a Structured Category

In the following example two types of views can be used to present the structured category:

Figure 324. Views of structured categories

How to work with structured categories in Bulk Data Manager see Executive Summary.

Create a Structured Category

Lets assume you want to produce different paint materials. To get the ordered tone and quantity of the paint you have to mixture two or more components in the true ratio of components. The number of components vary from product to product to be produced. In our example we create a Parameter Category where you might enter common product data like, article number, product id and the quantity, and based on a sub category called components where the kind of components and the number of components to produce the required paint product is outlined.

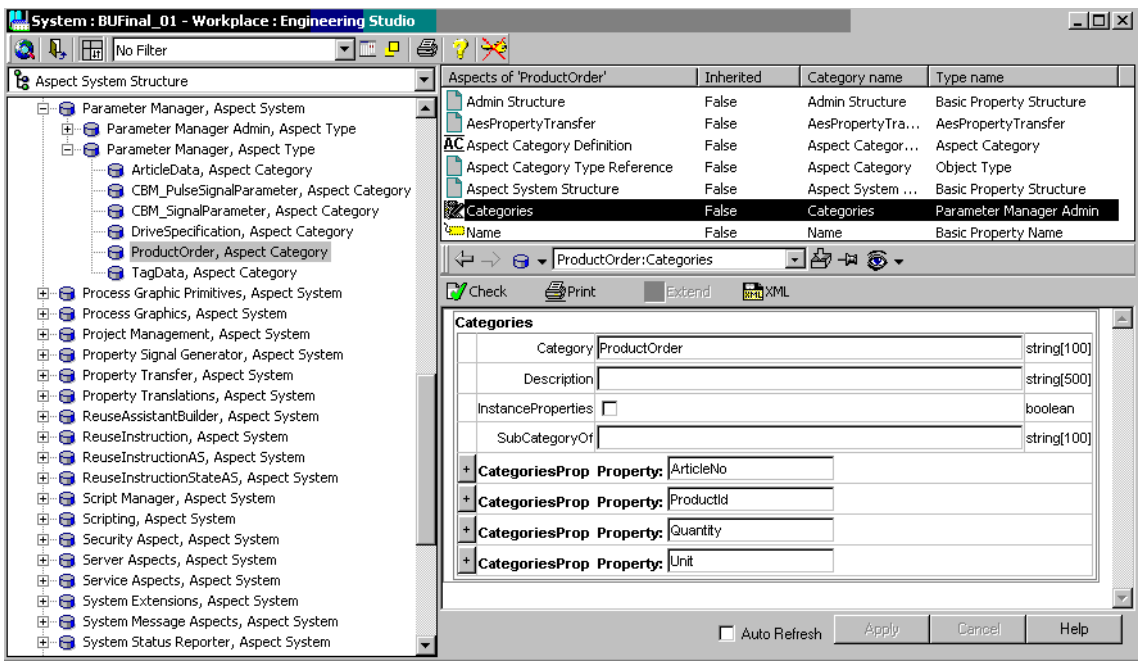


Figure 325. Category ProductOrder as Top Category

The structured category “ProductOrder” is used and defined in the following way:

- The top category is called ProductOrder, with few simple properties like “ProductId”, “ArticleNo”, “Quantity” and “Unit” (see Figure 325).
- There is one sub category called Components.

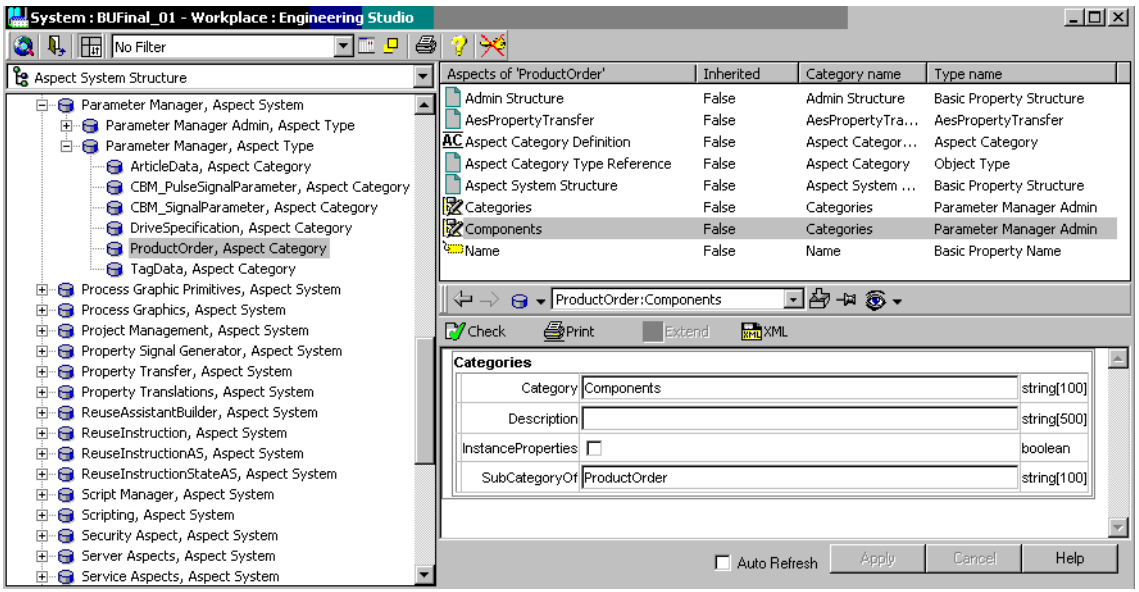


Figure 326. Category Components, Sub Category of ProductOrder

- The sub category Components keeps the properties: “ComponentId”, “Supplier”, “Quantity” and “Unit” (see Figure 327).

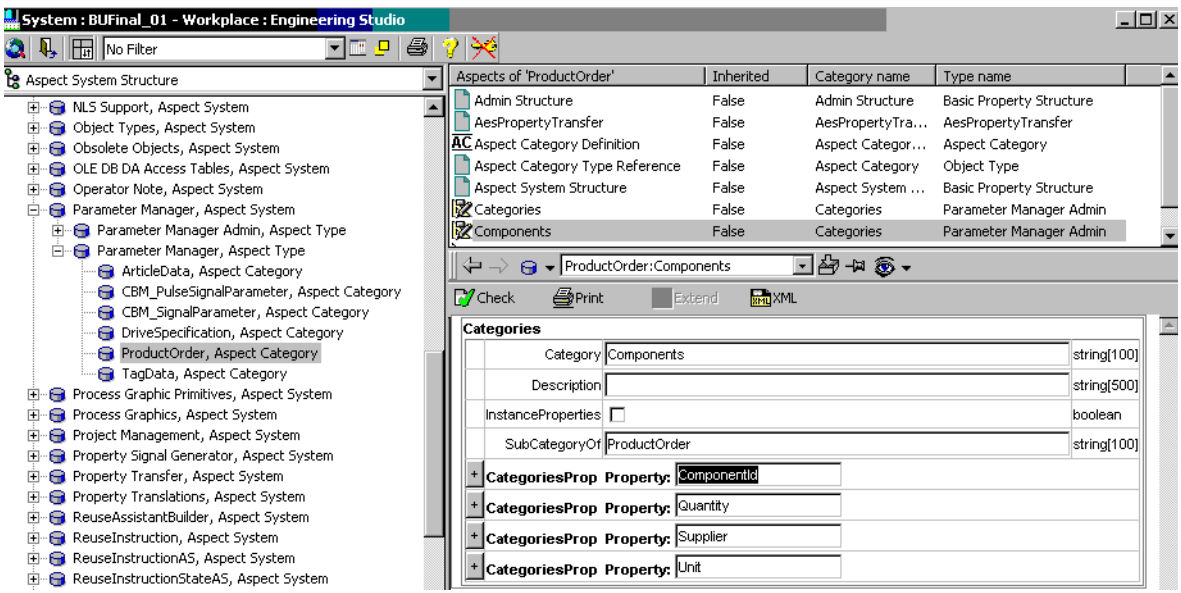


Figure 327. Properties of the Sub Category Components

- The property “ComponentId” has a special meaning. It is used as row identifier to identify the perhaps available several instances of sub category “Components”.

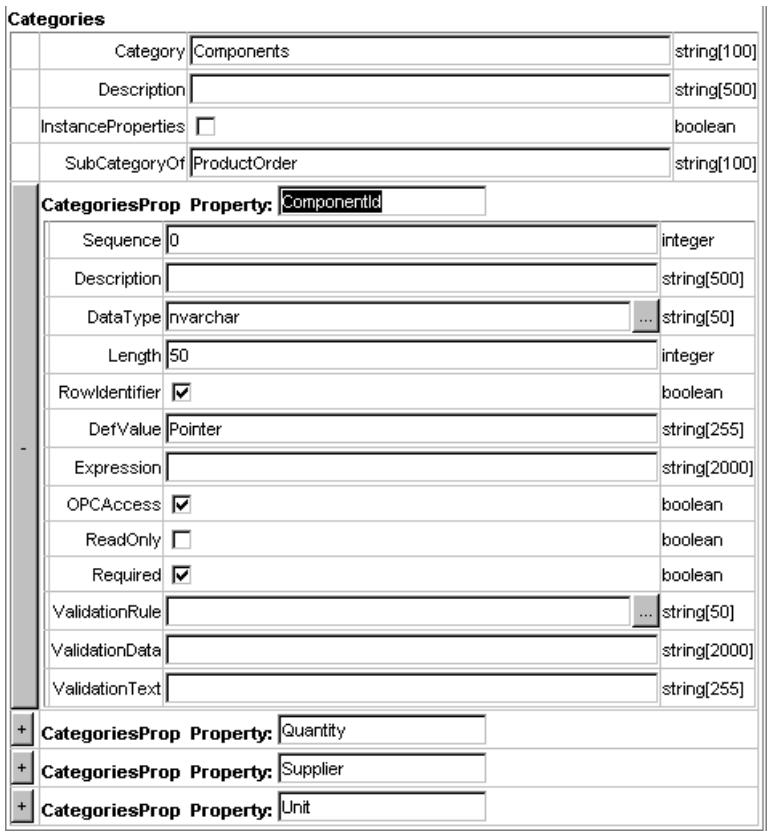


Figure 328. Property ComponentId used as Row Identifier

- Figure 328, shows the properties of the ComponentId property. You just check the RowIdentifier check box and enter any value as DefValue (in our example value *Pointer* was inserted).
- If you create an Parameter Aspect of our just created ProductOrder category you get a Aspect which only holds properties of the top category (see

Figure 329).

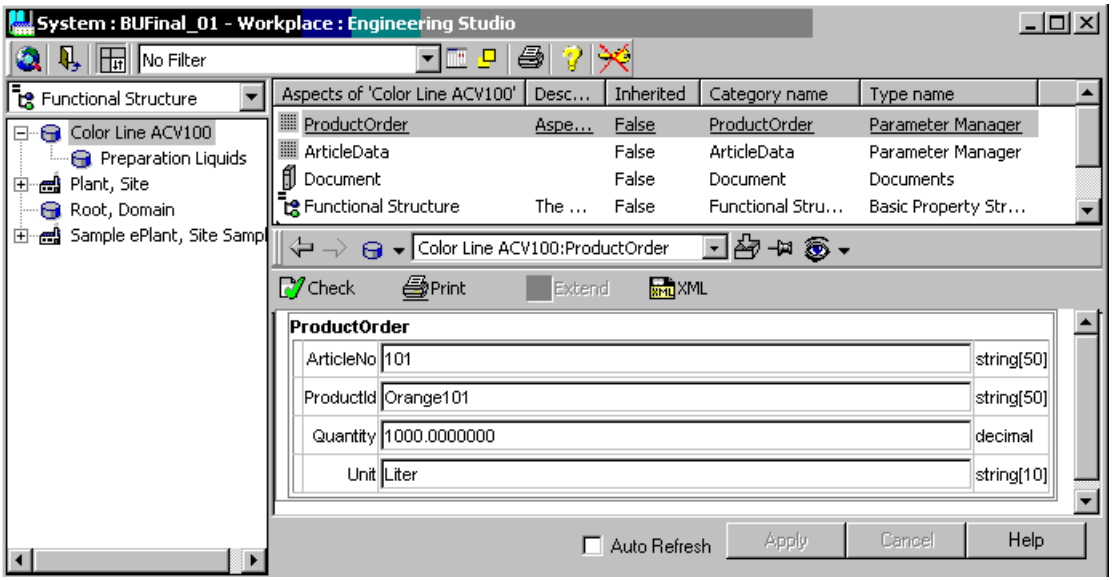


Figure 329. Parameter Aspect Instance of ProductOrder Category

- The order might be, produce a paint which has the tone Orange101 with the ArticleNo 101 and a Quantity of 1000 liter. To get such a tone you need two components. Add the Components instances and enter the data (see

Figure 330).

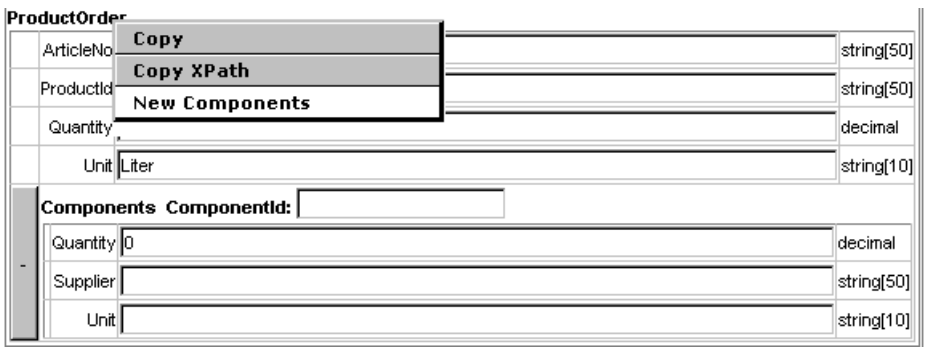


Figure 330. Add new Components Instance

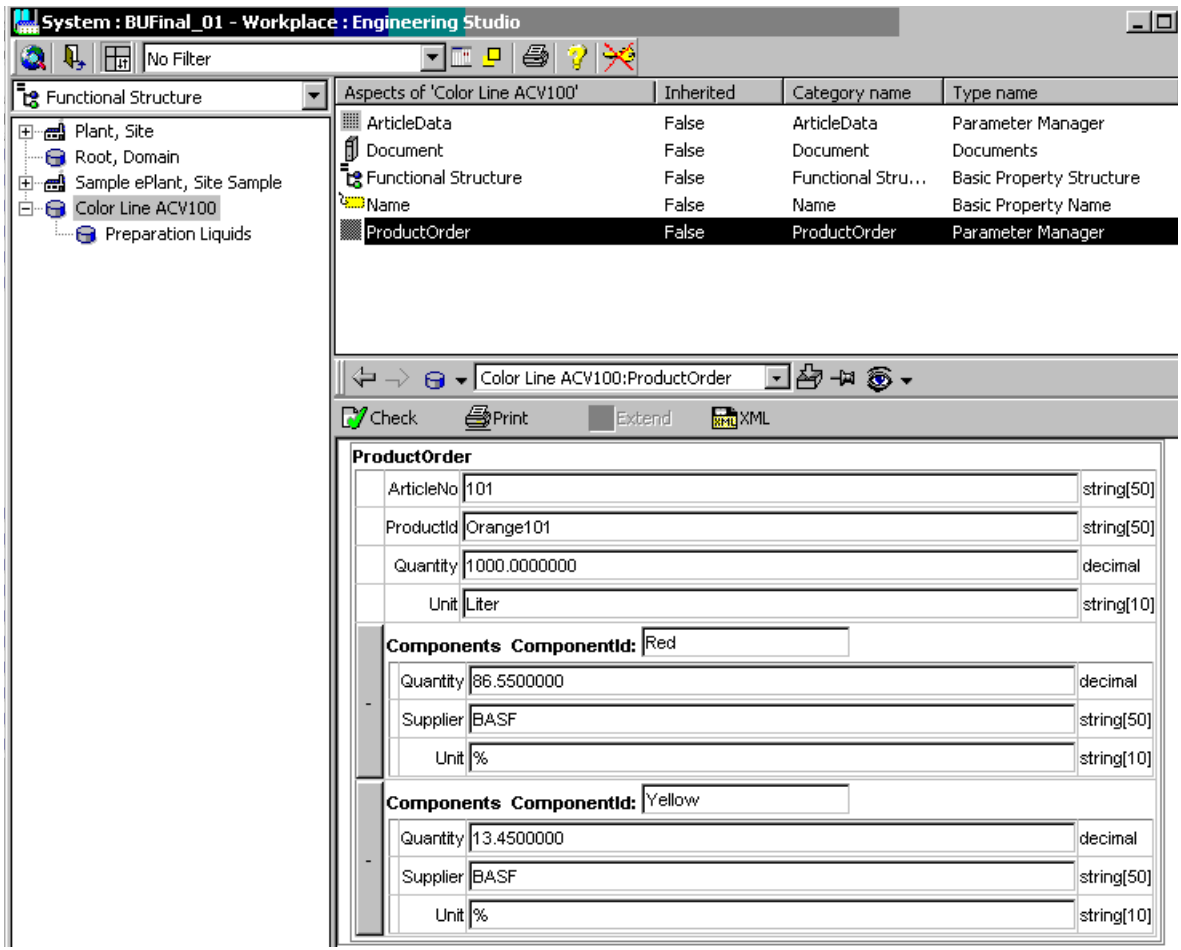


Figure 331. ProductOrder Aspect with two Components

- A ProductOrder aspect describes in this scenario which color is mixed in which proportion in order to produce a certain painting (Orange101 with the components Red and Yellow). Depending on the mixture, two ore more components need to be mixed for a certain painting. Therefore two or more

components can be added to a Product Order. For example to produce “Orange050” the colors “Red”, “Yellow” and “White” needs to be mixed.

Example 3: Extendable Categories

Documents

An extendable category has instance specific properties. For each different instance you can provide a maybe different set of (Property, Value).

In this example, the category Documents has the properties ‘DocumentId’, ‘DocumentName’ and ‘PreparedBy’ which are defined for all instances. Additional instance specific properties are also defined for every instance:

‘**Document1**’ has the instance specific properties ‘ApprovedBy’, ‘InternalCode’ added.

‘**Document2**’ has instance specific property ’InternalCode’ added.

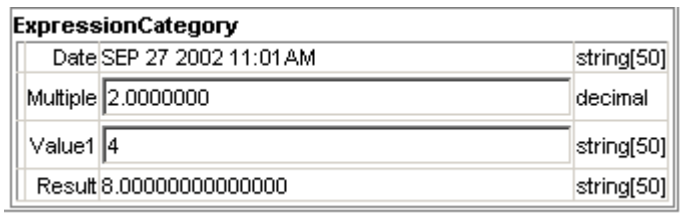
‘**FD1.1**’, ‘**FD1.2**’, ‘**FD1.3**’ have ‘Language1’, ‘Language2’ added.

Object Name	AspectName	DocumentId	DocumentName	PreparedBy	ApprovedBy	Lanugage1	Language2	InternalCode
Part1	Document1	10	FD	Mr. X	Mr. Z			10XY77
Part1.3	FD1.3	122	Rel Note	Miss ABC		english	deutsch	
Part1.2	FD1.2	123	Rel Note	Miss ABC		english	deutsch	
Part1.1	FD1.1	124	Rel Note	Miss ABC		english	deutsch	
Part2	Document2	20	UserManual	Mr.Y				10XY78

Figure 332. Example of an Extendable Category

Examples for Expressions

Some examples for expressions used for properties in Parameter Aspect Category Definition are given (see Features and Restrictions of Category-Properties).



ExpressionCategory		
Date	SEP 27 2002 11:01 AM	string[50]
Multiple	2.0000000	decimal
Value1	4	string[50]
Result	8.00000000000000	string[50]

Figure 333. Example for Expression GETDATE and Value multiplication

What you have to do to get the current date:

- Create and configure a Parameter aspect category as described below
- Create/ Add an Parameter aspect of your just created category on the selected Aspect Object
- After aspect creation the date and time will be presented (see Figure 333, Date property).

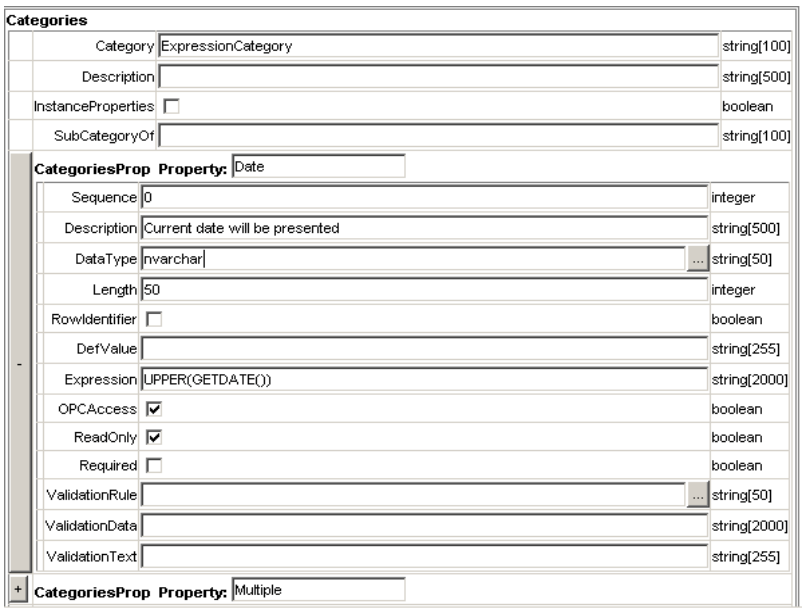


Figure 334. Expression property keeps the GETDATE() functions.

Figure 334 shows the configuration of the property Date to get the current date and time.

Note: For same reasons you have to enclose GETDATE by the UPPER() function to achieve the proper date (see Figure 334).

What you have to do to use multiplication function in your “ExpressionCategory” category:

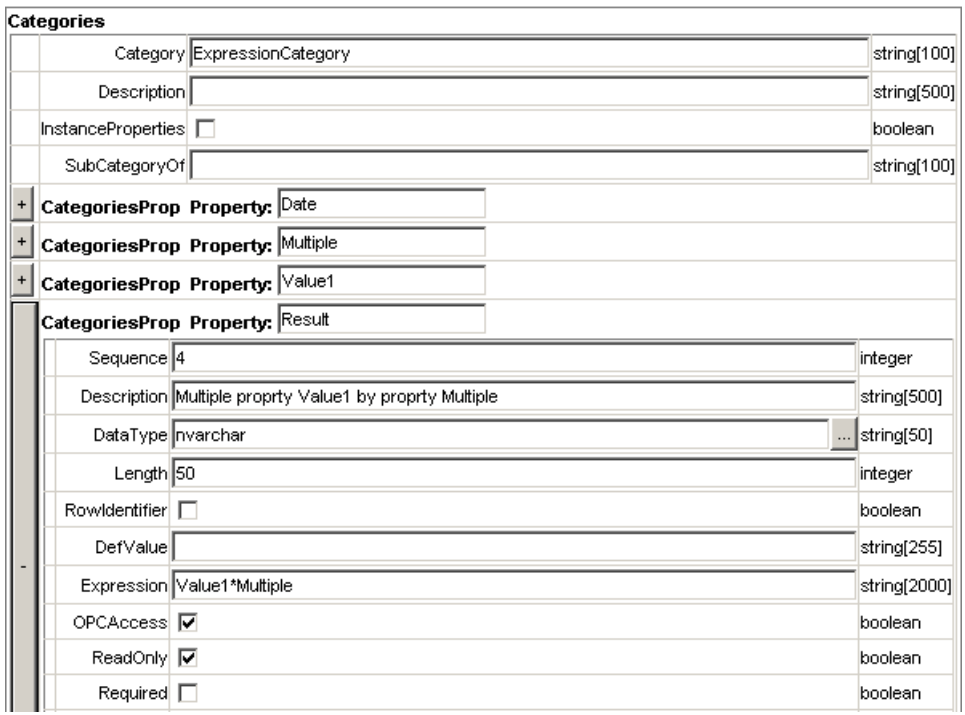


Figure 335. Expression property contain the property Value1 multiplied (*) by property Multiple

The property Result hold the result of the multiplication of the property Value1 and property Multiple as multiplier (see Figure 335); by the way properties which contain expressions are set to read-only.

What you have to do to fetch property data from other aspects in your “FetchData” category:

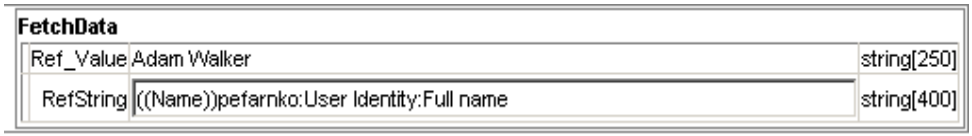


Figure 336. Example: Property Ref_Value contains the data of an Aspect Object

An example, the User Structure presented in the Plant Explorer contains all the users attached to the current project. An Aspect Object named *pefarnkop* - one of such project users - based on the Object Type User and holds a General Properties which is called User Identity. Here are the personal data of the user stored; for example: address, phone numbers, full name and so on. In the example above the Full Name was fetched and inserted into the property Ref_Value (see Figure 336).

What you have to do to use the reference mechanism:

- Create and configure a Parameter aspect category as described below
- Create/ Add an Parameter aspect of your created category on the selected Aspect Object
- Insert into the property RefString your appropriated reference string.

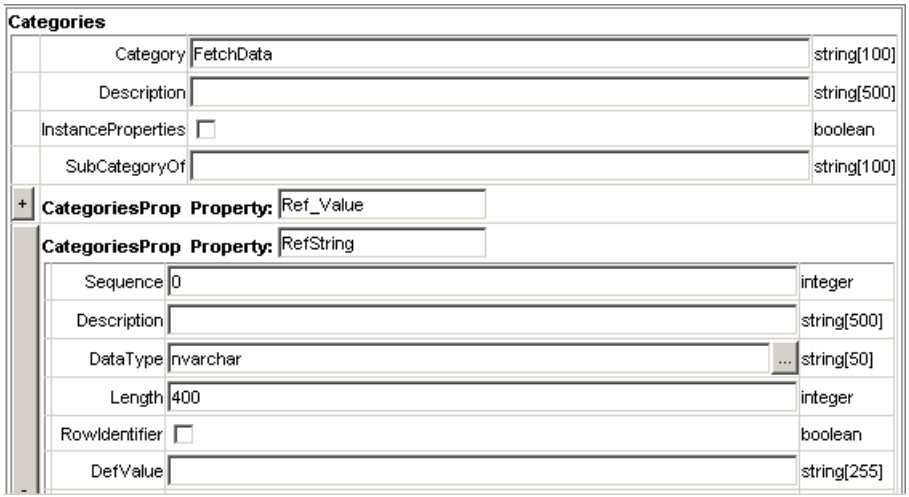


Figure 337. Example, property RefString where the Reference string are stored

There are two properties necessary. Property Ref_Value of data type nvarchar which keeps the fetched data and the property RefString which gets the reference string to identify the wanted data property.

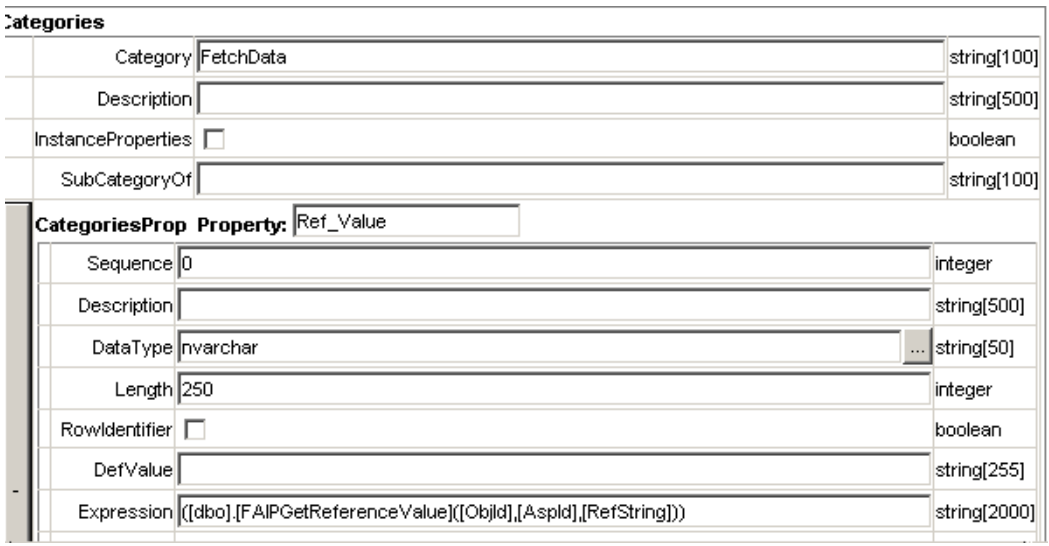


Figure 338. Example, property Ref_Value where the fetch data are presented and the reference function are stored

The function which execute the fetch condition has to be stored in the Expression property: (**[dbo].[FAIPGetReferenceValue]([ObjId],[AspId],[RefString])**). Of course, your are free to define your own property names in that case exchange RefString by your one. The syntax of the reference string which identify your wanted property follows the rules which are described in the Property Reference Dialog (for example, see Appendix B, Property References).

Additional Expressions Functions:

- Prop1/2: The value of the property Prop1 will be divided by 2.
  Note: Usable mathematical operators are: addition +, subtraction -, division /, multiplication *.
- DATENAME(weekday,GETDATE()): The value of the property will be the name of the actual weekday.

- STUFF(Prop1,2,3,'xyz') If Prop1='a123bc' then the value of the property will be 'axyzbc'. Remark: The STUFF function deletes a specified length of characters and inserts another set of characters at a specified starting point. Syntax: STUFF(string,start,length,string).
- DATEDIFF(<intervall definition>, date1, date2), interval definitions are: year, month, week, day. Example: (DATEDIFF(day, OrderDate, DeliveryDate), the properties OrderDay and DeliveryDate have to be from data type datetime. The property which holds this expressions yields the number of days between order date and delivery date.

Appendix E Word Report Templates

Principles within MS Word documents

The following section describes the principles within MS Word documents.

Use of Template Documents

The documents to be created are regarded just as reports based on

- Data stored in Excel
- A corresponding report template available as .doc file

The report template file (.doc) must contain following definitions

- Header/Footer definitions
  The information to be updated within header/footer cells must be defined as Custom Properties within MS Word
- Table definitions
  a report template can contain several Word tables. Each table can be mapped to a separate
  - Sheet
    or
  - Named Area

  within an Excel Workbook. A table can consist of
  - 0..n header rows
    Header rows must be marked within MS Word via menu item **Tables > Heading Rows Repeat**
  - 1..n data rows
    Only those rows representing one record from data source must be included in template. So one Excel row may be mapped to several Word table rows.

Mapping Principles Between Word and Excel

File Mapping

It is assumed, that all data to be mapped into one Word document are included within one Excel Workbook. So basically it is not intended to merge table data from different workbooks into one Word document.

The standard internal behavior when running the report functionality is:

1. The template .doc file is copied to the document .doc file on same folder with same name as .xls file
   Example:
   To generate a Word document from
   `C:\mydocuments\mydata.xls`
   the Word template is copied to
   `C:\mydocuments\mydata.doc`
2. The report function running on the `.xls` workbook opens the `.doc` file according the previous naming conventions.
   If the `.xls` workbook is registered as *Document* aspect to an Aspect Object, the existing property references are updated now via Document Manager.
3. The tables within .doc file are populated now with data from .xls workbook.

Table Mapping

Each table within Word document can be mapped to:

- A sheet within referenced Excel-Workbook
  In this case it is assumed, that within first row of the Excel sheet the column names are defined
- A “Named Area” within referenced Excel-Workbook

To identify the sheet/named area within the corresponding Word table a bookmark must be placed according to naming conventions:

- `Report_<sheetname>`
  Example: bookmark `Report_IO_LIST` references to sheet `IO_LIST`

- `Report_<areaname>_NA`
  Example: bookmark `Report_BOARDS_NA` references to named area `BOARDS`

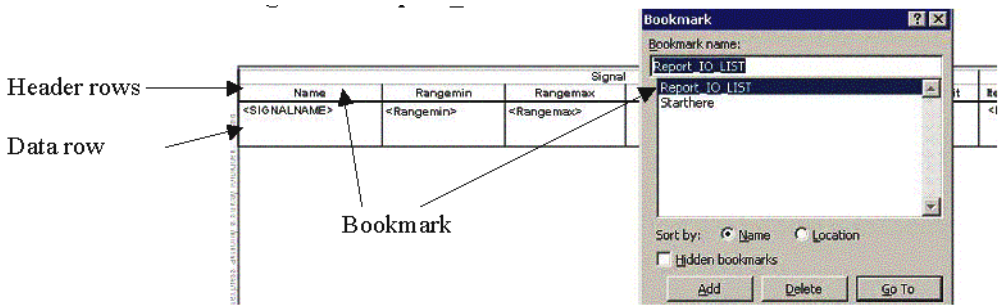


Figure 339. Marking a Word Table as Report Table

Note: names of sheets or named areas must not contain any spaces or special characters, because a bookmark does not allow this (only _ allowed)!!!

Field Mapping

Columns within Excel sheet (or named area) must be mapped to columns within Word tables.

Within Excel for this purpose the first row of a sheet (or a named area) must contain the field (=column) names

Within Word table the field name must just be included in **<..>**
(Example: **<Signal_Name>**)

Following field naming exception rules must be considered:

- The field names within Word must not contain any spaces

  Note: if the column name within Excel includes any spaces, the driver automatically uses "_" instead
  Example: Column **Signal Name** within Excel must be inserted as **<Signal_Name>** within Word

- The field names within Word must not contain any dots

  Note: if the column name within Excel includes any dots, the driver just removes those
  Example: Column **`Name.Name`** within Excel must be inserted as **`<NameName>`** within Word

Header/Footer Data Mapping

There are 2 variants, how to transfer header/footer data to corresponding *Custom Properties* within Word

Variant A: (only available within Engineering Workplace)
Just the Document Manager function is used. In this case must be ensured that

- the custom properties are named according to parameter referencing conventions and/or Document Manager conventions
- before startup of the Word file the GUIDs are set correctly within file
  Note: is done automatically by Excel AddIn

Variant B: (available also independent from Engineering Workplace)
Within Excel workbook a sheet **Header** exists. In this case must be ensured that

- within first row of **Header** sheet the column names are defined according to corresponding Custom Properties names with Word file
- the header data to be transferred to Word are included within second row

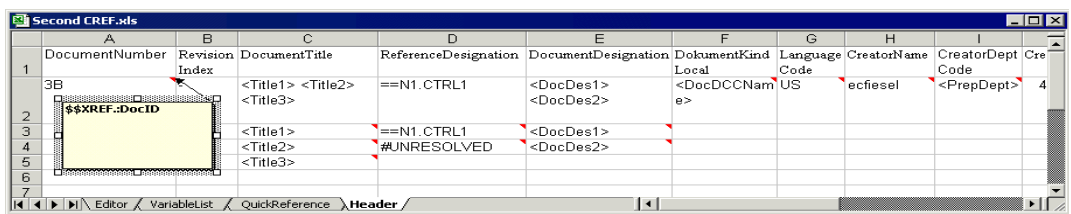


Figure 340. Header/Footer data mapping

If you use property references to update header data into Excel sheet, you can implement a mapping of any aspect properties to custom properties in Word templates, for example as provided from TTT (an ABB internal set of templates).

Special Word Report Functions

Using Expressions within Select

It is possible to add expressions to Word tables, which are considered within data selection from Excel. For proper understanding you must know, that a user defined expression is just added to the select statement in the way

SELECT <expression 1 ,>... <expression n ,> * FROM <table>

Each expression string must not exceed 255 characters. In summery all strings together must not exceed around 450 characters (complete select statement limited to 510 characters)

Within expressions you can use standard *Visual Basic* functions, like `LEFT, RIGHT, MID` or `REPLACE` which may be used for string formatting. Or you can use even `IIF,` which may be helpful for more logical operations.

An expression string is just placed somewhere into the data row and must be encapsulated by "{" <*expression*> "}"

Following field naming rules within expressions must be considered:

- If field names include special characters or spaces, it must be encapsulated with "[ ]"
  Example: **`Signal Name`** within Excel must be inserted as **`[Signal Name]`** within expression
- If field names include dots, the dot must be replaced by #
  Example: **`Name.Name`** within Excel must be inserted as **`[Name#Name]`** within expression
- If you insert an expression like **{**<*`fieldname`*> **`as`** <*`synonym`*>**}**, you can't use the fieldname any longer as merge field
  Example: if **{ `[Signal Name] as SN`}** is used as expression, the merge

field **<Signal_Name>** will not work any longer (**<SN>** must be used now instead)

Index	Object Name
<Ind>	<Label><NameName> {IIF (len([Name#Name]) > 0 , 'Name: ' , '') as Label }

Figure 341. Example for Creating Label Field via Expression

You must make sure, that you really use ' ' to encapsulate text strings. If you try to edit this in Word directly, Word uses ‘ ’ as default. So it is recommended to use Notepad to edit the expression and copy it into Word.

Using Field Formatting Functions

Because merging Excel data to Word tables is based on standard MS Word Mail Merge function, the same formatting mechanisms as provided on Mail Merge fields can be used. For details you can look into MS Word Help topic *Format merged data*.

The only difference is, that you type in the “formatting switch” directly into the field definition marked by “<...>”.

Application example:
When merging integer values from Excel to Word, the integer values are automatically converted to float (caused by ODBC driver behavior).

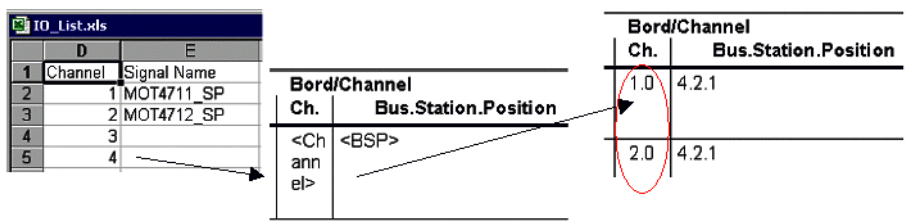


Figure 342. Wrong default formatting of Integer during Merge

To avoid this, you can add the formatting switch **\# "0"** to the field definition.

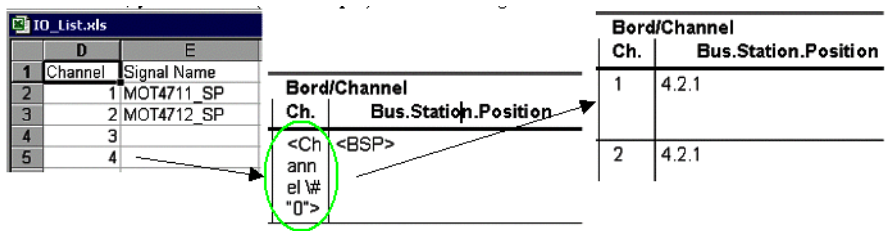


Figure 343. Use of formatting Switch within field definition

Here a short MS Word Help extract about the supported formatting switches:

Switch type	Description
Format (*)	Specifies number formats, capitalization, and character formatting; prevents changes to the format of the field results when a field is updated.
Numeric Picture (\#)	Specifies the display of a numeric result, including the number of decimal places and the use of currency symbols.
Date-Time Picture (\@)	Sets the format for fields that have a date or time result. For example, 12/16/96, Friday 12:35 p.m.

Figure 344. Extract from MS Word Help about Formatting Switches

Using Table Formatting Functions

There are some extra table formatting functions available within report generation.

The formatting functions are executed on certain formatting strings:

- `[PageBreak]`
  creates a page break on current row (except of first row in table)
- `[InsRow]`
  Inserts an empty row before current row (except of first row in table)
- `[DelRow]`
  deletes current row

- `[ClearCell]`
  clears current cell

The formatting strings can be created dynamically by using expressions.

Example:
It is assumed, that the data source for an I/O list includes a column **NextBoard**, which provides TRUE, if a new board starts within data source. So you can force a page break within the Word table according to following example

	Bord/Channel		
it	Item.designation	Ch.	Bus.Station.Position
	<Board_ID> <PB> {IIF(NextBoard, '[PageBreak]', '') as PB}	<Channel>	<BSP>

Figure 345. Forcing Page Break within Word Table

Appendix F Reuse Assistant Architect - Reference

This subsection lists all operations that are available in the Architect. The operations can be used with aspects of the following categories:

- Reuse Answer Operations
- Reuse Pre Operations
- Reuse Post Operations

Each of these aspects shows the same aspect page.

General Aspect Object Operations

New Object

This operation creates a new aspect object. The operation presents the following dialog.

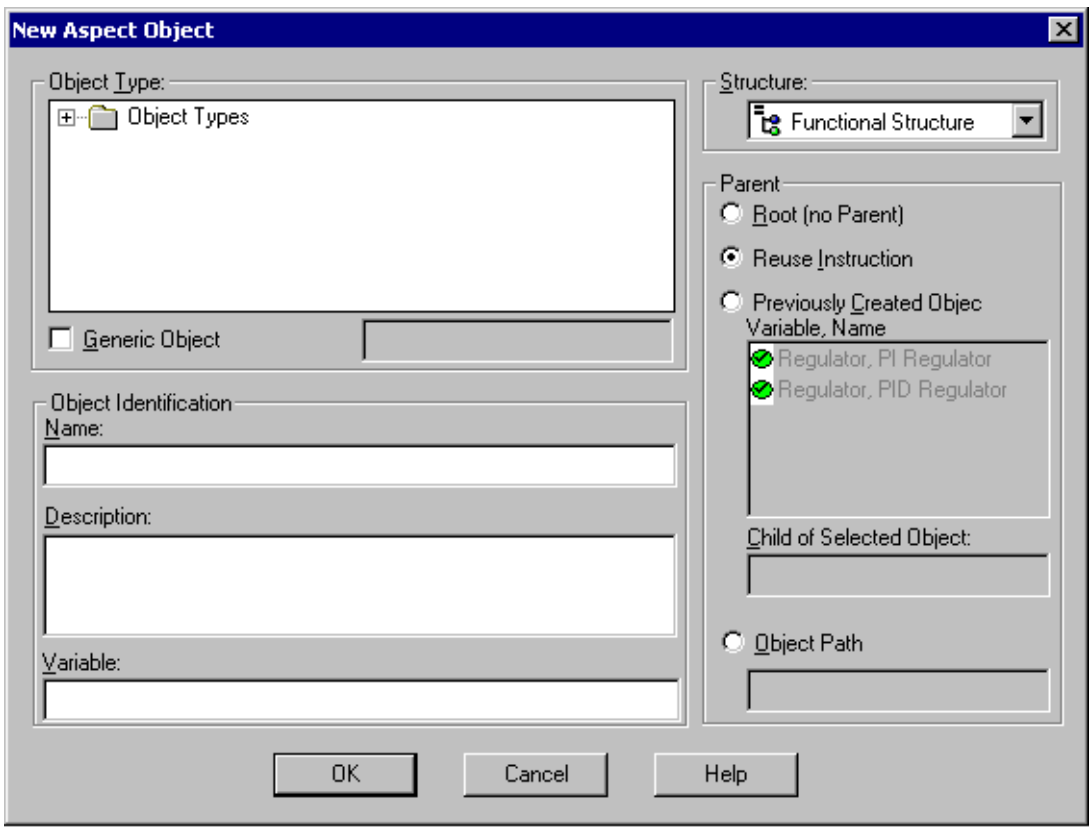


Figure 346. New Object Operation Dialog

Table 42. New Object Operation

Section	Field	Description
Object Type	Object Browser	Select the type of the object that you want to create. To create a Generic object, set the check box Generic.
Object Type	Generic Object	Check to create a Generic aspect object. If checked, the selection in the object type browser not relevant
Object Identification	Name	Name (Name.Name) of the aspect object.
Object Identification	Description	Description (Name.Description) of the aspect object.
Object Identification	Variable	Name of the variable that is used to reference this object from other operations of this instruction. Allowed Values: A...Z; a...z, 0...9, _ The variable name must start with a letter.
Structure	Structure Selector	Select the structure in which the aspect object is created. Please note: This is always valid, even is you specify a structure with the search path to the parent object!
Parent	Root	The new aspect object is created as a root object (no parent) in the selected structure.
Parent	Reuse Instruction	The new object is created as a child of the object with the Reuse Instruction aspect.
Parent	Previously Created Object	The new object is created as a child of an object that is also created by this Reuse Instruction. Select the variable of this object from the list below.

Table 42. New Object Operation (Continued)

Section	Field	Description
Parent	Child of Selected Object	The new object is created as a child of a child of an object that is created by this Reuse Instruction. Enter the name (Name.Name) of the child object. This selection.
Parent	Object Path	Enter the search path to the parent object. This path may include a structure. Please note that this structure may be different from the structure in which the object is created. If the object which you are addressing contains a character that is used as separator (like "."), you need to escape this character by **two** "\" characters. Example: For the object path "Test/AI1.1" you must enter: Test/AI1\\.1

New Aspect

This operation creates a new aspect. The operation displays the following user interface.

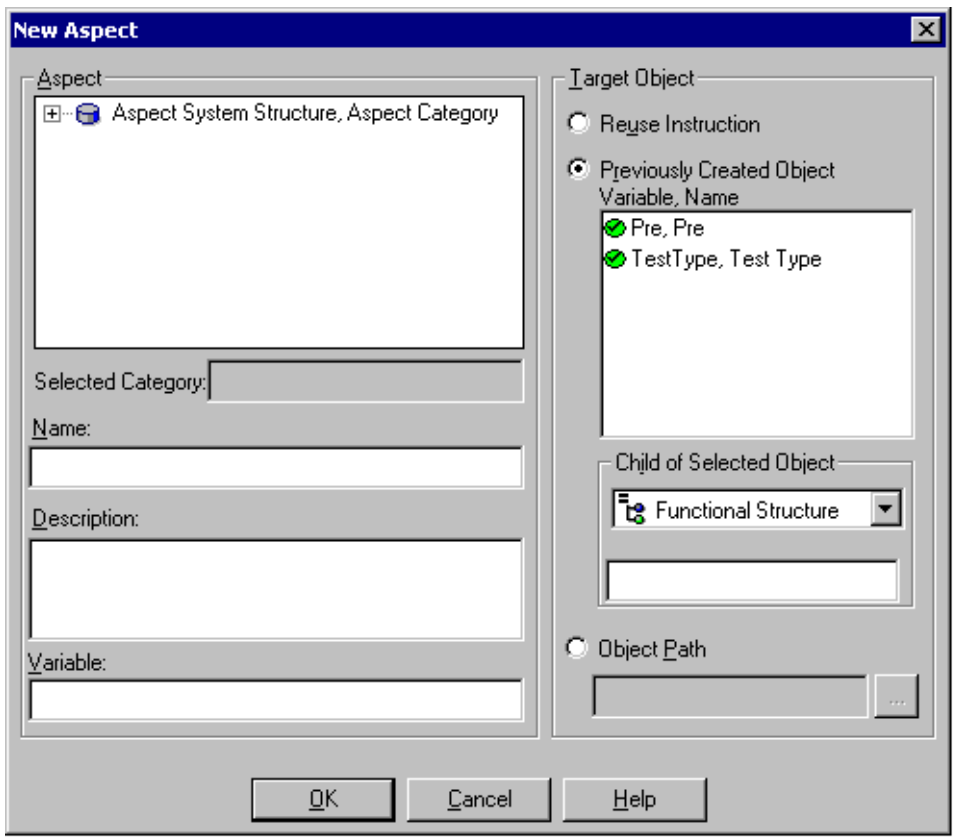


Figure 347. New Aspect Dialog

Table 43. New Aspect Operation

Section	Field	Description
Aspect	Aspect Category Browser	Select the category of the new aspect.
Aspect	Selected Category	The field is read only. It displays the selected aspect category.
Aspect	Name	Name of the new aspect.
Aspect	Description	Description of the new aspect.
Aspect	Variable	Name of the variable that is used to reference this object from other operations of this instruction. See Working with References for details. Allowed Values: A...Z; a...z, 0...9, _ The variable name must start with a letter.
Target Object	Reuse Instruction	If this radio button is selected, the aspect will be created on the object with the Reuse Instruction (the object where you start the Builder).
Target Object	Previously Created Object	If this radio button is selected, the new aspect will be created at an object that is created by the Reuse Instruction. Select the object from the list. The reference is build using the variable name of the object. See Working with References for details about references.

Table 43. New Aspect Operation (Continued)

Section	Field	Description
Target Object > Child of Selected Object	Structure Selector	This selection is relevant if the radio button Previously Created Object is selected. In this case, you can create an aspect at child object of the selected object. The structure selector defines, in which structure this child is located.
Target Object > Child of Selected Object	Edit Field	Enter the name (Name.Name) of the child object.
Target Object	Object Path	If this radio button is selected, the target object is identified via a path. Enter the path into the edit field. Example: [Functional Structure]Root/PI Regulator If the object which you are addressing contains a character that is used as separator (like "."), you need to escape this character by **two** "\" characters. Example: For the object path "Test/AI1.1" you must enter: Test/AI1\\.1

Modify Property Start Dialog

This operation modifies the value of an OPC property. The operation shows the following user interface.

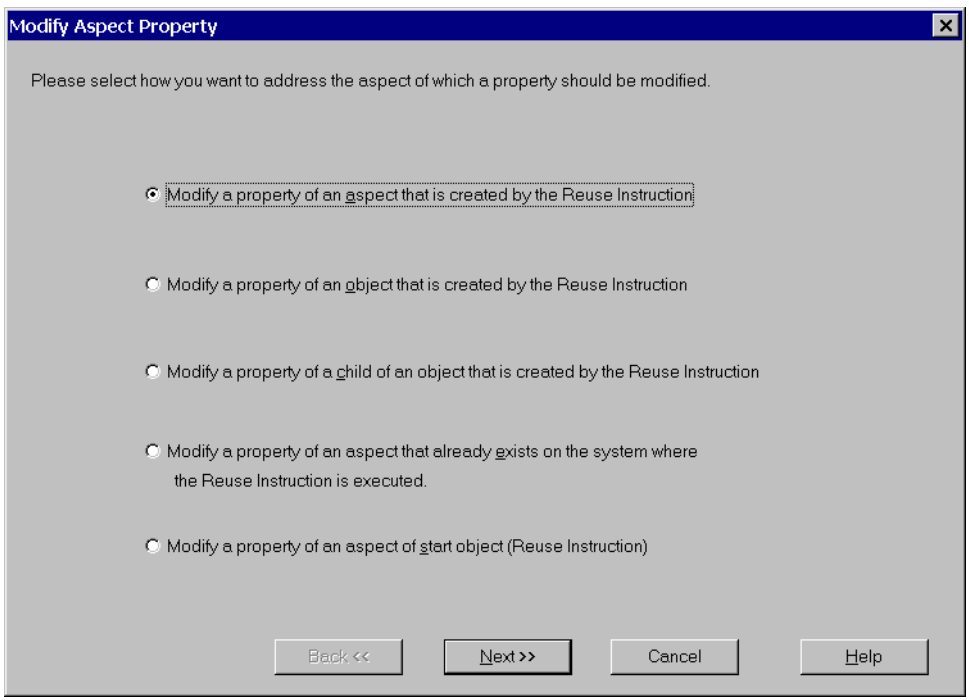


Figure 348. Modify Property Selector

This dialog serves as a startup for the following property addressing dialogs. For details see:

Modify Property of Previously Created Aspect

Modify Property of Previously Created Object

Modify Property of Child of Previously Created Object

Modify Property of Existing Object

Modify Property of Object with the Reuse Instruction

Modify Property of Previously Created Aspect

This operation lets you modify a property of an aspect that is created as part of the Reuse Instruction.

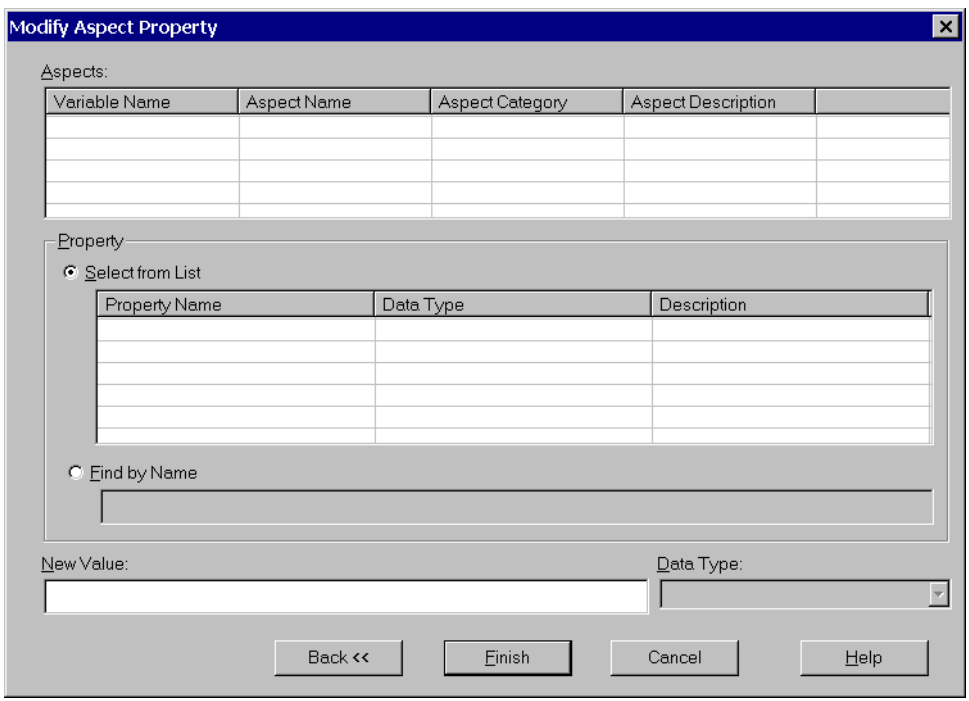


Figure 349. Modify Property

Table 44. Modify Property Operation

Section	Field	Description
	Aspect	Select the aspect that has the property that you want to modify. This list displays all available aspects that are created by this Reuse Instruction. The variable name is used to reference the aspect.
Property	Select from List	If this radio button is selected you may select the property you want to modify from the list below. The property name is used to address the property.
Property	Find by Name	If this radio button is selected you can enter the name of the property into the edit field below. The name is not case sensitive. You also need to define the data type of the property in the Data Type field.
	New Value	Enter the value new value for the property. You can use substitutions. You can also use script expressions. Examples: $FunctionName$_Test Ti* 3.14 SIN(TI) + COS(Kr) MyFunction(Kr) + 1.2
	Data Type	Select the data type of the property. The selection is automatically done for you if you select a property from the property list.

Modify Property of Previously Created Object

This operation lets you modify a property of an aspect of an object that is created by this Reuse Instruction.

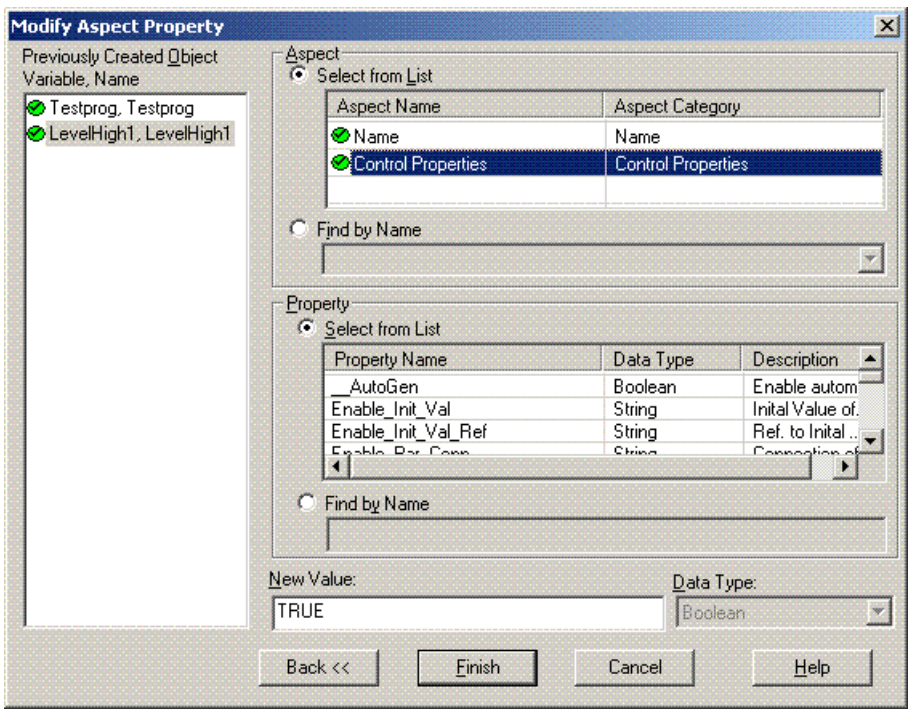


Figure 350. Modify Property

Table 45. Modify Property Operation

Section	Field	Description
	Previously Create Object	Select the object that has the aspect that has the property you want to modify. The variable name is used to reference the object. See Working with References for details.
Aspect	Select from List	If this radio button is selected you may select the aspect that has the property you want to modify. The aspect name is used to find the aspect.
Aspect	Find by Name	If this radio button is selected you can enter the name of the aspect into the edit field below. The name is not case sensitive. The aspect name is used to find the aspect.
Property	Select from List	If this radio button is selected you may select the property you want to modify from the list below. The property name is used to address the property.
Property	Find by Name	If this radio button is selected you can enter the name of the property into the edit field below. The name is not case sensitive. You also need to define the data type of the property in the Data Type field.

Table 45. Modify Property Operation (Continued)

Section	Field	Description
	New Value	Enter the value new value for the property. You can use substitutions. You can also use script expressions. Examples: $FunctionName$_Test Ti* 3.14 SIN(TI) + COS(Kr) MyFunction(Kr) + 1.2
	Data Type	Select the data type of the property. The selection is automatically done for you if you select a property from the property list.

Modify Property of Child of Previously Created Object

This operation lets you modify a property of an aspect that belongs to a child of an object that is created by the Reuse Instruction.

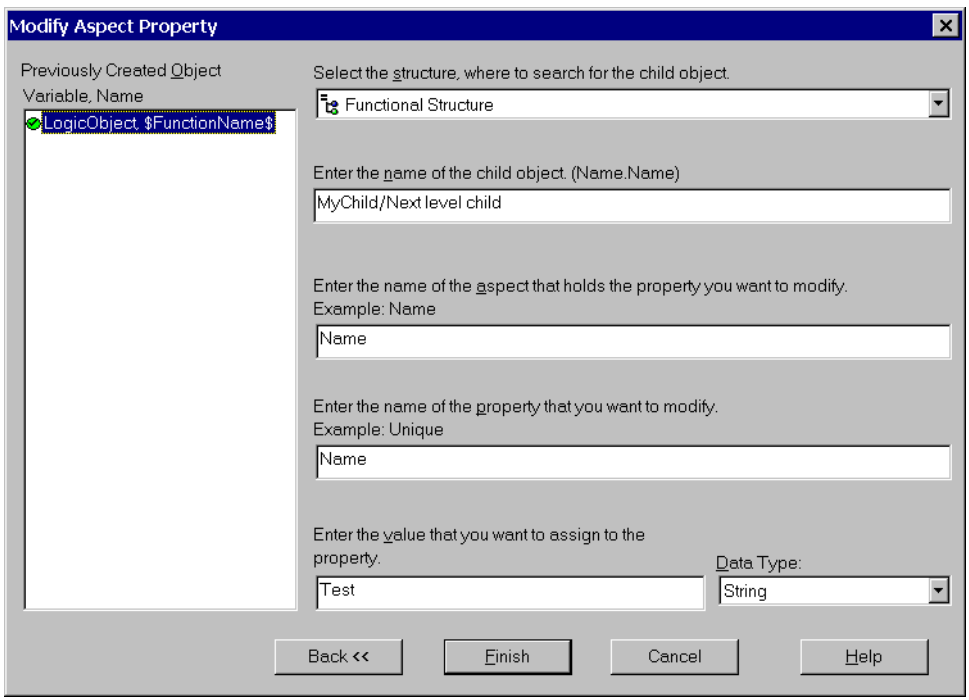


Figure 351. Modify Property Operation

Table 46. Modify Property Operation

Section	Field	Description
	Previously Created Object	Select the object that has the aspect that has the property you want to modify. The variable name is used to reference the object. See Working with References for details.
	Structure Selector	Select the structure in which the child object is located.
	Enter the name of the child object.	The name is used to search for the object. The name may contain several levels. Example: MyChild MyChild/Test MyChild/Test/Next Level Child
	Enter the name of the aspect that holds the property you want to modify.	The system will search for this aspect via the aspect name.
	Enter the name of the property you want to modify	The system will search for this property via the property name.

Table 46. Modify Property Operation (Continued)

Section	Field	Description
	Enter the value you want to assign to the property	Enter the value new value for the property. You can use substitutions. You can also use script expressions. Examples: $FunctionName$_Test Ti * 3.14 SIN(TI) + COS(Kr) MyFunction(Kr) + 1.2
	Data Type	Select the data type of the property.

Modify Property of Existing Object

This operation lets you modify a property of an object that already exists in the system where the Reuse Instruction is executed.

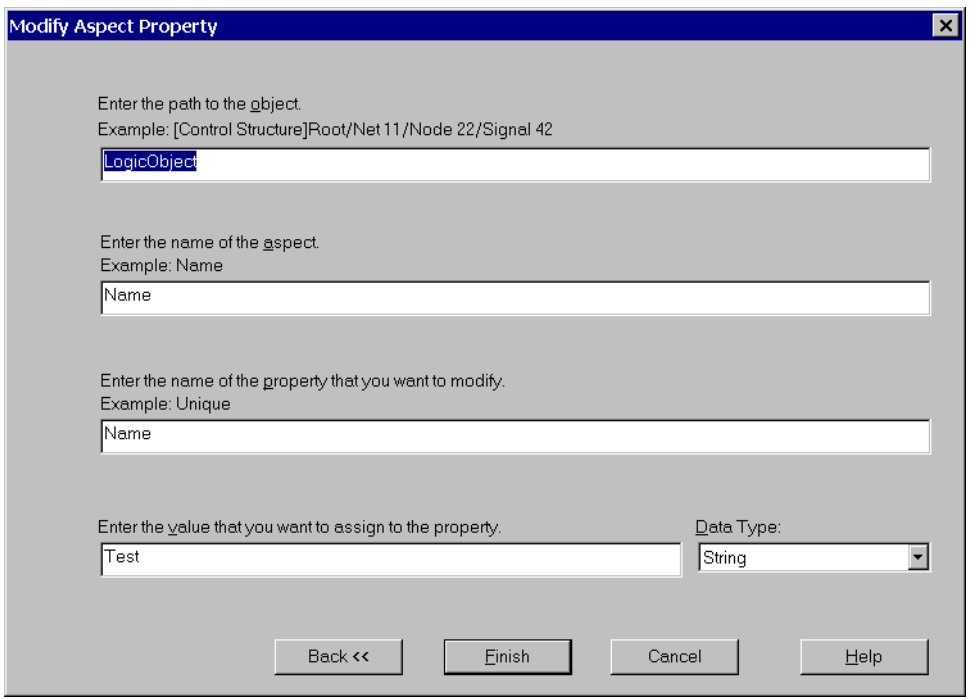


Figure 352. Modify Property Operation

Table 47. Modify Property Operation

Section	Field	Description
	Enter the path to the object.	The path is used to search for the object. The path may contain a structure. Examples: [functional structure]root root/myobject myobject myobject*
	Enter the name of the aspect	Name of the aspect that has the property you want to modify.
	Enter the name of the property you want to modify	The system will search for this property via the property name.
	Enter the value that you want to assign to the property.	Enter the value new value for the property. You can use substitutions. You can also use script expressions. Examples: $FunctionName$_Test Ti * 3.14 SIN(TI) + COS(Kr) MyFunction(Kr) + 1.2
	Data Type	Select the data type of the property,

Modify Property of Object with the Reuse Instruction

This operation lets you modify a property of an aspect which belongs to the object with the Reuse Instruction (the start object).

Script Block

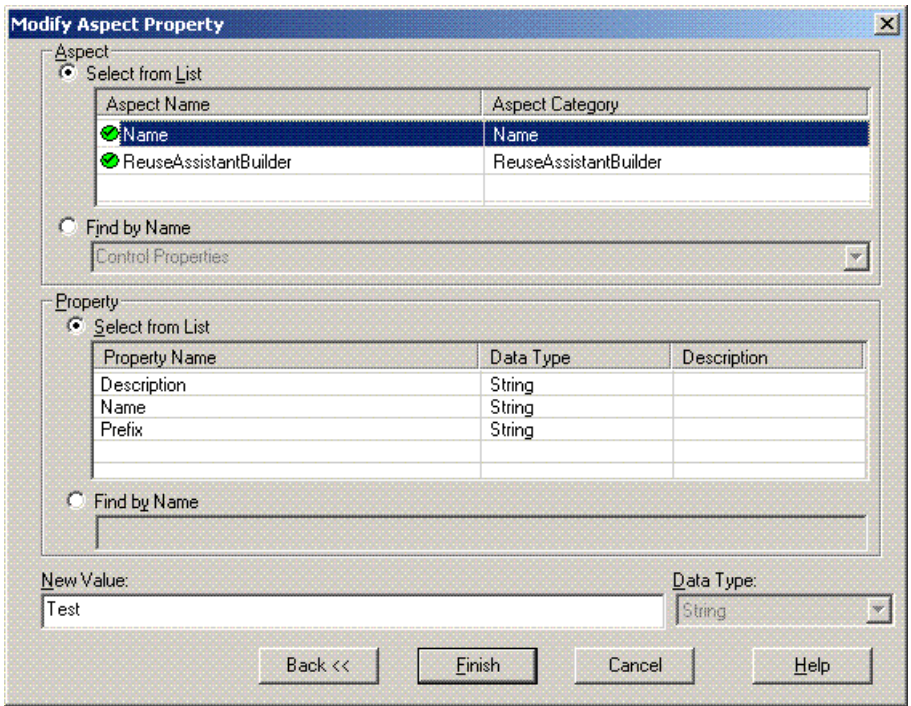


Figure 353. Modify Property

Table 48. Modify Property Operation

Section	Field	Description
Aspect	Select from List	If this radio button is selected you may select the aspect that has the property that you want to modify. This lists displays all available at the object with the Reuse Instruction. The aspect name is used by the system to find the aspect.
Aspect	Find by Name	If this radio button is selected you can enter the name of the aspect that has the property that you want to modify. The name is not case sensitive. The aspect name is used by the system to find the aspect.
Property	Select from List	If this radio button is selected you may want to select the property you want to modify from the list below. The property name is used to address the property.
Property	Find by Name	If this radio button is selected you can enter the name of the property into the edit field below. The name is not case sensitive. You also need to define the data type of the property in the Data Type field.

Table 48. Modify Property Operation (Continued)

Section	Field	Description
	New Value	Enter the value new value for the property. You can use substitutions. You can also use script expressions. Examples: $FunctionName$_Test Ti * 3.14 SIN(TI) + COS(Kr) MyFunction(Kr) + 1.2
	Data Type	Select the data type of the property. The selection is automatically done for you if you select a property from the property list.

This operation allows you to enter your own script code. For details regarding script programming see Architect - Script References. This operation displays the following user interface.

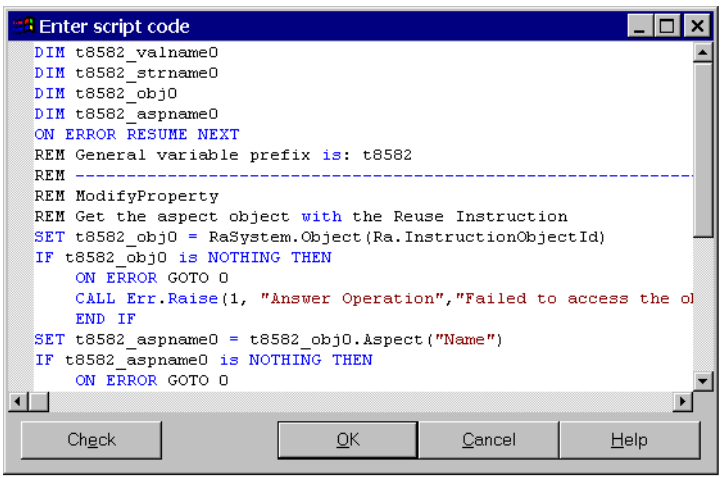


Figure 354. Script Block Operation User Interface

Table 49. Script Block Operation

Section	Field or Key	Description
Edit Field	CTRL+A	Select all the text in the edit field.
Edit Field	CTRL+F	Open the Find dialog.
Edit Field	F3	Find Next
Edit Field	CTRL+F3	Select & Find word at the cursor.
Edit Field	CTRL+F2	Set mark.
Edit Field	F2	Go to next mark.
Edit Field	SHIFT+F2	Go to previous mark.
Edit Field	CTRL+Z	Undo.
Edit Field	CTRL+Y	Redo.

Table 49. Script Block Operation (Continued)

Section	Field or Key	Description
Edit Field	CTRL+H	Open Find + Replace dialog.
Edit Field	CTRL+L	Delete current row.
	Check	Check the script code. This operation checks, if the script code can be loaded into the script engine.

Insert Object

This operation places an aspect object in a structure. The operation shows the following user interface.

I

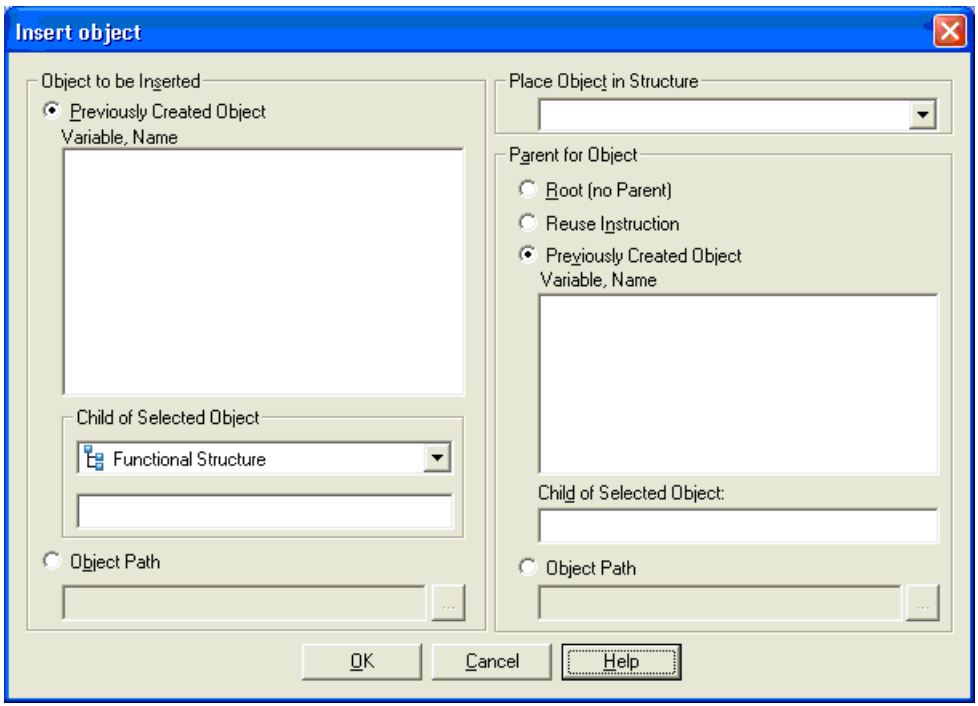


Figure 355. Insert Object Operation

Table 50. Insert Object Operation

Section	Field	Description
Object to be Inserted	Previously Created Object	If selected, an object that is created by the Reuse Instruction will be placed. Select the object from the list below. The object variable name is used to reference the object. See for details.
Object to be Inserted > Child of selected object	Structure Selector	Select the structure in which the child object is located.
Object to be Inserted > Child of selected object	Edit Field	The name is used to search for the object. The name may contain several levels. Example: MyChild MyChild/Test MyChild/Test/Next Level Child
Object to be Inserted	Object Path	Enter the search path to the parent object. This path may include a structure.
Place Object in Structure	Structure Selector	Select the structure where the object should be placed.
Parent for Object	Root	If selected, the object will be placed on root level.
Parent for Object	Reuse Instruction	If selected, the object will be placed as a child of the object with the Reuse Instruction. Note: it is assumed that the object with the Reuse Instruction exists in the selected structure.
Parent for Object	Previously Created Object	The object will be placed as a child to an object that is created by the Reuse Instruction. The variable name is used to reference the parent object. See Working with References for details.

Table 50. Insert Object Operation (Continued)

Section	Field	Description
Parent for Object	Child of Selected Object	The object will be places as a child of a child object of an object that is created by the Reuse Instruction (complicated!). Enter the name of the child object. Example: MyChild MyChild/Test MyChild/Test/Next Level Child
Parent for Object	Object Path	Enter the search path to the parent object. This path may include a structure. Please note that this structure may be different from the structure where is object is going to be placed. If the object which you are addressing contains a character that is used as separator (like "."), you need to escape this character by **two** "\" characters. Example: For the object path "Test/AI1.1" you must enter: Test/AI1\\.1

Override Aspect

This operation overrides an otherwise inherited aspect of an aspect object. The aspect shows the following user interface.

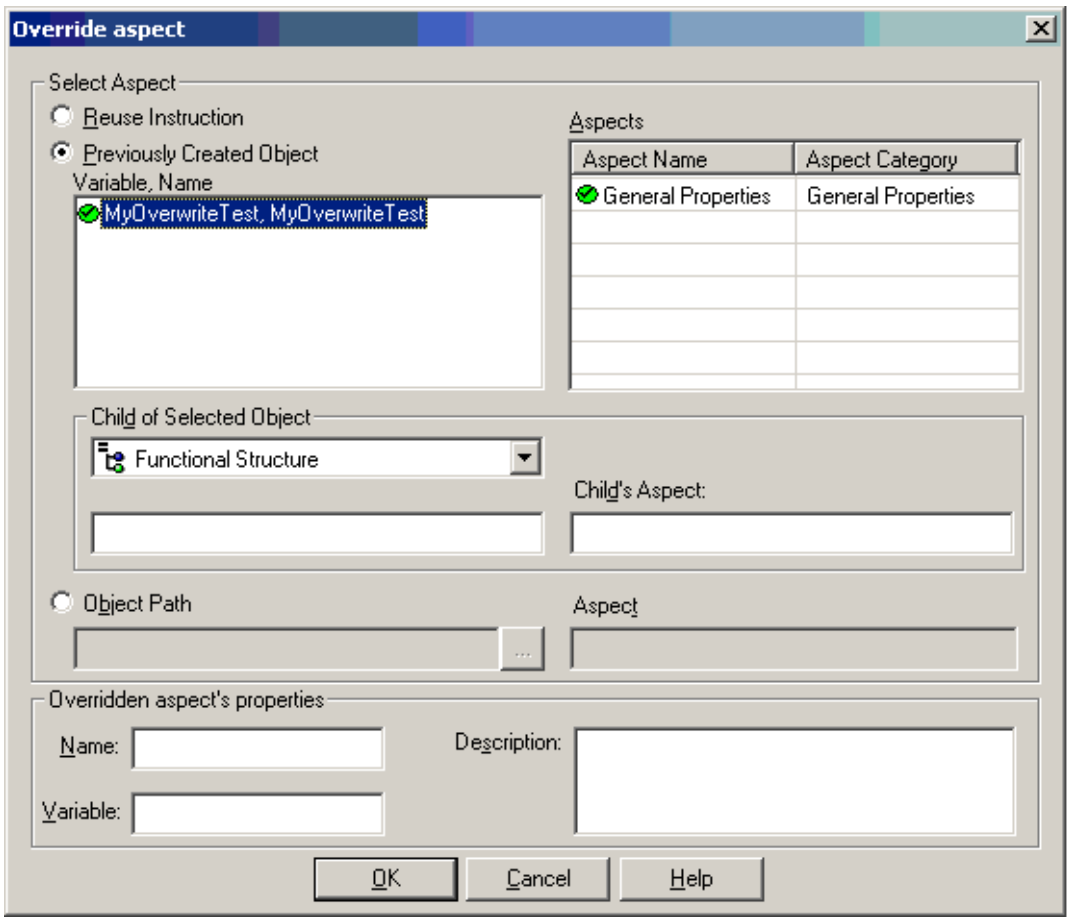


Figure 356. Override Aspect User Interface

The dialog has the following input fields.

Table 51. Override Aspect Operation

Section	Field or Button	Description
Select Aspect	Reuse Instruction	This operation will overwrite an aspect that is part of the object with the Reuse Instruction.
Select Aspect	Previously Created Object	This operation will overwrite an aspect that is part of an object that has been created by this Reuse Instruction.
Select Aspect > Child of Selected Object	Structure Selector	Select the structure in which the child object is located.
Select Aspect > Child of Selected Object	Child Path	The object will be places as a child of a child object of an object that is created by the Reuse Instruction (complicated!). Enter the name of the child object. Example: MyChild MyChild/Test MyChild/Test/Next Level Child
Select Aspect	Object Path	If selected, the edit field below the radio button specifies the path to the object for which an aspect will be overwritten. If the object which you are addressing contains a character that is used as separator (like “.”), you need to escape this character by **two** “\” characters. Example: For the object path “Test/AI1.1” you must enter: Test/AI1\\.1

Table 51. Override Aspect Operation (Continued)

Section	Field or Button	Description
Select Aspect	Aspect	If the radio button Object Path is selected, the edit field contains the name (Aspect Name) of the aspect that should be overwritten.
Select Aspect	Aspects	For the selections Reuse Instruction and Previously Created Object, this list shows all aspects which can be overwritten. Select one of the aspects in this list.
Overwritten Aspect's Properties	Name	Name (aspect name) of the newly created aspect.
Overwritten Aspect's Properties	Description	Description text for the new aspect.
Overwritten Aspect's Properties	Variable	The name of the variable which can be used to reference this new object type from other operations within this Reuse Instruction. For details see Working with References. Allowed Values: A...Z; a...z, 0...9, _ The variable name must start with a letter.

Aspect Object Type Operations

This section lists all available operations to create and change aspect object types. For a detailed description of the concept of object types, please refer to documentation *System 800xA, System Planning (3BSE041389*).*

New Object Type

This operation creates a new aspect object type in the Object Type structure. The operation shows the following user interface.

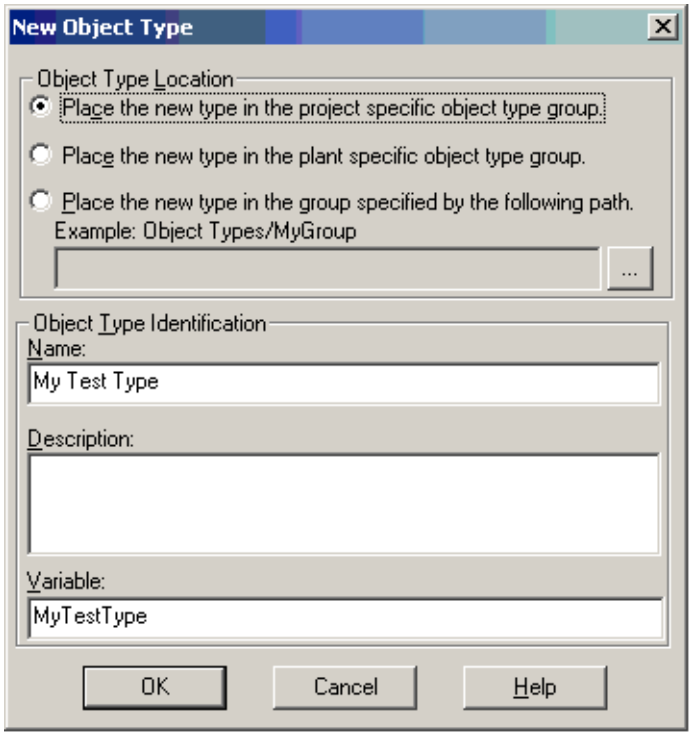


Figure 357. New Object Type Operation

The following table shows the input fields of this dialog.

Table 52. New Object Type Operation

Section	Field or Button	Description
Object Type Location	Place new type in the project specific object type group	The option will create the new type in the group with the name: Project <Project Name> Specific This group is automatically created when the project is created.
Object Type Location	Place new type in the plant specific object type group	The option will create the new type in the group with the name: Plant <Project Name> Specific This group is automatically created when the project is created.
Object Type Location	Place the new type in the group specified by the following path	The option will create the new type in the group that is specified by the path in the edit field below the radio button. If the object which you are addressing contains a character that is used as separator (like "."), you need to escape this character by **two** "\" characters. Example: For the object path "Test/AI1.1" you must enter: Test/AI1\\.1
Object Type Identification	Name	The name of the new object type.

Table 52. New Object Type Operation (Continued)

Section	Field or Button	Description
Object Type Identification	Description	The description of the new object type
Object Type Identification	Variable	The name of the variable which can be used to reference this new object type from other operations within this Reuse Instruction. For details see Working with References. Allowed Values: A...Z; a...z, 0...9, _ The variable name must start with a letter.

Modify Aspect Control

This operation modifies the aspect control of an aspect object type. The operation shows the following dialog.

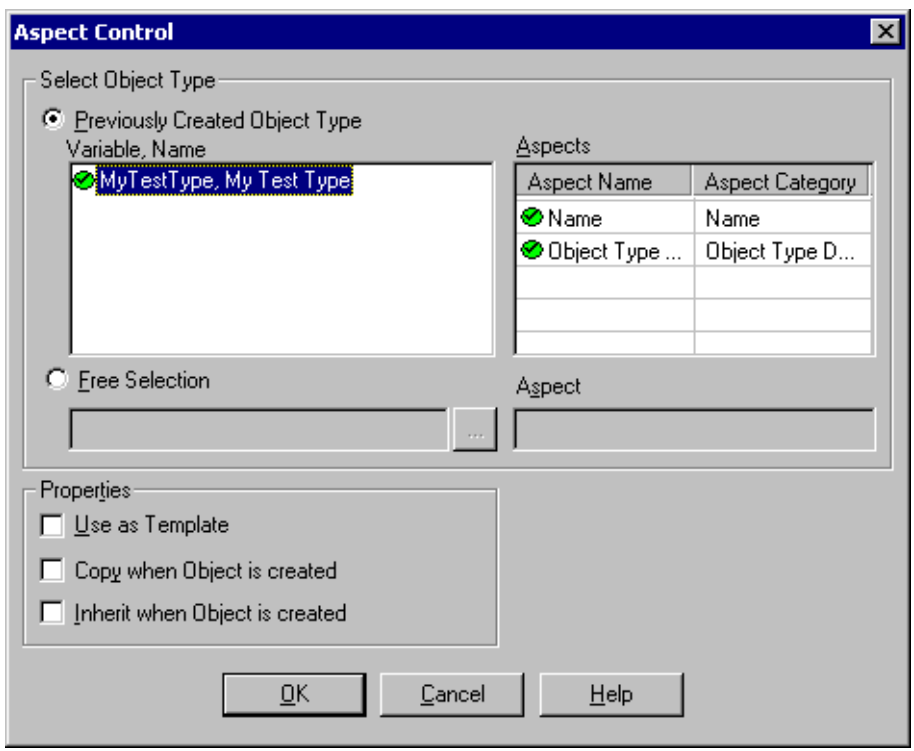


Figure 358. Modify Aspect Control

Table 53. Modify Aspect Control Operation

Section	Field or Button	Description
Select Object Type	Previously Created Object Type	If this button is checked, the aspect control of an object type that is created by this Reuse Instruction will be modified. Select the object type from the list below the radio button.
Select Object Type	Free Selection	The object type that is modified is given its path. Enter the path to the object type in the edit field below the radio button.
Select Object Type	Aspects	This list displays the aspects for which the aspect control can be modified. The list is disabled if the object type is selected via path.
Select Object Type	Aspect	Enter the name of the aspect for which the aspect control should be changed. The edit field is only valid if the object type is selected via path.
Properties	Use as Template	If selected, the flag “Use as Template” is set for the selected aspect in the object type definition of the selected object type.
Properties	Copy when Object is Created	If selected, the flag “Copy when Object is Created” is set for the selected aspect in the object type definition of the selected object type.
Properties	Inherit when Object is Created	If selected, the flag “Inherit when Object is Created” is set for the selected aspect in the object type definition of the selected object type.

Modify Category Control

This operation modifies the aspect category control of an aspect object type. The operation shows the following dialog.

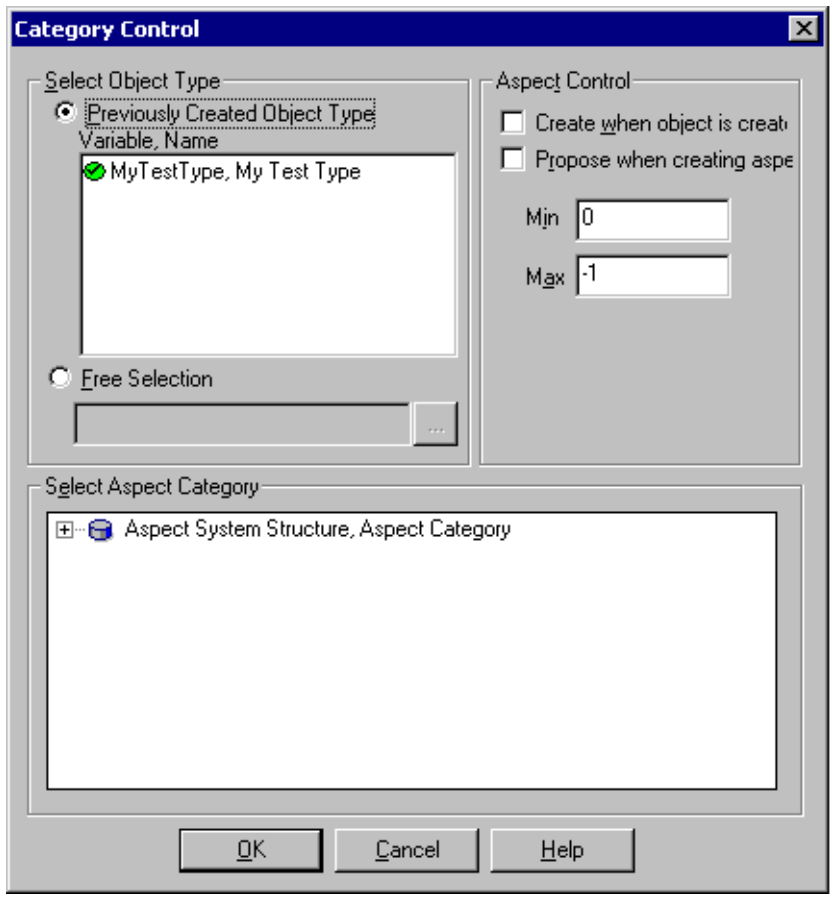


Figure 359. Modify Category Control

Table 54. Modify Category Control Operation

Section	Field or Button	Description
Select Object Type	Previously Created Object Type	If this button is checked, the operation is applied to an object type that is created by this Reuse Instruction. Select the object type from the list below the button.
Select Object Type	Free Selection	If this button is checked, the operation is applied to an object type this already exists in the system where the Reuse Instruction is executed. Enter the path to the object into the edit field below the button. If the object which you are addressing contains a character that is used as separator (like "."), you need to escape this character by **two** "\" characters. Example: For the object path "Test/AI1.1" you must enter: Test/AI1\\.1
Select Aspect Category	Aspect Browser	Select the aspect category for which you want to define an aspect control.
Aspect Control	Create when Object is Created	If checked, the aspect of the selected category is created when creating an instance from this object type.
Aspect Control	Min.	Enter the minimal number of aspects of the selected category that are required for an instance of the selected object type.
Aspect Control	Max	Enter the maximal number of aspects of the selected category that are allows for an instance of the selected object type.

Modify Child Control

This operation modifies the child control of an aspect object type. The operation shows the following user interface.

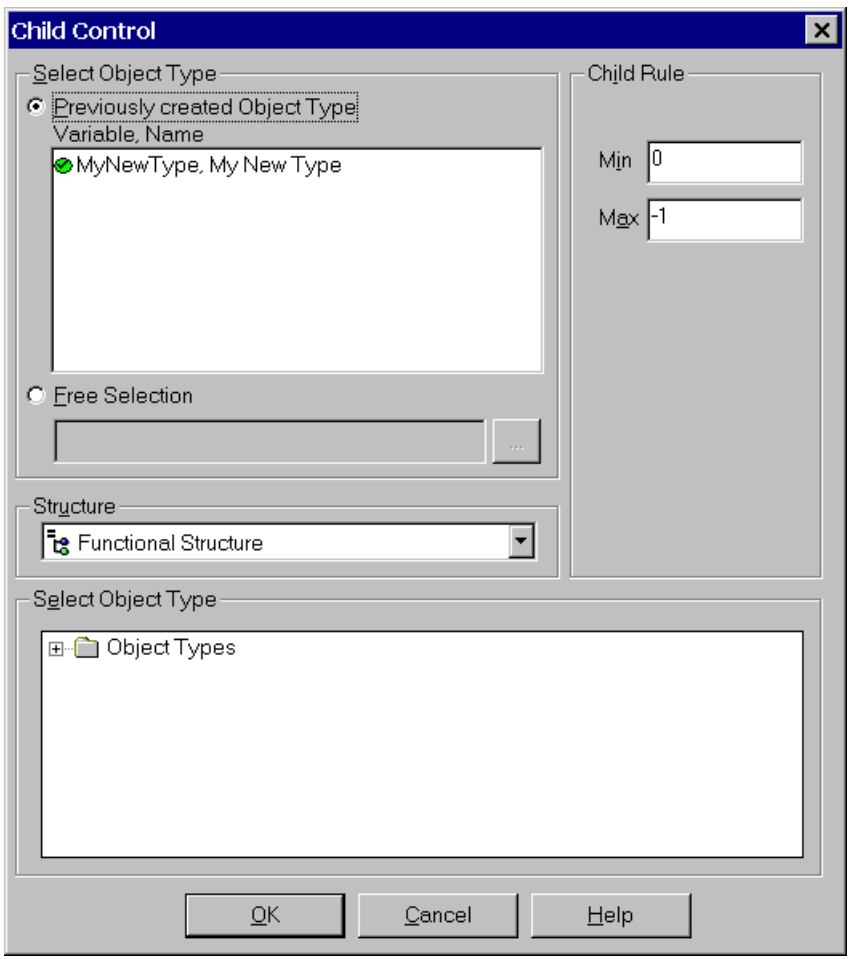


Figure 360. Child Control

Table 55. Modify Child Control Operation

Section	Field or Button	Description
Select Object Type	Previously Created Object Type	If this button is checked, the operation is applied to an object type that is created by this Reuse Instruction. Select the object type from the list below the button.
Select Object Type	Free Selection	If this button is checked, the operations are applied to an object type this already exists in the system where the Reuse Instruction is executed. Enter the path to the object into the edit field below the button. If the object which you are addressing contains a character that is used as separator (like "."), you need to escape this character by **two** "\" characters. Example: For the object path "Test/AI1.1" you must enter: Test/AI1\\.1
Structure	Structure Selector	Select the structure for which the child control should be applied.
Select Object Type	Object Browser	Select the object type of the child object for which the child control should be applied.
Child Rule	Min.	Enter the minimal number of child objects of the selected type.
Child Rule	Max	Enter the maximal allows number of child objects of the selected type.

Set Super Type

This operation defines the super type of an aspect object type. The operation shows the following user interface.

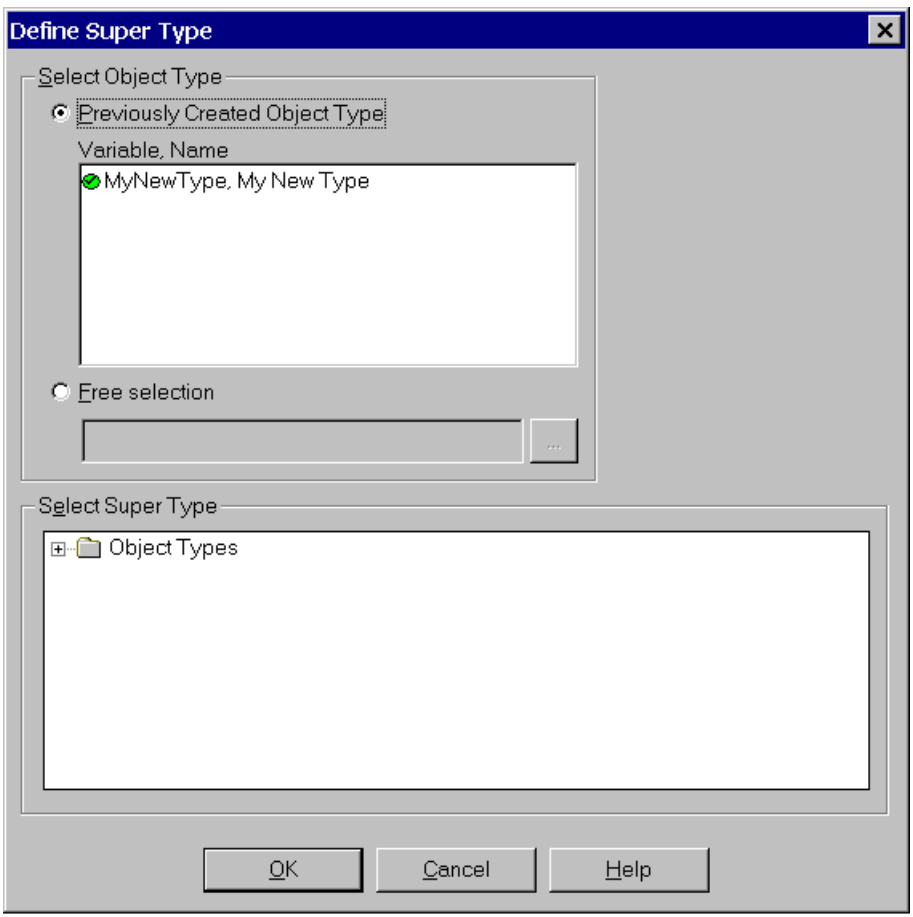


Figure 361. Define Super Type Operation

Table 56. Set Super Type Operation

Section	Field or Button	Description
Select Object Type	Previously Created Object Type	If this button is checked, the operation is applied to an object type that is created by this Reuse Instruction. Select the object type from the list below the button.
Select Object Type	Free Selection	If this button is checked, the operations are applied to an object type this already exists in the system where the Reuse Instruction is executed. Enter the path to the object into the edit field below the button. If the object which you are addressing contains a character that is used as separator (like "."), you need to escape this character by **two** "\" characters. Example: For the object path "Test/AI1.1" you must enter: Test/AI1\\.1
Select Super Type	Object Browser	Select the object type that should become the new super type.

Control Builder M Specific Operations

Add Variable

This operation adds a new variable to a Single Control Module, a Program or an Application. The operation shows the following user interface:

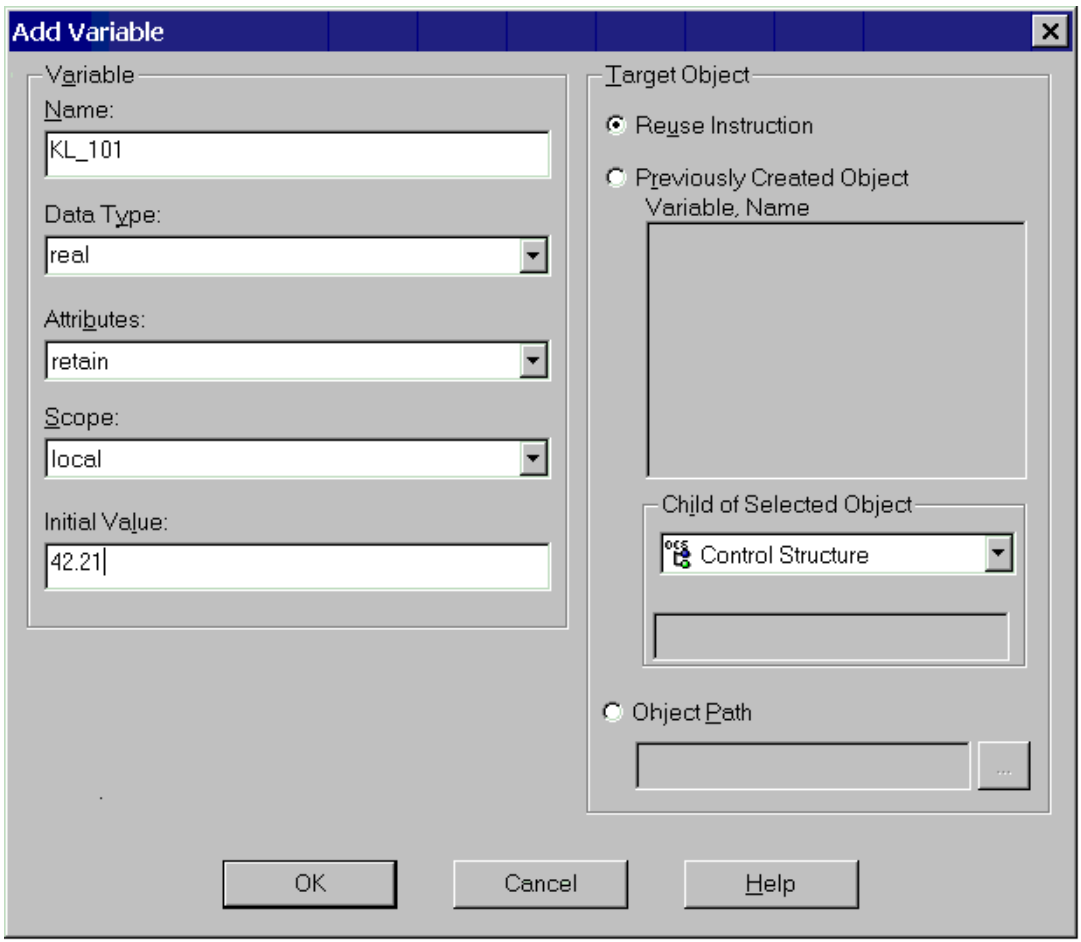


Figure 362. Add Variable

Table 57. Add Variable Operation

Section	Field or Button	Description
Variable	Name	Name of the new variable. Substitutions are allowed. The evaluation of the name must result in a string that is a valid variable name in the context where the new variable will be created.
Variable	Data Type	Data type of the new variable. Substitutions are allowed. The pick list contains the basic data type of the Control Builder.
Variable	Attributes	Attributes of the new variable. The pick list contains all attributes that are defined by the Control Builder M.
Variable	Scope	Scope of the new variable. For Single Control Modules, the scope “local” or “external” is allowed.
Variable	Initial Value	Initial value of the variable. Substitutions are allows.
Target Object	Reuse Instruction	If this radio button is selected, the aspect will be created on the object with the Reuse Instruction (the object where you start the Builder).
Target Object	Previously Created Object	If this radio button is selected, the new aspect will be created at an object that is created by the Reuse Instruction. Select the object from the list. The reference is build using the variable name of the object. See Working with References for details about references.

Table 57. Add Variable Operation (Continued)

Section	Field or Button	Description
Target Object > Child of Selected Object	Structure Selector	This selection is relevant if the radio button Previously Created Object is selected. In this case, you can create an aspect at child object of the selected object. The structure selector defines, in which structure this child is located.
Target Object > Child of Selected Object	Edit Field	Enter the name (Name.Name) of the child object.
Target Object	Object Path	If this radio button is selected, the target object is identified via a path. Enter the path into the edit field. Example: `[Functional Structure]Root/PI Regulator` If the object which you are addressing contains a character that is used as separator (like "."), you need to escape this character by **two** "\" characters. Example: For the object path "Test/AI1.1" you must enter: Test/AI1\\.1

Add Parameter

This operation adds a new parameter to a Single Control Module. The operation shows the following user interface:

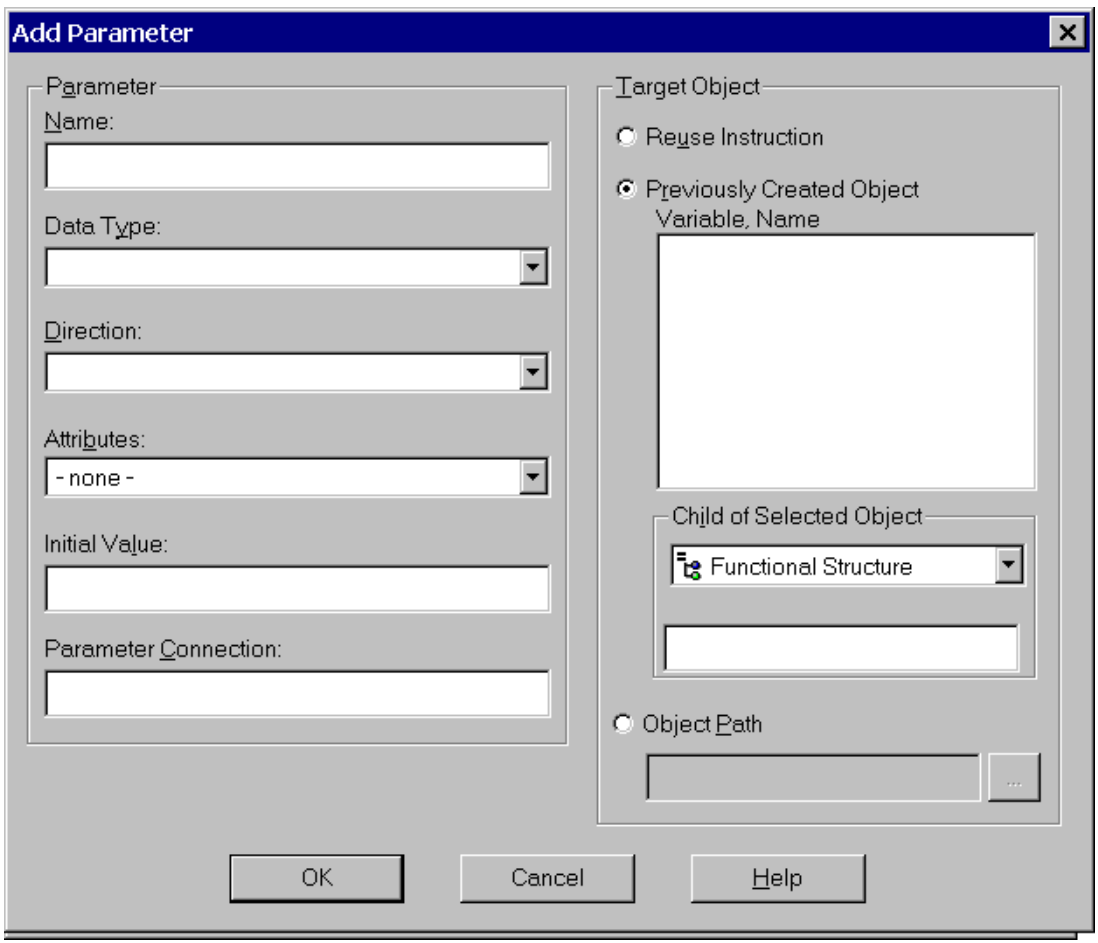


Figure 363. Add Parameter

Table 58. Add Parameter Operation

Section	Field or Button	Description
Parameter	Name	Name of the new parameter. Substitutions are allowed. The evaluation of the name must result in a string that is a valid parameter name in the context where the new parameter will be created.
Parameter	Data Type	Data type of the new parameter. Substitutions are allowed. The pick list contains the basic data type of the Control Builder.
Parameter	Direction	Direction of the new parameter. For Single Control Modules, IN_OUT must be selected.
Parameter	Attributes	Attributes of the new parameter. The pick list contains all valid attribute combination.
Parameter	Initial Value	Initial value of the parameter. Substitutions are allows.
Parameter	Parameter Connection	Default connection (source / sink) for the parameter.
Target Object	Reuse Instruction	If this radio button is selected, the aspect will be created on the object with the Reuse Instruction (the object where you start the Builder).
Target Object	Previously Created Object	If this radio button is selected, the new aspect will be created at an object that is created by the Reuse Instruction. Select the object from the list. The reference is build using the variable name of the object. See Working with References, for details about references.

Table 58. Add Parameter Operation (Continued)

Section	Field or Button	Description
Target Object > Child of Selected Object	Structure Selector	This selection is relevant if the radio button Previously Created Object is selected. In this case, you can create an aspect at child object of the selected object. The structure selector defines, in which structure this child is located.
Target Object > Child of Selected Object	Edit Field	Enter the name (Name.Name) of the child object.
Target Object	Object Path	If this radio button is selected, the target object is identified via a path. Enter the path into the edit field. Example: `[Functional Structure]Root/PI Regulator` If the object which you are addressing contains a character that is used as separator (like "."), you need to escape this character by **two** "\" characters. Example: For the object path "Test/AI1.1" you must enter: Test/AI1\\.1

Add Alarm

This operation adds a new Alarm Condition an existing aspect of the category Control Alarm Event. The operation shows the following user interface:

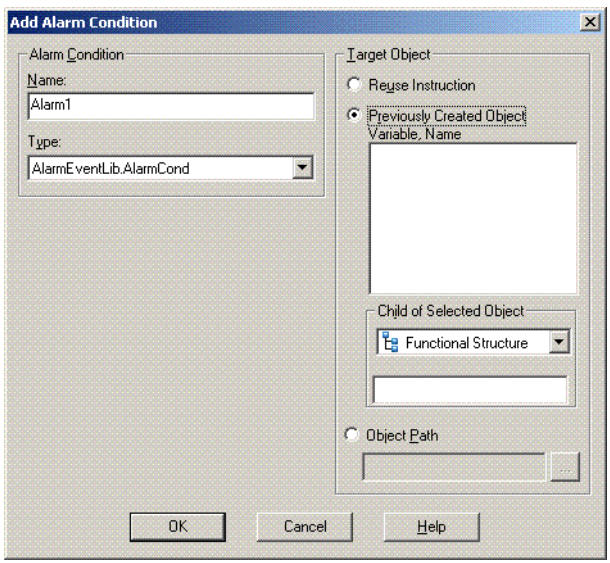


Figure 364. Add Alarm Condition

Table 59. Add Alarm Condition Operation

Section	Field or Button	Description
Alarm Condition	Name	Name of the new Alarm Condition. Substitutions are allowed. The string defines the name of the Function Block or Control Module instance that checks the alarm condition.
Alarm Condition	Type	Name of the Function Block type or Control Module type that is used to check the alarm condition. All supported types are included in the pick list.
Target Object	Reuse Instruction	If this radio button is selected, the aspect will be created on the object with the Reuse Instruction (the object where you start the Builder).
Target Object	Previously Created Object	If this radio button is selected, the new aspect will be created at an object that is created by the Reuse Instruction. Select the object from the list. The reference is build using the variable name of the object. See Working with References for details about references.
Target Object > Child of Selected Object	Structure Selector	This selection is relevant if the radio button Previously Created Object is selected. In this case, you can created an aspect at child object of the selected object. The structure selector defines, in which structure this child is located.

Table 59. Add Alarm Condition Operation (Continued)

Section	Field or Button	Description
Target Object > Child of Selected Object	Edit Field	Enter the name (Name.Name) of the child object.
Target Object	Object Path	If this radio button is selected, the target object is identified via a path. Enter the path into the edit field. Example: `[Functional Structure]Root/PI Regulator` If the object your are addressing contains a character that is used as separator (like "."), you need to escape this character by **two** "\" characters. Example: For the object path "Test/AI1.1" you must enter: Test/AI1\\.1

Modify Alarm

This operation changes the parameters of an existing alarm condition. The operation shows the following user interface:

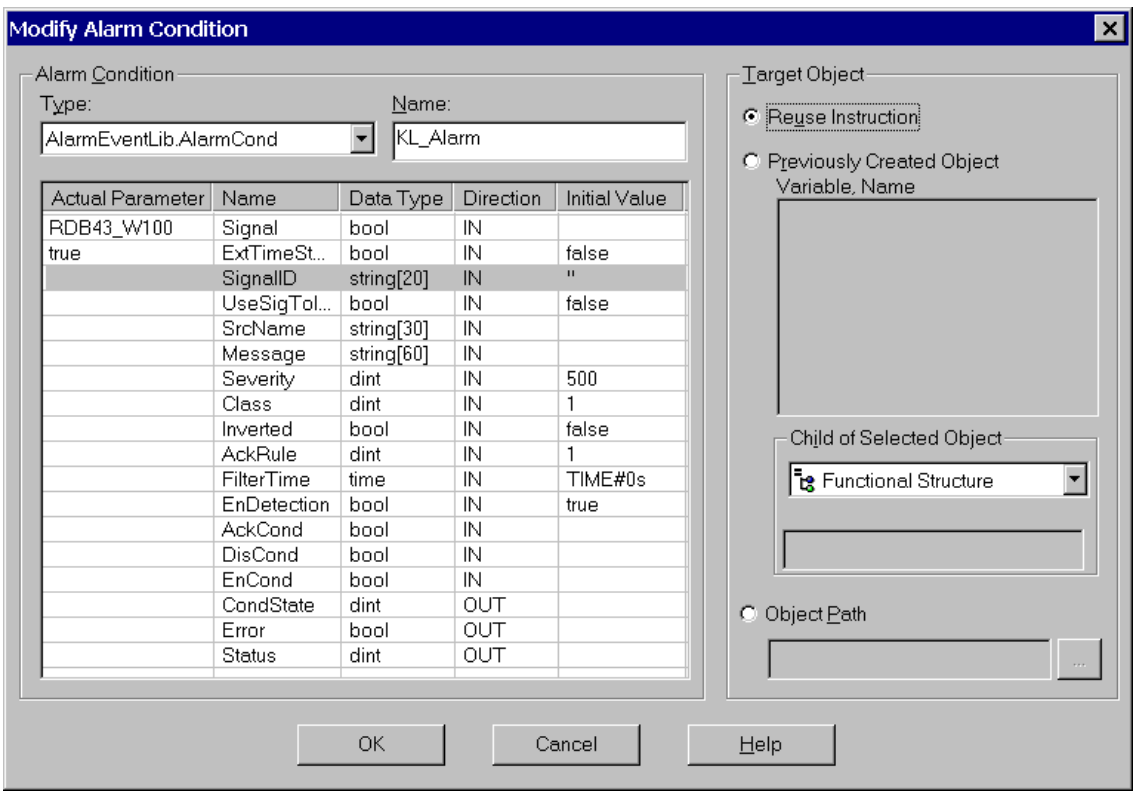


Figure 365. Modify Alarm Condition

Table 60. Modify Alarm Operation

Section	Field or Button	Description
Alarm Condition	Type	Name of the Function Block type or Control Module type that is used to check the alarm condition. All supported types are included in the pick list.
Alarm Condition	Name	Name of the Alarm Condition that is modified.
Alarm Condition		The table contains all parameters of the Alarm Condition that can be modified. The parameters depend on the selected Alarm Condition Type. To change a parameter select the row with the parameter, then click the Actual Parameter field with the mouse.
Target Object	Reuse Instruction	If this radio button is selected, the aspect will be created on the object with the Reuse Instruction (the object where you start the Builder).
Target Object	Previously Created Object	If this radio button is selected, the new aspect will be created at an object that is created by the Reuse Instruction. Select the object from the list. The reference is build using the variable name of the object. See Working with References for details about references.
Target Object > Child of Selected Object	Structure Selector	This selection is relevant if the radio button Previously Created Object is selected. In this case, you can create an aspect at child object of the selected object. The structure selector defines, in which structure this child is located.

Table 60. Modify Alarm Operation (Continued)

Section	Field or Button	Description
Target Object > Child of Selected Object	Edit Field	Enter the name (Name.Name) of the child object.
Target Object	Object Path	If this radio button is selected, the target object is identified via a path. Enter the path into the edit field. Example: `[Functional Structure]Root/PI Regulator` If the object which you are addressing contains a character that is used as separator (like "."), you need to escape this character by **two** "\" characters. Example: For the object path "Test/AI1.1" you must enter: Test/AI1\\.1

Generate FB Calls

This operation invokes the Generate FB calls operation of the Control Properties aspect of a Control Application object or a Control Program object. The operation generates an IEC1131 Structured Text code block in the Control Builder. The code block includes an invocation for each Function Block object that is a child of the Program or Application in the Control Structure. Function Blocks that do not have the __AutoGen set to TRUE are excluded. You can set the __AutoGen property from a Reuse Instruction using the Modify Property operation. The __AutoGen property is exposed by the Control Properties aspect of a Function Block aspect object.

The following figure shows the user interface of the Generate FB Calls operation.

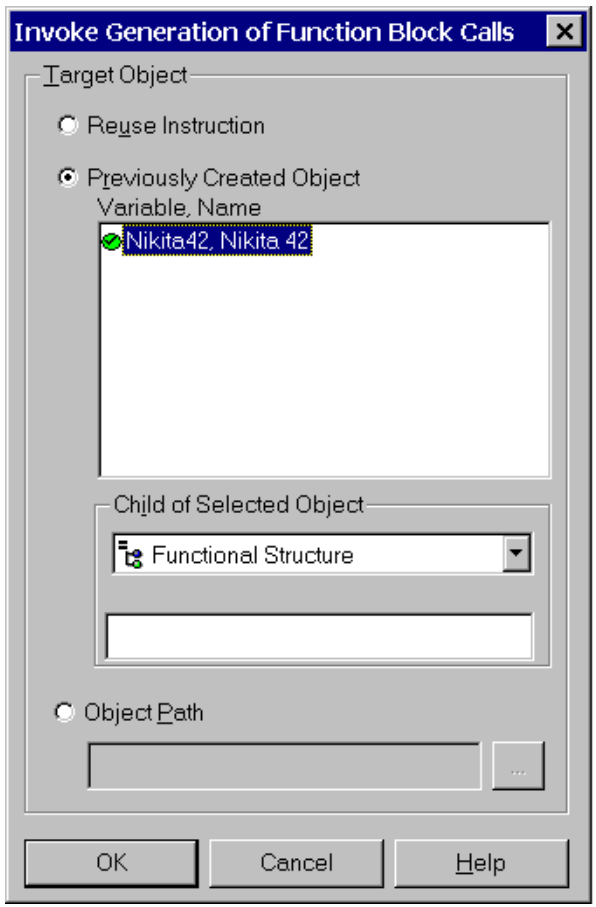


Figure 366. Automatic Generation of Function Block Calls

Table 61. Generate FB Calls Operation

Section	Field or Button	Description
Target Object	Reuse Instruction	If this radio button is selected, the operation will be invoked on the object with the Reuse Instruction (the object where you start the Builder).
Target Object	Previously Created Object	If this radio button is selected, the operation will be invoked for an object that is created by the Reuse Instruction. Select the object from the list. The reference is build using the variable name of the object. See Working with References for details about references.
Target Object > Child of Selected Object	Structure Selector	This selection is relevant if the radio button Previously Created Object is selected. In this case, you can invoke the operation on a child object of the selected object. The structure selector defines, in which structure this child is located.

Table 61. Generate FB Calls Operation (Continued)

Section	Field or Button	Description
Target Object > Child of Selected Object	Edit Field	Enter the name (Name.Name) of the child object.
Target Object	Object Path	If this radio button is selected, the target object is identified via a path. Enter the path into the edit field. Example: `[Functional Structure]Root/PI Regulator` If the object which you are addressing contains a character that is used as separator (like "."), you need to escape this character by **two** "\" characters. Example: For the object path "Test/AI1.1" you must enter: Test/AI1\\.1

Index

Symbols

A

B

C

D

M

N

O

P

Q

R

S

T

U

V

W

X

Revision History

Introduction

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.

Revision History

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	Updated for 800xA 5.1 Rev A	May 2011
B	Updated for 800xA 5.1 Rev B	June 2012
C	Updated for 800xA 5.1 FP 3	August 2012
D	Updated for 800xA 5.1 FP 4	February 2013
E	Updated for 800xA 5.1 Rev D	December 2013
F	Updated for 800xA 5.1 Rev E/FP4 Rev E	July 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 5, IO Allocation	Under *capacity and performance* subsection, information note for improved performance of automatic update is added.
Section 7, Document Manager	Under *Understanding Document Manager Template* subsection, added information about loop diagram template.
Appendix A, Engineering Templates	Added information in *Working with Engineering Templates* 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 3, Bulk Data Manager	Under *Automatic Update of CBM* subsection, information note for improved performance of automatic update is added.
Section 3, Bulk Data Manager	Under *Automatic Update of CBM* subsection, information note update properties of CBM_SignalParameter with respect to CBM.
Appendix A, Engineering Templates	Added information in *Working with Engineering Templates* subsection regarding columns of BDM template.

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 4, Bulk SPL	A new section Bulk SPL has been added.
Section 5, IO Allocation	Subsection PROFINET Device Support has been added.

Updates in Revision Index D

The following table shows the updates made in this User Manual for 800xA 5.1 FP4.

Updated Section/Sub-section	Description of Update
Section 5, IO Allocation	Information regarding CBM_SignalParameter has been added.

Updates in Revision Index E

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 2, Engineering Workplace	Added a bullet point, referring the Synchronization of Control Builder Name aspect and Name aspect.

Updates in Revision Index F

The following table shows the updates made in this User Manual for 800xA 5.1 Rev E/FP4 Rev E.

Updated Section/Sub-section	Description of Update
Section 3, Using Formulas Within a Data Area	1. Added the following information: *The Excel formula "OFFSET" is used for referencing the cells dynamically. An example is shown below with reference to Figure 27. The BDM formula (applied on column H) is used to subtract content of "Min" column from "Max" column for every row. In the formula, every cell of the particular row is referenced using the "OFFSET" function. An example of the formula used in the excel sheet is as follows:* *$$Formula=OFFSET(F1,ROW()-1,0)-OFFSET(G1,ROW()-1,0)* *Syntax:* *OFFSET(Reference cell, Row offset, Column Offset)* 2. Changed the image for *Figure: Referencing the cells dynamically*
Section 5, Writing Allocation into CBM	Added the following information: • *While allocating the pulse signals to DSDP 160 IO Boards, user needs to manually enter the datatype in the signal information aspect.* • *After allocation of multiple pulse signals to DP910*(F12 P) card and on performing **Write Allocation into CBM**, errors are generated. IO Allocation cannot be used to allocate multiple pulse signal to DP910*(F12 P) card.*

Updated Section/Sub-section	Description of Update
Section 5, Signal Data Entry	Added the following information: *For **SignalRange** property as part of the CBM_SignalParameter aspect, 4-20mA or 4..20mA can be used interchangeably without any impact on the functionality*
Section 5, IO Allocation	Added a new subsection: *Prerequisite for PROFIBUS Module Naming*
Section 3, Using Formulas Within a Data Area	Updated the following image: *Figure: Referencing the cells dynamically*

Contact us

www.abb.com/800xA
www.abb.com/controlsystems

3BDS011223-510 F

Power and productivity
for a better world™