

System 800xA

System Planning

System Version 6.0

Power and productivity
for a better world™



System 800xA

System Planning

System Version 6.0

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.

Products described or referenced in this document are designed to be connected, and to communicate information and data via a secure network. It is the sole responsibility of the system/product owner to provide and continuously ensure a secure connection between the product and the system network and/or any other networks that may be connected.

The system/product owners must establish and maintain appropriate measures, including, but not limited to, the installation of firewalls, application of authentication measures, encryption of data, installation of antivirus programs, and so on, to protect the system, its products and networks, against security breaches, unauthorized access, interference, intrusion, leakage, and/or theft of data or information.

ABB verifies the function of released products and updates. However system/product owners are ultimately responsible to ensure that any system update (including but not limited to code changes, configuration file changes, third-party software updates or patches, hardware change out, and so on) is compatible with the security measures implemented. The system/product owners must verify that the system and associated products function as expected in the environment they are deployed.

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

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

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

TRADEMARKS

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

Copyright © 2003-2016 by ABB.
All rights reserved.

Release: September 2016
Document number: 3BSE041389-600 B

Table of Contents

About this User Manual

Version Described in this User Manual	21
User Manual Conventions	21
Warning, Caution, Information, and Tip Icons	22
Terminology.....	22
Related Documentation	22

Section 1 - Preparation

Introduction	23
System 800xA Hardware Requirements.....	23
Functional Areas.....	24
Planning Considerations.....	25
Using Structures	26
Planning Considerations.....	26
Using Names and Designations.....	29
Planning Considerations.....	30
User Strategy	30
User Groups	31
Permissions	31
Planning Considerations for User Groups and Permissions	32
Protection of Engineering Work.....	32
Re-authentication and Double-authentication.....	33
Audit Trail	33
Performance and Capacity.....	34
Planning Considerations.....	34
External Data	35

Libraries	35
Standard Libraries and Templates.....	36
Use of Industry Specific Library Solutions	36
Predefined Workplaces.....	37
Operator Workplace	37
Planning Considerations	38
Plant Explorer Workplace and Engineering Workplace	39
Maintenance Workplace.....	40
Planning Consideration.....	41
Navigation Strategies	41
Planning Considerations for Basic Navigation	42
Additional Navigation.....	43
Security Strategy	44
Information System (IS) Security	44
Security Policy	45
Network Security Zones	46
Use cases for Connections via Firewalls.....	46
Planning Access Through a Firewall	47
Life Cycle.....	48
Clock Synchronization	48
Accuracy	49
Consider Allocation of Nodes to Network Areas	49
Which Node Types Support Which Protocols	50
Using One or Several Time Sources for The System	50
Local or Global Time Reference.....	50
Achieving Good Clock Synchronization Accuracy	51
Using External Time Source (NTP/SNTP).....	51
Network Setup and Management	52
Overview	52
Automation System Network Structure	52
Network Topologies.....	54
Choosing IP Addresses	55

Domains and Computer names	55
Building a Physical Network.....	55
Section 2 - Engineering Practices	
Distributed and Multi-user Engineering	57
Overview	57
Entities	57
Entity Types	58
Systems	59
Environments.....	60
Configure Deploy	61
Load Evaluate Go.....	62
Engineering Repository.....	62
Application Change Management.....	63
Entity Operations	64
Deploy	64
Update	64
Export/Import.....	65
Reservation	65
Multiple Control Builders on Terminal Server.....	65
Section 3 - Control Design	
Overview.....	67
Engineering Tools.....	67
Control Builder M Professional	67
Control Builder Design Considerations	68
Function Designer	70
Planning Considerations.....	71
Fieldbus Builder PROFIBUS/HART	72
Fieldbus Builder FF.....	73
Hardware Considerations	74
Implementing a Control Application.....	74
Use of Industry Specific Library Solutions.....	75

Use of Application/Project Specific Library Solutions	75
Safety/SIL Applications	77
Safety Principles	77
Management of Functional Safety	77
Hardware Consideration	77
Application Software	78
Reuse	80
Building Composite Objects	80

Section 4 - Application Design

Overview	83
Reuse	83
Planning Considerations	84
Documentation and Information Access	89
Planning Considerations	91

Section 5 - Alarm Management

Overview	93
Alarm Strategy	93
Alarms and Events	94
Alarm Generation	94
Alarm Priorities	95
Alarm Retrieval	96
Alarm Blocking	98
Remote Users	100

Section 6 - Information Management

Overview	101
System Dataflow	101
Planning Considerations for Dataflow	102
Data Collection	104
Process Data	104
Log Templates	104

Data Storage.....	105
Planning Considerations For Trend Logs (Short Term Logs).....	105
Planning Considerations For Long Term Logs	106
Event Data.....	107
Production Data.....	107
Planning Considerations for Production Data.....	107
Memory for Event and PDL Storage.....	108
Retention and Recovery	108
Connectivity Server Data Retention Online.....	108
Information Manager Server Retention Online.....	109
Archive Rates	109
Recovery Strategy	109
Archive Strategy	109
Moving Information to Secure Offline Storage	110
Making Archived Data Available	110
Archive Guidelines.....	110
Reporting Services.....	111
Report Templates.....	112
Report Destinations.....	112
Scheduling	112
Information Presentation/Desktop Services	113
Application Services/Application Data Management	114
Calculations.....	114
Softpoints	115

Section 7 - Batch Management

General	117
Batch Management Preparation	117
Planning Considerations.....	117

Section 8 - Device Management

General	121
Preparation.....	122

Decision on Protocol Type.....	122
Topology Planning	123
PROFIBUS Topology	124
PROFIBUS DP	124
PROFIBUS PA.....	127
HART Topology.....	127
FOUNDATION Fieldbus Topology	129
Field-level Topology	130
Controller-level Topology	130
IEC 61850 Connect System Topology.....	131
Installation	132
Hardware	132
Software	132
Configuration	134
Usage of Preconfigured Devices.....	134
Designation of Field Devices.....	134
Usage of Bulk Data Manager.....	135
Usage of Function Designer	136
Usage of Extended Status and Diagnosis	136

Section 9 - Connectivity to External Data Sources

Common Data Integration.....	139
Methods Overview.....	139
PLC Connect	140
Overview	140
Strengths and Weaknesses	141
Strengths	141
Weaknesses	142
When and Where to Use	142
How to Use	142
Generic OPC	143
Overview	143
Strengths and Weaknesses	145

Strengths	145
Weaknesses	145
When and Where to Use	146
How to Use.....	146
Controller as Gateway	147
Overview	147
Strengths and Weaknesses.....	148
Strengths	148
Weaknesses	148
When and Where to Use	148
How to Use.....	148

Section 10 - Maintenance Workflow Strategy

Overview.....	151
Maintenance Concept	151
Planning Considerations.....	151
User Concept	151
Workplace Concept	152
Structuring of Assets	152
Asset Optimization Reporting Requirements.....	152
Environmental Conditions	152
Asset Optimization Functionality.....	153
Asset Health Condition Monitoring	153
Planning Considerations for IT Asset Monitoring	154
Planning Considerations for Field Device Asset Monitoring	155
Planning Considerations for Process Equipment Monitoring	156
Asset Monitor Configuration	157
CMMS Integration	157
Planning Considerations.....	159
Calibration Management	160
Planning Considerations.....	160
Usage of Asset Optimization	161
CMMS System Setup and Configuration.....	161

Plan for Network Supervision..... 163

Section 11 - Commissioning the Solution

System Commissioning Considerations..... 165

- Diagnostics Collection Tool..... 165
- Control 166
 - I/O Checks 166
 - Task Tuning 167
- Logs 167
- Field Devices 167
 - Field Device Assignment..... 167
 - Upload device parameter 168
 - Check Device Parameter..... 168
 - Load OPC Server and Asset Monitor Server Configuration..... 168
 - Loop Check 169

Section 12 - System Configuration Documentation

Planning Considerations..... 175

Section 13 - Basic Concepts

Data Access..... 177

- Data Access Based on OPC 177
 - Overview 177
 - OPC Connect 179
 - DA Connector/Adapter 179
 - DA Injector 180
 - Redundancy 181
 - Failover Redundancy 181
 - Parallel Redundancy 181
 - Upload 183
 - Authenticated Write..... 183
 - Confirmed Write 183
- Configuration Change and Access Management 184

Introduction	184
Security	185
Authorization.....	186
Access Control	186
Authentication	187
Re-Authentication	187
Double-Authentication.....	187
Log Over	188
Digital Signature	188
Configuration Change Management	188
System Configuration Version	189
Synchronizing Changes Between Systems	189
Synchronizing Changes Between Environments	189
Archive	190
System and Device Checks	190
Multisystem Integration.....	191

Section 14 - Engineering Workflow

Introduction	193
Engineering Tasks.....	193
Analysis and Design.....	194
Domain Analysis.....	194
Identifying Aspect Objects.....	195
Identifying Structures.....	195
Identifying Object Types/Composite Object Types	195
Identifying Security Settings.....	195
Building of Objects	196
Overall Project Workflow	196
Build Project Environment.....	200
Create System.....	200
Load Libraries and Standards.....	201
Customize System.....	201
Build Project Library.....	202

Complete Project Library.....	202
Versioned Object Type Libraries	203
Build New or Adapt Existing Object Types.....	203
Specifications and Guidelines.....	204
Create Basic Structures.....	204
Identification of Aspect Objects	206
Importing Design Information	206
Functional Structure.....	207
Control Structure.....	207
Location Structure.....	209
Documentation Structure	209
Create Process Functions	210
Build Lower Level Functional Structure.....	210
Build Lower Level Control Structure	210
Build Lower Level Location Structure	211
Modify/Allocate Process Functions.....	211
Add Specific Control Logic	211
Assign I/O Signals	211
Allocate Process Functions to Applications in Control Structure	211
Allocate I/O Signals to Boards in Control Structure	212
Design/Build Graphic Displays	212
Finalize Graphic Displays from Object Types.....	212
Design and Build New Upper Level Graphic Displays	212
Extend Configuration.....	213
Add Production Management Functionality	213
Add Information Management Functionality	213
Add Asset Optimization Functionality	213
Finalize Lower Level Location Structure	213
Finalize Documentation Structure	213
Generate Application Software.....	214
Download and Test Application Software	214
Deploy Graphic Displays.....	214

Download Controller Code	214
Load other Run-time Configuration Data	215
Test Complete Configuration	215
Site Engineering Workflow	215

Section 15 - Configuration Tools

Automation Solution	219
Control Logic	219
Function Block/Program Programming	223
Control Module Programming	223
Diagrams	224
Functional Planning with Function Designer.....	226
Prerequisites	233
List Based Configuration	234
Documentation Printing	235
Control Module Graphics.....	235
Sequential Function Chart.....	235
Load-Evaluate-Go	236
Basic LEG Workflow	237
Further Readings	238
Control Hardware Units	238
I/O Signal Allocation	238
I/O Signal Groups	241
Topology Status Viewer	241
Process Graphics	243
Graphics Builder	244
Graphic Aspects	246
Using Object Types with Graphic Aspects	251
Display Services.....	251
Alarm and Event.....	252
Alarm Philosophy.....	254
Alarm/Event Overview.....	256
Alarm/Event Overview - Controller.....	262

Alarm and Event Buffer Configuration	265
Alarm/Event Overview - Message Logging	265
History Services Message Log Overview	266
Accessing Message Log Data	267
Guidelines for Configuring Alarm/Event Logging	268
SMS and e-mail Messaging	269
Import/Export	272
Entities	274
Dependencies	275
Resolving Errors	276
Upgrading Files.....	276
Filter Exports	276
Differences.....	276
Detailed Difference Report.....	277
Audit Trail.....	277
Recommendations for Archive Files.....	277
Configuration Using Spread Sheets	277
Data Area	279
Object Identification	280
Filter	280
Track Changes.....	280
Property References	281
Structured Properties.....	283
Formatted Templates.....	284
Preconfigured Bulk Data Applications	284
List Based Data Exchange	289
Export Data	289
Import Data.....	289
Extended Configuration Data.....	289
Parameter Management	290
General Properties.....	291
Softpoint Services	292

Calculations.....	293
Documents.....	293
Document Management.....	293
Dynamic Documents.....	294
Document Versions.....	295
Import of Document Files.....	295
Information Access.....	295
File Viewer.....	295
Bookmark.....	296
Web Page.....	296
ActiveX Wrapper.....	297
Windows Application.....	297
Help Aspect.....	297
Advantages for Object Type.....	297
Advantages for Object Instance.....	297
AUTOCAD Drawings.....	298

Section 16 - Preparing for Operation in an FDA Regulated Industry

Requirements.....	299
Overview.....	299
Production Records.....	302
National Language.....	302
Training.....	302
SOPs for 21 CFR Part 11 and cGMP Compliance.....	303
Considerations within System 800xA to Meet Requirements.....	303
Authorization.....	303
Access Control.....	303
Digital Signature.....	304
Versioning and Life Cycle Management.....	304
Audit Trail.....	304
Archive.....	305
System and Device Checks.....	305

Production Records.....	306
Training	306
SOPs for 21 CFR Part 11 and cGMP Compliance	306

Appendix A - Aspect Objects and Structures

Aspect Objects in Engineering.....	307
Objects	307
Aspects	309
Views	309
Object Types	310
Creating a New Object Instance	314
Basic Structures	315
Names and Designations.....	316
Identification of Aspect Objects	316
Additional Names	318
Name Handling	318
Reference Designations	319
Composite Object Types	321
Adding Additional aspects.....	323
Allowed Operations on Composite Instances	324
Details About Formal Instance List	324
Typical Solutions	325
Reuse Instructions.....	325
Library Handling.....	326
Library Management.....	330
Operations on a Library	330
Libraries as Distributable Units	331
Library States	332
Extension Library States	333
Entities	334
Reservation	338
Reserve/Release	338
Environments	339

Aspect Versions	341
Production-only Aspects	342
Object Structures	342
Admin Structure	344
Aspect System Structure	344
Asset Structure	345
Control Structure	345
Documentation Structure	348
Equipment Structure.....	348
Functional Structure	348
Graphics Structure.....	349
Library Structure	350
Location Structure	351
Maintenance Structure	352
Node Administration Structure	352
Object Type Structure.....	353
Obsolete Structure.....	354
Procedure Structure	355
Product Type Structure.....	355
Product Structure.....	355
Reuse Design Structure	355
Scheduling Structure	355
Service Structure	355
System Structure	355
User Structure.....	356
User Roles	356
Workplace Structure.....	357
Aspect Object Architecture	358
Client/Server Model	359
800xA OPC	360
Aspect Systems	361
System Status	362

Appendix B - Structured Plant Modeling

Introduction	365
Functional Relations Between Aspect Objects	365
Basic Object Types for Functional Structure	366
Usage of Function Reference Designations	368
Creation of Objects in Functional Structure	369
Relations Between Aspect Objects of the Control System	371
Basic Object Types for Control Structure	371
Usage of Control Reference Designations	372
Creation of Objects in Control Structure	374
Relations Between Functional Structure and Control Structure	374
Location Relations Between Aspect Objects	375
Basic Object Types for Location Structure	378
Usage of Location Reference Designations	379
Creation of Objects in Location Structure	380
Relations Between Aspect Objects for Documentation	380
Basic Object Types for Documentation Structure	381
Usage of Documentation Reference Designations	383
Creation of Objects in Documentation Structure	383

Appendix C - Designation System

Introduction	385
Function / Plant Designation (=)	386
Example of Function/Plant Designation According to IEC	386
Example of Function/Plant Designation According to DIN	386

Revision History

Index

About this User Manual



Any security measures described in this User Manual, for example, for user access, password security, network security, firewalls, virus protection, and so on, 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 is a guideline for the engineering planning to be used in the early phases of a System 800xA project. The planning part of this user manual is mainly addressed to Project Managers and Lead Engineers. However, it can be used as a source for anyone who wants an overview and recommendations for preparation and planning and a better understanding of the system.

In addition, deeper concepts, not described in the *System 800xA System Guide Functional Description (3BSE038018*)*, are included to provide information important for executing an engineering project.

Version Described in this User Manual

Unless otherwise noted, the versions of all 800xA Base System and Functional Area software described in this user manual are the latest release of 800xA 6.0.

User Manual Conventions

Microsoft Windows conventions as defined in the *Microsoft Manual of Style* are normally used for the standard presentation of material when entering text, key sequences, prompts, messages, menu items, screen elements, and so on.

Warning, Caution, Information, and Tip Icons

This publication includes **Information** icons 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:



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 the *System 800xA System Guide Functional Description (3BSE038018*)*. The listing includes terms and definitions as they 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*.

Related Documentation

A complete list of all documents applicable to the *System 800xA Released User Documents (3BUA000263*)*. This document lists applicable Release Notes and User Instructions. It is provided in PDF format and is included on the Release Notes/Documentation media provided with your system. Released User Documents are updated with each release and a new file is provided that contains all user documents applicable for that release with their applicable document number. Whenever a reference to a specific instruction is made, the instruction number is included in the reference.

Section 1 Preparation

Introduction

System 800xA is a comprehensive process automation system. System 800xA extends the scope of traditional control systems to include all automation functions in a single operations and engineering environment; enabling process plants to perform smarter and better at substantial cost savings.

The System 800xA products have been developed by incorporating information technology with the experience and know-how collected over decades of successful deliveries and customer installations.



VB based Process Graphics are not supported in System 800xA 6.0. The VB editor is not supported and VBPG graphics can no longer be edited in 800xA 6.0. It is mandatory to migrate all VB process graphics to PG2 graphics before upgrading to System 800xA 6.0.

System 800xA Hardware Requirements

System 800xA nodes require hardware or emulated hardware (Virtual Machine) that meet the minimum requirements as mentioned in *System 800xA Virtualization (3BSE056141*)*. These requirements are CPU, memory, network and disk performance. Additionally, disk and hardware redundancy features are supported with System 800xA software to improve system reliability. Not all node types have the same hardware requirements. For more information, refer to *System 800xA 6.0 System Guide Technical Data and Configuration (3BSE041434*)* to select the correct hardware for each node type.



Please see the Industrial IT Certification home page for a complete and up to date list of certified computers: <http://www.abb.com/product/us/9AAC171278.aspx>

Note that the certification itself is performed towards one or more specific versions of System 800xA.

Functional Areas

When executing an engineering project, it is very important to identify the functional software required to support the process or site. The functional software available with an 800xA System include the following:

- Base System.
- Operations.
- Batch Management.
- Information Management.
- Control and I/O.
- Safety including Safety Integrity Level (SIL).
- Engineering.
- Asset Optimization.
- Device Management.
- Multisystem Integration.

The 800xA System supports integration for the following optional controllers:

AC 800M, AC 100, Advant/Master, Advant/MOD 300, DCI, Harmony/INFI 90, Melody, Freelance, and Safeguard.

Additional supporting hardware and software components are:

- AC 800M Hardware.
- S800 I/O.
- PROFIBUS Network Components.
- FOUNDATION Fieldbus Network Components.
- HART Multiplexer Network Components.
- Device Management System for HART Instruments.
- IEC 61850 Connect.
- Integration of the Intergraph SmartPlant Instrumentation application.

Refer to the customer contract, technical requirement documents, and technical design documents to understand the process or site and the required functional software products. It is recommended to derive a list of the functional areas as well as the detailed functional and infrastructure requirements of each functional area from these documents. If you map these requirement list items to functional and infrastructure options of 800xA you will get a list of install packages to install and system extensions to load in your target (production) system and in your engineering system(s).

The following 800xA System documents provide overview functional descriptions of the products in an 800xA System, as well as technical performance data for each functional area:

- *System 800xA System Guide Functional Description (3BSE038018*)*.
- *System 800xA System Guide Technical Data and Configuration (3BSE041434*)*.

How to install, initially configure these packages and for a system upgrade is described in the:

- *System 800xA Installation and Upgrade Getting Started (2PAA111708*)*
- *System 800xA Post Installation (2PAA111693*)*.

Available functionality to implement the functional areas is described in:

- [Section 15, Configuration Tools](#).

Basic required workflow steps and additional tool or aspect system specific configuration instructions are described in the:

- *System 800xA Configuration (3BDS011222*)*.

For hardware related issues refer to the:

- *System 800xA Site Planning (3BUA000258*)*.
- *System 800xA Network Configuration (3BSE034463*)*.

All documentation related to system 800xA is available in ABB Library or ABB SolutionsBank.

Planning Considerations

1. Check project contract documents for large scale functional and infrastructure requirements.

2. Determine if Multisystem Integration functionality (operating 800xA systems from one location) is being utilized.
3. Create a list of functional areas.
4. Map them to 800xA functional- and infrastructure options.
5. Identify the needed install packages.
6. Identify the needed license quantities.
7. Plan to install the install packages.
8. Plan to do the necessary post-installation steps.
9. Continue with an investigation of detailed requirements.
10. Plan for implementation of these requirements using the engineering and configuration tools.

Using Structures

The concept of structures is central in the 800xA architecture. All Aspect Object structures are hierarchical, that is, the structures are defined by parent-child relations between Aspect Objects. Standards dealing with structural relationships between entities, such as IEC 61346 and ISA88/IE61512, are supported. Structures provided in 800xA are described in *Object Structures* on page 342.

[Table 1](#) compares the project requirement to object structure. The **Functional Structure**, **Control Structure**, **Library Structure**, and **Object Type Structure** should always be used. Other provided standard structures may also be used.

All other structures than listed in [Table 1](#) serve administration purposes or special tool purposes.

It is not recommended to “misuse” a structure even if the system might allow this.

Planning Considerations

1. Analyze project requirements from a functional perspective.
 - a. Prioritize the process function and functional planning / design perspective for engineering of control logic and/or documentation and for

Table 1. Overview Using Structures

Required in Project	Use of Structure	Explanation
Operator display hierarchy according to process functions.	Process function related objects with Graphic Displays in Functional Structure .	Functional plant design is done in engineering. A plant is typically operated from functional perspective.
Diagram based engineering, documentation and maintenance of the process control application according to hierarchy of process functions.	Process function related objects with Function Diagrams in Functional Structure . Allocated to control objects of Control Builder M in Control Structure .	Application is designed, configured, tested, documented, and maintained in Function Diagrams* from functional perspective (based on functional plant design) by use of Function Designer. * Nested networks of interconnected block symbols, including engineering signals and field devices.
Controller configuration based engineering, documentation and maintenance of the process control application.	Control objects (Control Network, Control Project, Controllers, Hardware Units, Control Application, Tasks, Programs, Control Modules, Function Blocks, Diagrams) of Control Builder M in Control Structure .	Basic Control Structure hierarchy is predefined by the Control Builder M. Application is designed, configured, tested, documented, and maintained from control perspective by use of Control Builder M.
Structured access to location information of plant entities for maintenance.	Objects representing plant entities in Location Structure.	Locational plant design is done in engineering. For example, Location Structure can be used to present the locational assignment of Process Functions, Controller or Hardware Unit objects to areas, buildings, floors, rooms, coordinates, and even mounting locations.

Table 1. Overview Using Structures (Continued)

Required in Project	Use of Structure	Explanation
Structured access to asset monitoring information.	Objects equipped with asset monitoring placed in Asset Structure.	Asset Monitoring planning is an engineering activity. Asset Structure and Maintenance Workplace can be used for optimized access to Asset Monitoring information. Requires Asset Optimization option.
Structured access to plant documentation.	Objects representing plant documentation entities in Documentation Structure.	Documentation planning is an engineering activity. Documentation Structure can be used to present and structure the plant and system documentation.
Own template-type aspects.	Objects with such aspects in Library Structure .	Examples: History Log Templates, Trend Templates, Document Manager Templates,
Own versioned object type libraries or control object type libraries.	Library and library version objects in Library Structure , library version and object type objects in Object Type Structure .	Example: AC 800M Connect object type libraries.
Own object types (not contained in versioned libraries).	Object group and object type objects in Object Type Structure .	Example: CBM_Signals Object Type Group.
Own object types with variants.	Design in Reuse Design Structure, object group and object type objects in Object Type Structure.	Variants selectable and parametrizable during instantiation. Requires option Reuse Assistant.

maintenance: Plan for application implementation with Function Designer in Functional Structure.

- b. Prioritize control application/program and task perspective for configuring control logic, documentation and for maintenance: Plan for application implementation with Control Builder in Control Structure.
- 2. In all cases plan for Graphic Displays in Functional Structure.

3. Process Functions, Controller or Hardware Unit objects (and possibly others) shall easily be identified by their locational assignment. Engineer a Location Structure to the required depth to meet this requirement.
4. Objects representing plant assets shall be grouped and easily accessible. Engineer an Asset Structure with the required grouping and depth to meet this requirement. Requires Asset Optimization option.
5. Plant documents shall be accessible easily as a whole structured collection. Engineer a Documentation Structure to the required depth to meet this requirement.
6. Reusable entities (Libraries/Object Types/Templates) have to be designed. Engineer these entities in Library Structure/Object Type Structure to meet this requirement.
7. Object types with variants have to be designed. Consider engineering them in the Reuse Design Structure using Reuse Assistant.

Using Names and Designations

A name typically identifies an object without taking care of the hierarchy level in the structure the object resides. Whereas a designation typically identifies the object based on references to the hierarchy levels. The relative reference designation is identifying a single hierarchy level and the absolute reference designation provides an hierarchical name concatenating the reference designation of the levels above and of the level the object resides on.

While smaller projects typically use names, larger projects sometimes use designations additionally, especially in highly standardized application areas.

When an engineering project is started there can be customer requirements and/or ABB project execution standards for how to name or designate objects in the plant. Even if there are no requirements the recommendation is to define a good naming and/or designation standard for executing the project.

Plant Explorer supports the two different concepts:

1. Identification of Aspect Objects by name:
This is the default using the Name aspect. Basic recommendation is that you create unique names. If no other requirements exist you can for example follow the naming conventions described in the *System 800xA Control AC 800M Planning (3BSE043732*)*. This proposal is based on ANSI/ISA S5.1 standard.

2. Additional identification of Aspect Objects by reference designations kept in own Designation aspects and optional display of the absolute reference designation: This is the concept, if the customer requires reference designations, for example according to IEC 61346. IEC 61346 references further standards such as IEC 61175, ANSI/ISA S5.1, IEC 60750, IEC 61082-1, ISO 3511-1 and -2, ISO/DIS 1219-2. and so on. Basic recommendation is that at least the combination of name and designation shall be unique.

Refer to [Names and Designations](#) on page 316 to read more.

Planning Considerations

1. Check customer's naming and designation requirements.
 - a. If existing: Evaluate standards required for naming and/or designation.
 - b. If none: Naming and/or designation according to ABB project execution standards or own best practice.
2. Evaluate required visibility of names and/or designations in structures, graphic displays, faceplates, Function Diagrams, dynamic documents and project documentation.
3. Reduce use of additional Name and Designation aspects to the required minimum.
4. Design and describe naming and designation conventions to use in project.
5. Implement project specific additions to naming and/or designation functionality given in the system by default.

User Strategy

When planning a user strategy for your system, key elements like user accounts, user groups and permissions are critical. The administration of these elements is handled in Plant Explorer and control personnel authorization and access control.

To be able to work in the system each unique user has to be assigned to one or several 800xA user groups, then permissions control actions and restricts access for groups or specific users. The general principles when configuring 800xA system security is to keep it as simple as possible and it is recommended to base the system on unique user accounts but to keep the administration of 800xA permissions and roles on user group level due to faster and easier administration.

User Groups

In a planning phase the two most relevant user groups are Operators and Application Engineers. These roles and permissions are stated in [Table 2](#) below:

Table 2. User Groups and Permissions

User group	Role	Permissions
Operators	This role performs process operations but do not tune and configure. The traditional operator functions like the process graphics, alarm and event lists, trend displays, history logs are available. The system allows the operator to have direct access to documentation, standard operating procedures, drawings etc. depending on how the process application is configured.	Operate and Configure
Application Engineers	This role allows a user to do engineering operations and work with objects and applications built for objects (programs, graphics etc.) This role tunes the process and performs application configurations.	Tune and Configure

Permissions

Once users are assigned to user groups they are given permissions which control their actions. Permissions can be set for users, user groups, a structure or part of a structure, or an aspect object. Permissions can also be set on nodes.

An important principle is to not give extended permissions to a whole user group just because a few users need it. For example, if some operators are given extended permissions, say Tune permissions or permissions for Emergency Shutdown for a specific valve or tank, it is recommended to create another user group for these operators, instead of giving all operators in the existing group extended permissions.

When a permission is set on node a user can be given different authority depending on her/his location. This method can be used to control operations which require to be performed from nodes within visible views of the area they control.

Planning Considerations for User Groups and Permissions

1. Start by reading the general security planning guidelines given in the *System 800xA Administration and Security (3BSE037410*)*.
2. Specify operators and engineering project members, for example their tasks, access rights and restrictions.
3. Specify which nodes the users have to use. For Engineering, think of Single Engineering Nodes, nodes of multi-node Engineering System and nodes of Production System.
4. If more than one node: Decide whether to use a domain or a workgroup. A workgroup is recommended due to a low cost entry point. For system solutions containing more than nine nodes a Domain has to be used.
5. Introduce Windows users/user groups on the nodes, depending on what kind of node it is: Single node, node of a domain or node of a workgroup.
6. Assign users/user groups to the groups IndustrialIT User and/or IndustrialIT Admin.
7. Assign 800xA user roles to these users/user groups in the created 800xA systems.
8. Use the built in tools to set permissions and restrictions. Do not give users privileges to more than they need to perform their actions. Do not give users Administrator rights unless absolutely necessary since the security is turned off for this role. Use guest accounts for temporary users.
9. If none-default access rights are needed:
Create and configure Security Definition aspects to implement these none-default access rights.

Protection of Engineering Work

An additional topic is the protection of engineering work packages against unintentional or intentional corruption by some user having access to the engineering system.

1. Use the basic mechanism to create and configure Security Definition aspects to achieve work package protection. At least the objects/top objects for the work packages must exist to hold the Security Definition aspect.

2. Specify users' access to nodes and work packages. Be aware of that by default for example a user with the Application Engineer role typically can configure all work packages on all nodes.
3. User group assignment can also be used to limit access to engineering work packages covering objects or object substructures to certain engineering users or user groups.

Re-authentication and Double-authentication

For process critical operations advanced authentication operations like Re-authentication and Double-authentication can be used.

- Re-authentication is used to guarantee that an operation is performed by the operator currently logged in.
- Double-authentication is used for operations critical to regulations or product quality. When Double-authentication is used an operation has to be verified by a second operator before a change is approved.

Refer to [Re-Authentication](#) on page 187 and [Double-Authentication](#) on page 187 for more information.

Audit Trail

The system supports Audit Trail which means that the production process can be logged. The logs can be stored and archived and viewed at a later stage.

When planning for audit trail it is important to decide what level of granularity to use.

Archiving of the Audit Trail is accomplished by transferring the audit events to Information Management, in the form of OPC Events. When the system message server receives a new audit event it converts it to an OPC event and sends it to the Information Management History Services OPC message log. During the creation of the OPC Event some information is expanded from the internal format stored in the system message server into plain text information. For instance the UserId is converted into a string with the users Windows login name.

The Information Management History Services has archive facilities for long time storage of the Audit Trail. For more informations, refer to the *System 800xA Information Management Configuration (3BUF001092*)*.

As the Audit Trail is stored as system messages the Alarm/Event viewer aspect can be used as a basic viewer to display audit events. The alarm/event viewer can only display a subset of the information available in an audit event. Information that must be visible in the alarm/event viewer must be formatted into the message column.

Performance and Capacity

The 800xA System is scalable both in functionality and size. The rules for how the system can be configured depend on how functionality is combined in different node types, and on the size of the system and application. Different system functions are provided by different node types in an 800xA System installation. A node in this context is a workstation that has a network address on the 800xA system network, is used as a server, or is used as a client.

The *System 800xA System Guide Technical Data and Configuration (3BSE041434*)* is a key document when planning system 800xA performance and capacity. The document contains, for example, performance and capacity data, configuration and dimensioning rules, system configuration examples and technical data for Control Software and Control Builder M key functions. It contains valuable information for different project phases like System Design, Engineering and Commissioning.

The document is available in ABB Library or ABB SolutionsBank. It is also included on the System 800xA software media together with release notes.

Planning Considerations

1. Check project requirements for performance requirements.
2. Verify if performance requirements can be fulfilled in principle by checking against information given in the *System 800xA System Guide Technical Data and Configuration (3BSE041434*)*.
3. Read corresponding system documentation for information on how to design and configure hardware and software system configuration for performance.
4. Check corresponding engineering and configuration documentation for information on how to design application for performance.

5. Consider how to achieve early performance tests (under application load)
 - a. of sub-systems.
 - b. and the whole system.

External Data

What data is available from the customer - instrument lists (device lists, etc.), INtools v6.x database, SmartPlant Instrumentation v7.x database, etc.

- Excel based instrument and device data lists
 - Is the data organized by tag (single I/O channel) or control loop (multiple I/O channels plus loop type)?
- Intergraph INtools v6x database
 - Is the database Oracle 11 or Microsoft SQL Server 2008? (Sybase is not supported by 800xA).
 - Was the database configured using 800xA hardware as target?
- Intergraph SmartPlant Instrumentation v7.x database
 - Is the database Oracle 11 or Microsoft SQL Server 2008?
 - Was the database configured using 800xA hardware as target?
 - Was the ABB S800 I/O Definition file used?

Libraries

Libraries can advantageously be used when:

- An application contains specific type solutions that are reused in several instances within the project.
- The type solutions can be reused in other projects.

Libraries can have versions, be packaged as system extensions and contain types for hardware as well as types used for applications. When working with libraries it is important to be aware of how versions are handled, how libraries are connected and how they are affected when changed.

Detailed information about the library concept and how to build and manage a library structure for your organization is found in the *System 800xA Control AC 800M Binary and Analog (3BSE035981*)*.

Standard Libraries and Templates

The System 800xA standard libraries contain a broad range of objects with different functionality. They contain ready-made objects that simply have to be connected to your environment to work and they contain objects that can work as templates. Existing Standard Libraries cover the needs for several areas and purposes like, for example Alarm and Event, Control, Batch and Communication.

By using templates, libraries process objects can be modified to fit specific requirements. The following libraries include process objects which can work as templates:

- BatchAdvTemplatesLib
- BatchLib
- ControlFuzzyLib
- ControlObjectLib
- ControlSolutionsLib
- GroupStartLib
- ProcessObjBasicLib
- ProcessObjExtLib
- SeqStartLib
- SupervisionBasicLib
- ProtectionExampleLib

These libraries and the commonly used standard libraries are briefly described in the [Section 3, Control Design](#). For detailed information on libraries and templates refer to the *System 800xA Control AC 800M Binary and Analog (3BSE035981*)*.

Use of Industry Specific Library Solutions

In some cases, specifically developed libraries for different industries and customers extend the 800xA standard library functionality. For example there are specifically developed libraries for industries like:

- Pulp.
- Metals.
- Building materials.

Usually these library solutions comes with their own guidelines on how to apply the library objects in the 800xA system.

Please contact ABB for more information about specific libraries and how to make them available.

Predefined Workplaces

Several Workplaces are available in 800xA:

- Operator Workplace.
- Plant Explorer Workplace.
- Multiscreen Workplaces and Extended Operator Workplace.
- Engineering Workplace.
- Maintenance Workplace.

The Operator Workplace, the Plant Explorer Workplace, the Multiscreen and Extended Operator Workplaces are default in the system. The Engineering Workplace is available if Engineering Studio is used. The Maintenance Workplace is available if Asset Optimization is used. The different workplaces can either be used as they are or configured to suit special needs.

Operator Workplace

The Operator Workplace ([Figure 1](#)) is the environment from which the operator views and controls the plant process. The workplace can be configured per user or user group, and have different or identical layouts according to the users' preferences.

It can have a single- or multi-screen setup. In the multi-screen setup, one screen can be dedicated to always show alarms and another to always present the trend display. The operator keyboard can be a standard computer keyboard with a mouse and some additional, dedicated hot keys for fast direct actions.

More information about the Operator Workplace and operator workplace configuration is found in the *System 800xA Operations Operator Workplace Configuration (3BSE030322*)*.

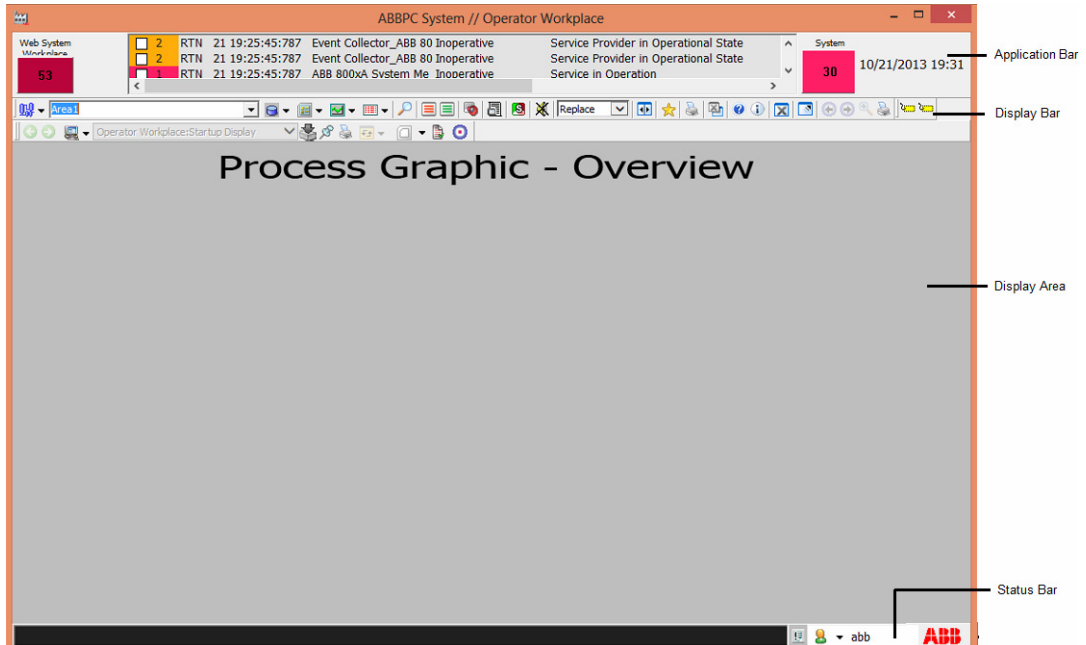


Figure 1. Operator Workplace

Planning Considerations

When planning your Operator Workplace decisions on how your process is to be controlled, how information is to be presented and which tools to be used, has to be made. It is recommended to always create a new Operator Workplace and configure it to meet your needs instead of copying and modifying an existing one.

Consider the following before configuration:

- How will the monitor(s) be used.
- Which tools are needed for operator control.

- How many, and how large displays should be used.
- Will the default color settings be used or will you create your own.
- Will default Alarm- and Event lists be used or will you create your own.
- How will operators navigate in the system.
- Which Workplace will be the default workplace in the default system.
- Will you receive information from other systems.

Plant Explorer Workplace and Engineering Workplace

The Plant Explorer is the system main tool for exploring and building hierarchically structured models of a plant or a system as well as to create and configure the objects and aspects of these structures. The Plant Explorer is also the main tool for Application Engineers for configuration of the Operator Workplace.

The appearance of the Plant Explorer user interface is configurable and depends on the security settings and user roles. For details about Plant Explorer Workplace refer to the *System 800xA Operations Operator Workplace Configuration (3BSE030322*)*.

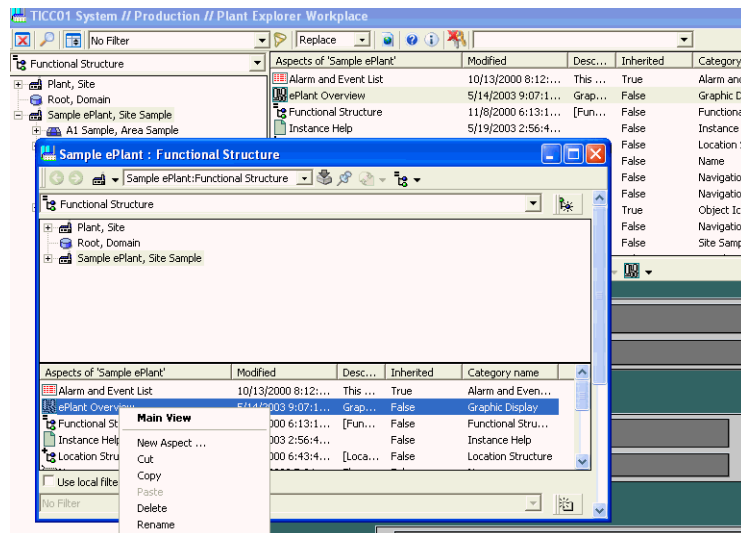


Figure 2. Plant Explorer Workplace

The Engineering Workplace (Figure 2), which is delivered with Engineering Studio, is a special configuration of Plant Explorer Workplace and provides additional engineering functions. For more information on the Engineering Workplace refer to the *System 800xA Engineering Engineering Studio (3BDS011223*)*.

Maintenance Workplace

Asset Optimization information can be accessed from any workplace in the system. The Maintenance Workplace is the default Workplace for Maintenance personnel. It is basically a Plant Explorer Workplace with an Alarm Band that shows Asset Monitoring Alarms for default Asset groups.

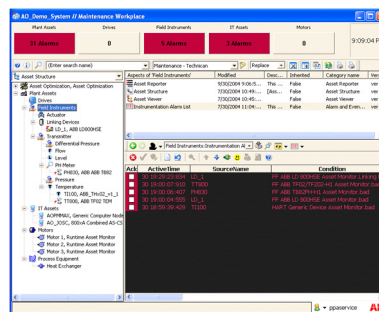


Figure 3. Maintenance Workplace

The Asset Structure provides the possibility for maintenance personnel to group and arrange plant assets in the way it is needed for efficient daily work. Control topology constraints can be overcome and even devices with different fieldbus protocols can be grouped together if they require similar maintenance procedures.

The Asset Structure comes by default with five major groups of Assets:

- Drives.
- Field instruments.
- IT assets.
- Motors.
- Process equipment.

The typical structure to work with in the Maintenance Workplace is the Asset Structure (refer to the following figure). The Asset Structure shows, by default, five groups of plant assets. The Field Instrument group has subgroups for Actuators, Linking Devices and Transmitters. The Transmitter group is further structured to allow a unique group for the different transmitter types.

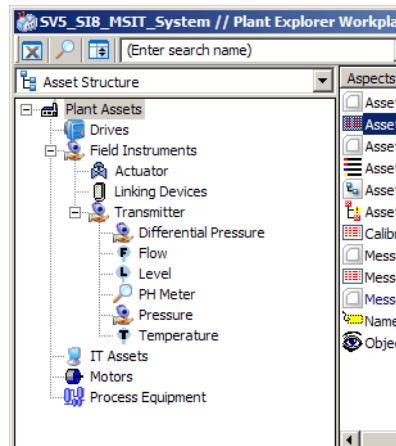


Figure 4. Asset Structure

Planning Consideration

When planning your Maintenance Workplace consider the following:

- Are the default Asset Groups sufficient for efficient maintenance management? If not, rename or arrange the plant assets in the Asset Structure for efficient maintenance management.

For more information on Maintenance Workplace refer to the *System 800xA Asset Optimization Operation (3BUA000150*)*.

Navigation Strategies

There are several ways for users to navigate in the system. The system interface is based on a context environment. The roles and permissions as well as the configuration of an object automatically defines the possible selections available in

the context menu. Engineers might for example have access to configuration functions, while operators have access to functions for process control.

The system supports tools for quick access to displays and information. The navigation tools listed in [Table 3](#) can be considered:

Table 3. Quick Access Tools

Access tool	Description
Hot Keys	Hot Keys provide the ability to map key strokes or key stroke combinations to perform an action available to a selected object. The mapping of keystrokes is user configurable. The Operator Workplace includes default mappings for key actions such as Alarm Acknowledgement.
Favorites	The Favorite aspect system is a function similar to the Favorites function in Internet Explorer. By using Favorites the most visited aspects can be grouped and organized for fast access.
Shortcuts	Shortcuts make it easy to navigate to important and frequently used aspects of different aspect objects.

For more information about context based navigation and quick access tools refer to the *System 800xA Operations Operator Workplace Configuration (3BSE030322*)*.

A common base for the navigation is to create the process areas in a suitable hierarchical structure in the Functional Structure then assign users or user groups or operators to the process areas.

Planning Considerations for Basic Navigation

Consider the following steps to setup the navigation for a workplace:

1. If the workplace is using two screens, use one of them for Process related information and the other for System related information.
2. Preconfigure the “Object Toggle browser” to show Functional Structure as default in order to easy see the process area hierarchical structure.
3. Use Hot Keys. To get direct access to up to 12 process area overview displays, use Ctrl + F1 - F12 as Hot Key that gives direct access to the major process area overview displays.

4. Create the same navigation possibilities from the display bar. Place the same 12 process area overview display shortcuts in the display bar. These up to 12 icons (can be a different icon per process area) are directly accessible from the display bar. By using display shortcuts that are inherited from the functional structure (plant object aspect has all the 12 display shortcuts), it is possible to configure the display shortcut aspect to support inheritance.
5. The display shortcuts will enable drop down aspect lists for each of the process areas. They will always appear on the displays in the same order and have unique icons. Unique icons if that is defined as an override.
6. Use the filter definition capability and define different filter content for different user groups like engineers, operators etc. or for different users depending on their need. The operator wants in most cases to have access to graphic displays, alarm lists, trend displays, etc. Start the definition by increasing/decreasing the content in the process operation filter that has to be assigned to each user or user group using it.
7. For viewing information from other process areas (similar to next page/previous page function) place aspect links in the graphic display to follow the process flow forward or backward.
8. Use display shortcuts in the display bar instead of numeral aspect links in the graphic displays. Aspect links take graphics display area space and if the application is redesigned they require a lot of reconfiguration when changing. With display shortcuts, the configuration changes are minimal by using the inheritance feature.

Additional Navigation

- Additional to the navigation recommendations mentioned previously, each operator or operator group can have a set of favorites preconfigured. The favorites can be the most used graphics, the operator instructions, documentation etc.
- In the application bar, additional alarm bands can be configured for the process areas. If the plant is divided between operators or operator groups, the alarm band content will vary. A way to solve this on a user/user group basis is to

define unique workplaces per user group or user. Start from the default provided Operator Workplace.

- The application bar can provide for a fixed display area. This area can be used for showing the most important process values for the controlled process or aspect links to critical status information.
- The tabs are automatically created using the object names and are used to navigate to the corresponding graphic display. If a tab object includes child objects, these appear as a dropdown list within the tab, and the user can navigate to the required object. The Tabbed Workplace also includes Status Indicators and a Breadcrumb list for each tab. For more information on the tabbed workplace and tabbed navigation, refer to *System 800xA Operations (3BSE036904*)*.

Security Strategy

The purpose of this paragraph is to discuss possible security measures that a user of an automation system may want to consider, based on a risk assessment for a particular application and installation. The described measures are not necessarily complete or effective for a particular application and installation.

The 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 automation system.

Information System (IS) Security

Information system (IS) security measures aim at protecting the confidentiality, integrity, and availability of a computer system from being compromised through deliberate or accidental attacks. This is accomplished by implementing and maintaining a suitable set of controls to ensure that the security objectives of the organization are met. It is recommended that these controls include policies, practices, procedures and organizational structures, as well as software and hardware implemented security functions.

There is no single solution or technology for IS security that fits the needs of all organizations and applications. The security measures applied to a specific

installation should be proportional to the assessed risk in terms of probability of a successful attack and the potential consequences.

For automation systems in particular, the impact of an attack may include, for example, endangerment of public or employee safety, violation of regulatory requirements, loss of proprietary or confidential information, loss of production, damage to equipment, and loss of public confidence.

100% security is not possible in an interconnected environment. Even a network that is perceived as being totally isolated may be vulnerable to security from different sources, such as the occasional connection of portable computers or unauthorized installation of software.

Security Policy

It is recommended to establish an adequate IS security policy at an early stage. This may be based on an analysis and assessment of structures and information and control flows, probability calculations, potential consequences as well as available technical security solutions.

A security policy should also include plans for regular audits of the IS security, for training of personnel and partners, and for incident response, including how to recover from potential disasters. The distribution of responsibilities between different parts of the organization should be defined. An efficient security administration, with enforcement of good user practices as well as regular implementation of vendor recommended updates for operating systems, application software, and security related software, is also required.

The security policy should be based on the principle of least privilege and compartmentalization. Every application, user, or subsystem should be restricted to the minimum number of rights for the minimum number of resources that is necessary to fulfill its purpose. Network access to functions that are not explicitly required should be disabled. This reduces the possibilities that an attacker can exploit, and limits the damage in case an intrusion attempt is successful. For mission critical systems, such as manufacturing and control systems, it may be appropriate to entirely prevent the use of functions that are known as common infection routes, such as e-mail, instant messaging, and Internet browsing.

Network Security Zones

A common approach to network security is based on a layered zones of trust to provide increasing levels of security according to increasing security needs. Each zone is inside the next, leading from the least trusted to the most trusted. Connections between zones are only possible through secure interconnections, such as a firewall or a system of layered firewalls (Figure 5).

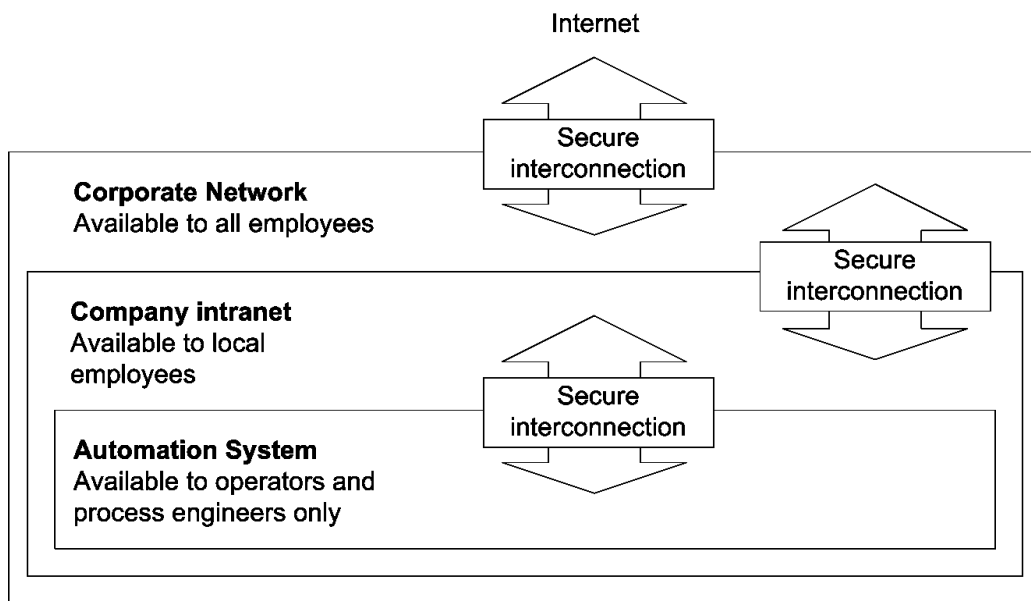


Figure 5. Network Security Zones

For more information on IS security considerations for automation systems, refer to the *Security for Industrial Automation and Control Systems (3BSE032547*)*. For specific network issues check the Network paragraph below in this section, or refer to the *System 800xA Network Configuration (3BSE034463*)*.

Use cases for Connections via Firewalls

From a network security point of view the easiest configuration to maintain is an isolated Automation System Network which is not connected to any other network.

However there are a number of cases where it is necessary to connect the Automation System Network with some other network:

- Remote/external client
The 800xA system needs to be accessed from a user on a computer which is not connected to the automation system network. This means that the Automation System Network needs to be connected to an external network and the client functionality needs to be made available through some kind of firewall system.
- Site to site connection
The automation system is located on two or more geographical sites and the remote connection needs to go via an external network.
- Integration with external 3rd party system
Data in the 800xA system needs to be accessed from a 3rd party system on an intranet. This can be done in some different ways where the intranet and the Automation System Network are connected in some way.

These use cases are described in the *System 800xA Network Configuration (3BSE034463*)*.

Planning Access Through a Firewall

Setting up a firewall system should be done based on a security policy as described above in the Security paragraph.

For the actual configuration of the firewall system there are a number of details to consider that are specific for each system. The following list describes some common questions to consider:

- Which services need to be accessed through the firewall?
This is typically a question of which port numbers to open access for. Refer to the *System 800xA Network Configuration (3BSE034463*)* for information about port numbers for 800xA system functions.
- Which nodes on the Automation System Network need to be accessed?
- Which users need to access through the firewall and which user rights do they need on the inside?
- Which computers on the intranet need to access through the firewall?
- Are there specific times that the access needs to be allowed and other times when it can be blocked?

- What addresses are used on the intranet and what addresses are used on the Automation System Network?
- Is it possible to initiate all traffic through the firewall from nodes on the Automation System Network or is it necessary that a node outside the firewall reads data from or sends data to the inside?
It is more secure to configure a firewall only allowing accessing from the inside to the outside.
- How much data needs to be transferred through the firewall?
Throughput (Mbps)?
Number of concurrent connections?
It may be decide which type of firewall to use.

The recommended approach is to restrict everything which is not explicitly known to be needed. Block access for all nodes, users, services etc. except the listed ones.

Life Cycle

The software for a system version is actively maintained as long as the system version is actively sold - the version is in an active phase. When a new version is released, the previous version is still supported for a number of years. This means that critical errors will be corrected, service packs may be planned, and Microsoft security updates will be verified.



An updated version of the AC 800M controller firmware, no matter how it gets updated (major revision, minor revision, service pack, rollup, or temporary correction), requires that a singular controller be shut down. In redundant controller situations, the online upgrade feature can be used to apply updated firmware versions while keeping the process running.

For more information on Life Cycle policy, refer to *System 800xA Life Cycle Policy (3BSE042159)*.

Clock Synchronization

A plan for how to synchronize the real time clocks in different nodes in the system should describe:

- What to use as time source(s). It may be one or several nodes in the system or some external unit or units.

- Which protocols to use to distribute the time from the time source(s).
- The time synchronization flow through the system from time source to each node in the system.
- How to verify that the planned synchronization works as intended.

It is important to define the requirements for the time synchronization early in the planning phase.

Accuracy

One of the main requirements is which accuracy is needed throughout the system. Typically this is the same as asking: How small time differences are important to be able to securely detect if data or events are time stamped in different nodes?

The requirements of the accuracy may be different between different nodes. Often the nodes can be divided into groups with similar requirements. An example of groups of nodes that may have different time sync requirements between the groups but similar requirements within the groups are:

- Controllers with SOE devices.
- Other controllers.
- Connectivity servers with history logging.
- Other Connectivity servers.
- Other servers.
- Clients.
- Domain Controllers.
- Switches supervised with SNMP.
- Other equipment in the system.

The rest of this section gives some advice about how to choose between different alternate solutions.

Consider Allocation of Nodes to Network Areas

The requirements are typically higher between nodes within the same process area than between nodes in different process areas. Consider this when planning allocation of nodes to different network areas.

Which Node Types Support Which Protocols

The available alternatives for how to distribute the time through the system depends on which clock synchronization protocols the different node types support. For details refer to the *System 800xA Network Configuration (3BSE034463*)*.

Using One or Several Time Sources for The System

Is it more important to have accurate clocks in any specific part of the system? Could that part of the system act as time source for the rest of the system? If this is not the case it might be a good alternative to use a global time source in each area.

One reason to use several time sources may be that it improves the system availability by making different parts of the system less interdependent. Consider if it is possible/acceptable/sufficient from accuracy/reliability/security point of view to synchronize between different areas.

Local or Global Time Reference

Consider if the clocks in the system need to be synchronized with global time or if it may be sufficient with a local time reference for the system.

Reasons to choose to use a global time source may be that:

- It must be possible to compare time tags from the system with time tags from other systems in the world.
- It is not possible to define one node in the system from which it is possible to synchronize all the other nodes with a good accuracy.
The reason may be, for example, that the communication paths may include too many router hops or that it is not desired to set up communication paths throughout a large system from a security point of view.
Another reason may be that there are nodes in the system for which there is no clock synchronization protocol that they both support.
- You want to avoid completely the needing to manually adjust a local time source. Even if it would be done only once a year it might lead to problems.

Achieving Good Clock Synchronization Accuracy

To achieve a good time accuracy in the whole system, the best method is normally to distribute the time “upwards” in the system, from the Control Network to the Client Server Network. The protocols on the Control Network support a higher accuracy and the implementations of the real-time clocks in the Controllers are better than the protocols used between workstations and the Windows time.

It is often a good idea to let a controller act as time master for the Control Network, then the rest of the system will be synchronized from this network. Details about how to synchronize the rest of the system from the controllers in a control network are described in the *System 800xA Network Configuration (3BSE034463*)*.

In a system with several network areas it might not be possible to define a path “upwards” in the system which covers the whole system. This can for example be a system with two control networks connected to one Client Server Network. For such a system there are two main alternatives:

1. Let one control network be the time source for the whole system and let the other control network fetch the time via routing.
2. Use an external time source in each part of the system.

The first alternative is probably cheaper in most cases but the accuracy is probably better with the second alternative. Using external time sources on several parts of the system also means that the system is less dependant on routing paths between the different parts which always have to be up and running.

Using External Time Source (NTP/SNTP)

In most cases the recommended method for synchronizing nodes in an 800xA system with a global time source is to use an NTP (or SNTP) server with a GPS receiver.

NTP (Network Time Protocol) and SNTP (Simple Network Time protocol) are internet standards. SNTP is a subset of NTP. The difference between SNTP and NTP is not important in the 800xA system. The NTP/SNTP protocols can be used via routers but communicating through several network devices decreases clock synchronization accuracy.

To achieve the best accuracy the time server should be connected to the system network as close as possible (that is with few network devices in between) to the nodes acting as clients to it. The AC 800M controllers and computers running Windows support the SNTP/NTP protocol.

For Controllers that do not support SNTP/NTP some other methods may be used. Some controllers support a minute pulse. Check the documentation for the particular controller type to find out the available options.

Network Setup and Management

Overview

The Automation System Network is for communication between *workplaces*, *servers* and *controllers*. It is a local area network (LAN) optimized for high performance and reliable real-time communication with predictable response. Servers run software that provides system functions. Workplaces run software that provides various forms of user interaction. Controllers are nodes that run control software.

Fieldbuses are used to interconnect field devices, such as I/O modules, smart sensors and actuators, variable speed drives, PLCs, or small single loop devices, and to connect these devices to the system, either via controller, as indicated in [Figure 6](#) below, or directly to a server.

The Automation System network should be a trusted network zone which is not directly connected to any external network. Connections may be done via firewalls.

Automation System Network Structure

The Automation System Network for System 800xA is based on Ethernet and one or several TCP/IP networks. The networks can be made redundant using the routing protocol RNRP (Redundant Network Routing Protocol).

[Figure 6](#) shows the network topology with a Client/Server- and Control Network. The three types of network are briefly described below.

- The Plant Network can be dedicated for process automation purposes or be a part of the plant intranet already available on a site. The Plant Network can be, but is normally not made redundant with RNRP.
- The Client/Server network is used for communication between servers, and between client Workplaces and servers. Due to security and performance reasons normally only certified 800xA products are connected on this network.

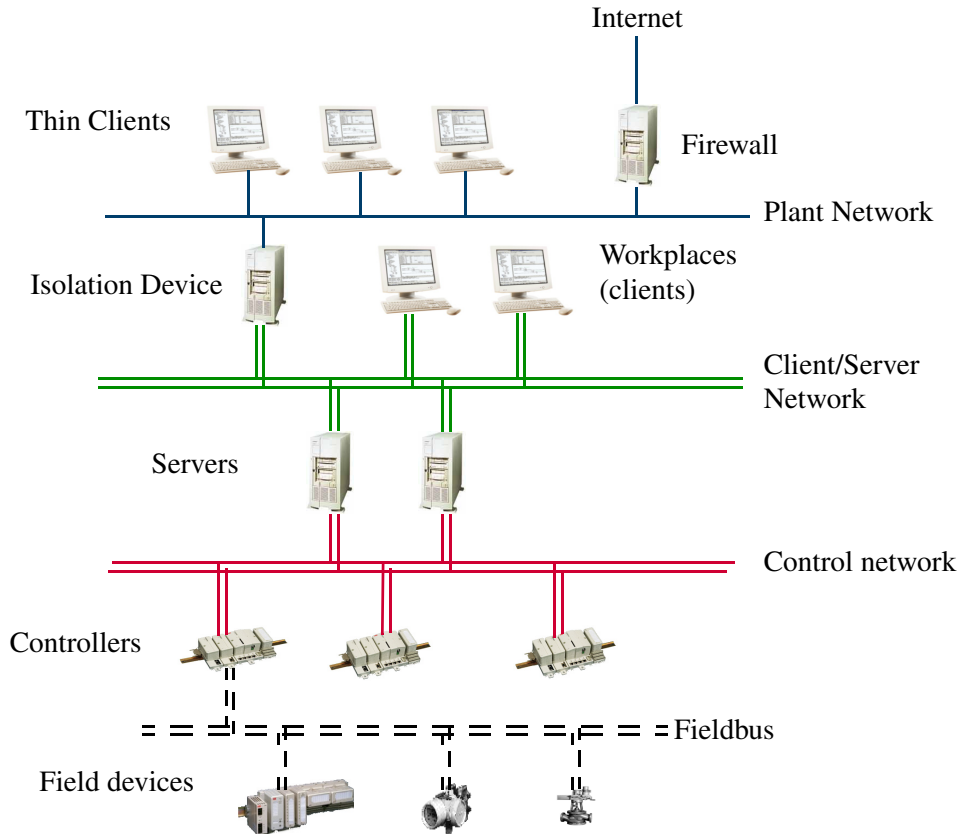


Figure 6. Network Topology

- The Control Network is a Local Area Network (LAN) optimized for high performance and reliable communication, with predictable response times in real time. Controllers and Connectivity Servers are connected to the control network.

The Control Network is based on Ethernet using the MMS protocol on top of a TCP/IP protocol stack, plus additional services for time distribution, redundancy features etc.

For safety and performance the Control network should be kept as isolated as possible from all traffic that does not belong to controller products. Other traffic may jeopardize performance.

Only 800xA certified products should be connected on the control network. It is recommended that separate networks are built for Ethernet based fieldbuses for example, FOUNDATION Fieldbus HSE or TCP/IP based communication with the INSUM System.

Network Topologies

The Automation System Network can be built using different topologies. When planning the topology consider the following:

- How many 800xA nodes will there be in total in the system?
- How many controllers will there be in the system?
- How many connectivity servers will there be in the system?
- Is a redundant network needed?
- Will the nodes be located with large distance in between them?
- Which nodes will communicate heavily with each other?
- Which nodes will be placed together?

Typical topologies used for the Automation System Network like:

- Combined Client/Server Network and Control Network.
- Separated Client/Server Network and Control Network.
- Using multiple Control Networks.

are described in *System 800xA, Network Configuration (3BSE034463*)*. This manual also describes common types of distributed network solutions like for example Remote Clients, Extended Control Network, Connecting Clients close to remote controllers.

Choosing IP Addresses

All nodes on the Automation System Network use static IP addresses. It is recommended to use private addresses. The recommended network ID for the system network is 172.16.x.x/22 (a subnet mask of 255.255.252.0). Refer to the *System 800xA Network Configuration (3BSE034463*)* for a recommended set of addresses to use for all nodes in the 800xA System. Using the recommended addresses reduces the configuration effort. It is possible to use other addresses but that means that more data has to be configured for each node.

Refer to the *System 800xA Network Configuration (3BSE034463*)* for considerations on how to choose IP Addresses in case the Automation System Network is connected to another network, for example the recommended usage of Network Address Translation (NAT).

Domains and Computer names

Refer to the *System 800xA Network Configuration (3BSE034463*)* for things to consider when planning how to allocate 800xA systems to domains and how to choose names for domains and workstations.

Building a Physical Network

Refer to the *System 800xA Network Configuration (3BSE034463*)* for things to consider when planning the physical installation of the Automation System Network. For example there are recommendations regarding:

- Which network equipment to use, for example switches and cables.
- Which ethernet speed to use.
- How to allocate nodes to switches.
- How to build redundant networks.

Section 2 Engineering Practices

Distributed and Multi-user Engineering

Overview

The 800xA System SV 5.0 and later supports Distributed and Multi-user engineering. Different Engineering Practices are possible. A general workflow is described in [Section 14, Engineering Workflow](#).

For this discussion the following three concepts are central:

- Entities - an entity is a collection of objects and aspects that is treated as a unit.
- Systems - different systems allow local distribution of self-sufficient sets of engineering workstations.
- Environments - a separate engineering environment allows engineering changes to be performed and integrated in an environment isolated from the production environment.

Entities are always supported, while multiple systems and support for environments are selectable. Selecting support for environments is irreversible. The only way to deselect it is to delete the system and create a new one.

An overview of the operations that can be performed on entities is also described in this section.

Entities

Previous versions of System 800xA have supported Aspect Objects and Aspects as the units that a user works with in engineering. To accommodate for the improved multi-user and distributed engineering features, the concept of entities was added in SV 5.0.

Entities makes it easier for users to understand which sets of Aspects and Aspect Objects are to be treated as a unit and allow features like:

- Reservation within a system.
- Export/Import between Systems.
- Deploy between environments.

The most common entities are Projects, Applications, Libraries, Controllers, Diagrams, Single Control Module (SCM). All entities maintain a list of references to dependent entities, aspects and aspect objects. Operations such as cut, copy, move and paste apply to the whole entity as Export/Import does. Import replaces the currently loaded version of an entity.

Entity Extensions can extend object types of an entity by aspects, which are being worked on by a different set of engineers. A practical example is the control logic in a given Control Module Type and the graphical aspects as Faceplates and Graphic Elements.

Entity Types

Entity types are defined by the system, as defined in [Table 4](#) below:

Table 4. Entity Types and Entity Extension Types

Entity Types	Definition
Control Library Version	<p>A Control Library can contain multiple versions in the library structure. Each of these versions includes a set of datatypes and hosts function block and control module types in the object type structure. An aspect key object in the Administration Structure describes the existence of the library version.</p> <p>The extension Library is an entity extension type. Aspects on the types in the Object Type Structure can be separated into an extension, so that different locations and users can acquire solely control over it.</p>
Control Project	Control Project is the host for a set of Control Applications and Controllers.

Table 4. Entity Types and Entity Extension Types (Continued)

Entity Types	Definition
Control Application	Control Application is a substructure in the Control Structure and a list in the Object Type Structure hosting the application specific types.
Controller	Controller is the sub structure in the Control Structure describing configuration of an AC 800M controller, its communication interface and I/O devices.
Control Module Type	Control Module Type is a substructure in Object type structure starting with an object of type Control Module Type.
Function Block Type	Function Block Type is a substructure in Object type structure starting with an object of type Function Block Type.
Diagram Type	Diagram Type is a substructure in Object type structure starting with an object of type Diagram Type.
Function Diagram	Function Diagram is the graphical diagram presentation of a substructure in the Functional Structure that represents a diagram below an Application in the Control Structure.
PROFIBUS/HART Device Topology	Device Topology is a substructure in Control Structure starting with the Device Topology object.
FF HSE Subnet	HSE Subnet is a substructure in Control Structure starting with an object of type HSE Subnet.

Systems

An engineering setup for distributed engineering can comprise multiple independent systems. The systems can be in different domains and workgroups and spread locally. The key issue during planning is to decide how to split the work packages depending on the engineering team structure and worker location.

The different systems are typically described by the purpose of the system. One system can perform any number of combinations of purpose. Typical systems are:

- Engineering System - A system, inhouse or locally close to the production system where engineers perform the engineering changes and/or where engineering changes from multiple systems get integrated and tested. An Engineering system can be based on one or several notebooks/workstations. If a direct connection is not needed nor available, the system can be configured as a standalone system. In this case the work has to be transferred by the use of Export/Import.
- Production System - A system which is dedicated to run the plant. Once critical automation is being required to operate even parts of a plant it is suggested to do bulk configuration changes in a separate Engineering System to guarantee excellent operating conditions.
- Simulation System - A system typically connected to a process simulation and being used for operator and engineering training.

Environments

It is possible to configure the 800xA system based on the engineering project phase. The two engineering environment configuration options available are Configure/Deploy and Load Evaluate Go (LEG).

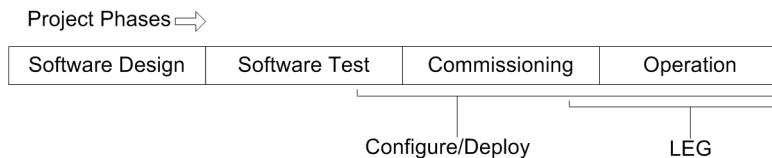


Figure 7. Configuration Methods

Valuable options are:

1. Configure/Deploy: This option is used in the test and commissioning phase of an automation project. It allows changes in the Engineering Environment while the current version of the configuration is being tested in the Production Environment. Once the Engineering version of the configuration is ready, a change impact analysis will be supported by an aspect based difference report.
2. In the detailed engineering phase / software test, of an automation project the engineering environment can be used as an integration environment for the

changes coming in from other engineering systems to integrate for example graphics and control logic configuration (Figure 8).

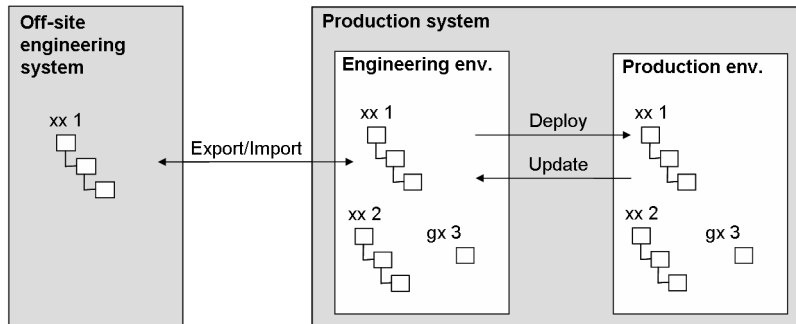


Figure 8. Conceptual view of a System Setup

By using the Export/Import tool configuration data can be moved between separate systems. With Deploy and Update, changes and updates can be moved between the Engineering and the Production Environment.

3. Load Evaluate Go (LEG): This option is used during the late commissioning activities and during operation or maintenance. It allows small changes to be introduced into running process control with reduced risk of process bumps and erroneous changes.

Configure Deploy

Configure Deploy supports Engineering changes to be done in an isolated Engineering environment then deployed in a controlled way to production. Deploy moves the result of changes done in the Engineering environment to the Production or Evaluate environment.

Some advantages by using Configure Deploy are:

- Deploy is a specific action. Several changes can be made which are dependent on each other, for example, graphics consistent with a change in control logic.
- Two or more persons can work with less dependencies between them especially during production and commissioning phase. Changes can be made in some parts of the application while other parts are being verified towards the plant/process.

- Loop checking can be made online while engineering is still performed in the engineering environment.
- A difference report is received at deploy which can verify that the intended changes, and no other changes are made.

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

Load Evaluate Go

Load Evaluate Go allows the user to modify, download and evaluate a version of a running application to a controller without interfering with the running application version. This in turn reduces the risk during application changes in continuous processes by:

- Reducing the risk of applying erroneous application changes to the process.
- Reducing the risk of creating process bumps at download.

Load Evaluate Go is supported for all single and redundant CPUs. The controller CPU load and memory consumption will increase with the same amount as the load/memory used by the running application version.

Load Evaluate Go can be used for PA Applications only.

Load Evaluate Go is optional and requires a separate license.

For more information on Load Evaluate Go, refer to the *System 800xA Engineering Engineering and Production Environments (3BSE045030*)*.

Engineering Repository

The engineering repository is an application server to which engineering solutions from an engineering environment and associated production environment can be saved, compared, and retrieved. The engineering solutions that can be saved are control projects, control applications, controllers, and libraries. These capabilities are available from the Plant Explorer (through context menus) and Engineering Repository Browser windows.

The engineering repository server provides these services to 800xA System nodes using the Windows IIS service. Engineering solutions are handled as AFW files and the meta data is also stored in order to better store and retrieve AFW files. Refer to the *System 800xA Maintenance (3BSE046784*)* for more detailed information.

Application Change Management

Application Change Management (ACM) is a version control tool used for engineering solutions in 800xA System. Multiple versions of 800xA application configuration can be archived in the ACM host and provide an integrated configuration management system utilizing proven in use .afw files technology.

ACM is designed to track history and revisions of previously saved files. This helps users to revert back to any version of a particular file, and identify the difference between any of the version of an entity with the entity in a 800xA system. It prevents two users from working on the same file and overwriting changes.

In order to save an object to the ACM server, it needs to be an entity or a part of any entity. For the object and aspect that are not part of an entity, a separate entity called Custom Entities are defined.

Following are the entities that can be added in ACM server:

- Control Project
- Control Application
- Controller
- Library

Multiple 800xA systems can be connected to the same ACM host in order to share configuration artifacts effectively. Redundant servers in different security zones can be build up utilizing Microsoft Sharepoint Server technology, where connection can be a limited to a dedicated server to server link.

The ACM client is used to configure the ACM server and perform operations such as CheckIn to ACM Server, CheckOut from Server, Get Latest and show version history. ACM provides configuration management capabilities for graphic displays, control libraries, control applications and controllers.

Application Change Management features are as follows:

- Check-in (archive version), check-out (mark a version in ACM host as checked out with or without retrieving it to the 800xA system) and compare on various levels.
- Baseline for system in ACM host and roll back to required older baseline.
- Supports multiple versions of the configuration artifacts.
- Supports Activity log.

- Supports different access control privilege levels.
- Support for versioning at the object level.
- Handles reservations in the ACM Server.
- Provides separate error logs for troubleshooting.

For more information on ACM, refer to *System 800xA Engineering Application Change Management (2PAA108438*)*.

Entity Operations

Deploy

Deploy is used to copy a version of an entity from one environment to another by replacing the current version in the target environment. Typically deploy is performed from the engineering environment to the production environment (Figure 9).

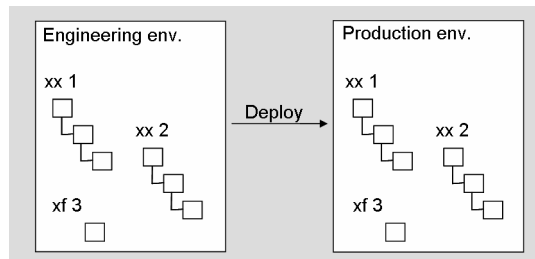


Figure 9. Deploy Changes from Engineering Environment to Production Environment

In a Load Evaluate-Go session an entity version will be deployed from the engineering environment in the evaluation environment. From there the LEG session picks up.

Update

Update is used to selectively copy changes done in the production environment over to other environments. An option to update is offered if applicable before deploying into production environment, but can be done at any point in time, to keep the engineering environment up to date.

Export/Import

Export/Import is used to move solutions from one system to another to allow for Distributed Engineering.

The scope of the Export/Import can be any combination of entities, aspects and aspect objects, for example libraries and applications. Entities are normally exported and imported as units, but can be exported as parts for special purposes such as debug or rescue.

At Export, before the import takes place the Import tool indicates certain types of dependencies:

- Dependencies between instances and their types.
- Dependencies to entities.
- Dependencies to system extensions.

Reservation

Reservation of an item prevents overwriting by other users. Reservation can be made to entities like libraries and application, entity-extensions like for example extension library versions and single aspects like graphic displays.

Privileges control that specific users can take over a reservation to prevent items from being locked up indefinitely. A reservation is only local to a system and an environment.

Multiple Control Builders on Terminal Server

It is possible to run Control Builder Professional as a terminal session on a terminal server. Start the terminal session using the Remote Desktop Connection command. Some restrictions and recommendations are:

A maximum of 5 concurrent Control Builders sessions are supported on a terminal server.

There can be only one Control Builder session per interactive Windows user.

It is not recommended that a Soft Controller be run on the terminal server.

Section 3 Control Design

Overview

The engineer needs to consider the following when designing a control design:

- Engineering Tools.
- Engineering Hardware.
- Application Development.
- Safety.
- Reuse.

Engineering Tools

The following engineering tools are included in the System 800xA. The paragraphs below describe their main purpose and functionalities:

- Control Builder M.
- Function Designer.
- Fieldbus Builder (for PROFIBUS/HART and FOUNDATION Fieldbus).

Control Builder M Professional

The Control Builder M Professional is the main tool for AC 800M controller programming. Control Builder M handles all the IEC 61131 languages and can be used for all types of applications. In Control Builder, the creation of logic can also be done graphically using two extended languages, Control Modules and Diagrams.

Control Builder M is required for compiling and downloading applications to the controller as well as handling controller firmware upgrades, system diagnostics, controller log files etc. Control Builder M also provides the search and navigation (cross referencing) tool in both online and offline mode.

When designing type solutions that should be reused within several projects the recommendation is to use Control Builder M in order to make use of performance optimized statements and languages like structured text.

Control Builder M handles PROFIBUS fieldbus instruments on a controller and fieldbus communication level. When inserting a library containing hardware types, the fieldbus device is available in the Control Builder M hardware structure. Alternatively, Control Builder Device Import Wizard can be used to create a hardware type from a device specific GSD file. For more extended PROFIBUS functionality like accessing device diagnostics through FDT/DTM technology the Fieldbus Builder for PROFIBUS/HART shall be used.

Control Builder M handles Test Mode for test of the controller application. For more extended testing in the System 800xA environment the project can be downloaded to a soft controller hardware.

Control Builder Design Considerations

Before development is started it is recommended to form a strategy for each of the areas below:

Project Constants. Project constants can be used to assign constant values in the 1131 application code. They can also be used for consistent assignment of alarm classes, alarm severities, alarm messages etc. This assignment enables consistency throughout the entire control project independent of different application programmers. The strategy for use of project constants and which constants to use shall be defined at an early stage of the project.

NLS for Controller Alarm and Event. If the control application requires NLS capabilities of the alarm and event conditions and alarm messages an Alarm and Event Translator aspect must be defined and applicable NLS resources must be defined and also used consistent throughout the entire control project. The strategy for use of alarm and event NLS capabilities must be defined at an early stage of the project.

Aspect Object Property. When designing library solutions one have to consider the use of the aspect object property. Only relevant control modules/function blocks shall be defined to be instantiated as aspect objects. To these objects additional aspect are added. A correct strategy for the instantiate as aspect objects will reduce the number of aspect objects in the aspect directory.

Alarm Classes and Severities. The parameters Class and Severity on alarm control modules/function blocks are essential for alarm and event list filtration and sorting. In an early stage of the project the strategy for use of Class and Severity settings must be defined in order to achieve a proper alarm and event list handling on the System 800xA.

Name Syntaxes. During control application development the general AC 800M Connect name syntaxes shall be followed in order to achieve a good and common structure with the standard objects from AC 800M Connect libraries. Refer to standard Control and I/O documentation for name syntaxes.

Name Uploader Aspect. The Name and Description properties of a Control Module or a Function Block aspect will be uploaded if the Name Uploader aspect is started. The Name and Description properties are reserved for use of the Name Uploader aspect. In an early stage of the project the strategy for use of Name and Description properties must be defined. Refer to standard Control and I/O documentation for Name Uploader details.

OPC Property Settings. Variables and parameters defined in Control Builder M will be displayed as OPC properties in the Control Module, Function Block, Program, Diagram or Application aspect.

Variables not applicable as OPC properties shall be defined with the attribute hidden. Variables and parameters not defined as hidden shall have a write permission defined. The write permission is required to apply the System 800xA security on the OPC property.

In an early stage of the project the strategy for property permission must be defined. Refer to standard Control and I/O documentation AC 800M Connect property permission settings.

Peer to Peer Communication. Peer to peer communication between applications in different AC 800M controllers and between applications in the same controller can be defined using Inter Application Communication (IAC). IAC uses communication variables for cyclic communication between diagrams, programs, and top level single control modules. Communication variables can exist in the same application, the same controller, or in another controller (peer to peer). The IP

address of the communication variables is resolved during compilation in the Control Builder. Peer to peer communication using IAC is not possible with distributed application (an application that is executed in more than one controller). Peer to peer communication can also be defined to other controllers with different protocols like MB300. In an early stage of the project the strategy for peer to peer communication regarding update rates, performance etc. must be defined.

Diagrams . They are designed to configure Basic Process control as well as Safety Instrumented Systems logic in a graphical presentation. In order to reduce the complexity for the user Diagrams, do not support global and external variables. Communication between Diagrams is done using Communication Variables and IO connections that are done on Variables of the Diagram or Block on it.

Tasks. The required task times for the control application must be analyzed and applicable tasks in each AC 800M controller must be defined and connected to the different parts of the control application. The defined required task time must be compared to used and actual task time during commissioning in order to verify that the requested settings actually are fulfilled by the controller and not causing too much controller load. The load of the controller will impact the general controller performance like communications etc. Refer to standard Control and I/O documentation for details of task tuning and controller performance.

Function Designer

The Function Designer supports controller application configuration according to a system-wide Functional Planning approach. It is a tool for working on top of the Control Builder M. The applications configured with Function Designer can be downloaded to applicable controller hardware by the Control Builder M. If combined with an application design with each loop or tag contains one diagram. The Function Diagrams can automatically be created and parametrized based on signal and tag list utilizing the Bulk Data Manger or Process Engineering Tool.

Basic control application design issues to take care for Function Block and Control Module programming in Control Builder M also apply to control logic configured using Function Designer.

Functional Planning allows you to work in comprehensive Function Diagrams that can cover control logic, engineering signals, field devices and even representations of information items of other aspect systems important for the plant functionality.

Function Diagrams allow you to configure, commission, maintain and document the control applications of a plant from a functional perspective. To configure control logic within Function Diagrams you insert Function Components representing Control Modules and / Function Blocks or Functions from AC 800M Connect libraries or industry specific libraries, connect them graphically and generate them into diagrams allocated to a control application in Control Builder M Professional.

Planning Considerations

The main engineering project requirements that can be fulfilled by using Function Designer are:

- Configuration driven by Functional Structure (functional decomposition) of the plant.
- Control application independent planning in early planning stages. Control logic allocation to application can be done in a late stage.
- Support for IO allocation tool: Allocation of engineering signals to hardware units can be done in a late planning stage.
- Simplified graphical Function, Function Block, Control Module and Sequence Programming in Function Diagrams.
- Extended Name Handling:
 - Non-IEC61131 compatible object names can be used.
 - Name synchronization between Object Name, Control Builder Name and Name property of Function Block or Control Module.
 - Rename during copy.
- Electronic Function Diagram documentation (offline and online) to ease commissioning and maintenance.
 - Diagram reference based horizontal navigation following the signal flow.
 - Diagram page reference based sheet navigation in diagrams following the signal flow across pages.

- Diagram based vertical navigation following the diagram hierarchy in the Functional Structure, going from parent diagram to child diagram and back as well as navigating to other Aspects of the objects as Trends, Faceplates, or Event Pages.
- Live data display and value forcing in online mode.
- Context sensitive aspect navigation from within diagrams.
- Bulk printing of Function Diagram documentation including generated contents documentation and generated hidden items documentation.
- Automatic creation of connection variables.
- Function Components are not restricted to control logic elements as provided by AC 800M Connect libraries and industry specific libraries. By default Engineering Signals and Field Devices are supported additionally. The ability to configure own Function Components allow to represent other elements in Function Diagrams if needed.



Currently Function Designer is not released for configuring SIL control applications. The use of Function Designer with Batch Management is not recommended at this time.

Fieldbus Builder PROFIBUS/HART

The Fieldbus Builder PROFIBUS/HART integrates the FDT/DTM technology in the IndustrialIT 800xA system and provides the infrastructure in ABB Aspect platform to access field devices.

In accordance with the functionality of the DTM, the Fieldbus Builder PROFIBUS/HART accesses field devices for:

- Configuration
- Parameterization
- Diagnostics
- Maintenance
- Documentation



The current version of the Fieldbus Builder PROFIBUS/HART supports FDT 1.2 and addendum standard interfaces.

The Fieldbus Builder PROFIBUS/HART supports the following functions in the hardware topology of the Plant Explorer Control Structure:

- Saving and reading DTM instance data set.
- Configuration of the OPC Server PROFIBUS/HART.
- Provision of DTM internal instance data for bulk data applications (if supported by the DTM).
- Support of 800xA System Audit Trail interface.
- Topology planning and managing the fieldbus structure down to PROFIBUS and HART devices.
- Managing (add, move, remove, or copy) PROFIBUS devices connected to AC 800M / CI854 and HART devices connected to local I/O, remote I/O or Multiplexer.
- Assigning the HART devices to I/O channels.
- DTM call up from System 800xA nodes.
- Configuring and removing the communication paths for online communication between the DTM and the physical field device in the system.
- Comparing field device online and offline data.
- Uploading/downloading the current DTM instance data and device commissioning.
- Exporting/importing device instance data records.
- Historical data set storage to document the life cycle of the device instance.

Apart from the above functionalities Fieldbus Builder for PROFIBUS/HART provides OPC connectivity to higher level application such as Asset Optimization.

Fieldbus Builder FF

Design recommendation for using FF libraries in Control Builder M is found in the Device Management section.

Hardware Considerations

1. Performance of AC 800M central units - RAM, task times etc.

The applicable PM8xx central unit must be selected based on project size, required spare available after project.

2. Type of IO - SOE, HART, EX, SIL?

Several different I/O types are available to the system. Specific applications requires specific types of I/O like SIL, EX, connection to HART instruments, support for sequence of events etc. I/O can be added to the central unit in various ways like electrical module bus, optical module bus, PROFIBUS etc. Optical module bus also supports modems for remote I/O up to 1000 meters. Different types of I/O also requires different types of MTU.

3. Drives integration - module bus, drive bus, PROFINET I/O?

For advanced drive control a specific drive bus communication unit, CI858, is used. Drives can also be added on the optical module bus and through PROFINET I/O.

4. External communication - MB300, serial etc

The required external communication must be considered in order to define the required communication hardware to be added to the central unit.

5. Fieldbus - PROFIBUS DP/PA, FF, HART, IEC 61850.

6. Redundancy - System 800xA supports redundancy on all levels and the level of redundancy and required hardware (MTU, LAS, DP/PA converters etc) to fulfill the redundancy must be considered.

7. SIL applications require PM865/SM811 or PM867/SM812, and specific safety I/O.

Implementing a Control Application

When designing a control application the required standard system 800xA libraries must be added to the control project. The System 800xA standard libraries contain a broad range of objects with different functionality.

Refer to the Library Objects Overview appendix in the *System 800xA Control AC 800M Binary and Analog (3BSE035981*)* for a summary of commonly used libraries.

ABB also supply different type of standard libraries targeting different industries or applications.

Use of Industry Specific Library Solutions

In some cases, specifically developed libraries for different industries and customers extends the 800xA standard library functionality. Usually these library solutions comes with their own guidelines on how to apply the library objects in the 800xA system. Please contact ABB for more information about specific libraries and how to make them available.

Use of Application/Project Specific Library Solutions

As an addition to the standard libraries inserted in the control project application/project specific libraries can be added to the control project. Certain type solutions that are used several times within the control project could be designed once as a control module or function in a library and then be reused throughout the different applications in the control project. The library can also be used in other control projects.

Specific applications like batch applications requires that the template control modules in BatchAdvTemplatesLib are copied and further developed in application and or project specific libraries.

Control Builder M application code can be organized in two different ways in the applications, either as control modules or as programs. The following Tank example distinguishes between the two different approaches.

A tank including a valve and some sequence and interlock code can be implemented in two principle ways:

1. The tank is defined as a type solution control module within a library including the valve, sequence and interlock code applicable for a tank. An instance of this tank is then inserted in the Control Modules and all application code related to the tank is defined as the T100 control module.
2. A program named T100_Program is created. The Program includes the valve, sequence and interlock code applicable for the tank. Refer to the following figure.

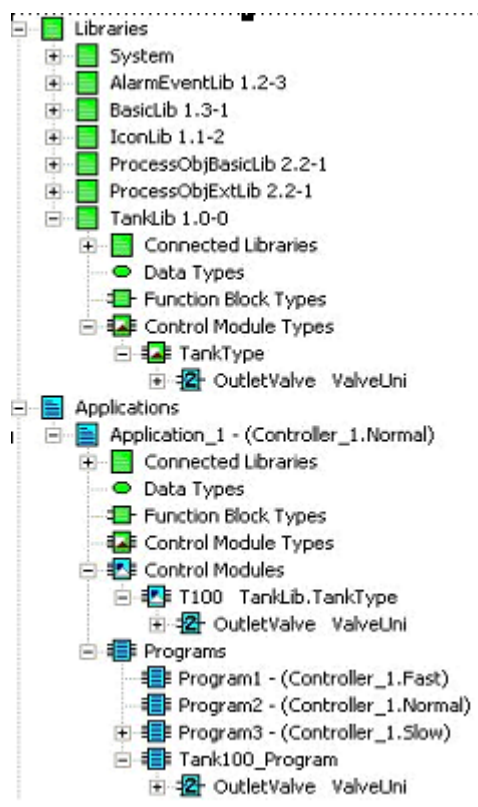


Figure 10. Tank application implemented in two different ways

In both cases the same tank control logic can be accessed by open the editor either for the T100 control module or the T100_Program. The control module and the program are only containers for 1131 application code. The two solutions differ however in some principal important ways:

- The TankType object type is a composite object. This composite object can be extended with additional aspects like graphical elements, faceplates, documentation etc.

- The TankType object type including all of its aspects can be reused within all applications in the control project or reused within other control projects.
- The instance T100 object can be accessed as an aspect object in the System 800xA Operations. The T100 object is handled in a consistent way regarding navigation in an Operator Workplace, display of alarm and event lists and trend displays etc.

Safety/SIL Applications

This section provides an extract of guidelines and safety considerations related to AC 800M HI. In a planning phase the guideline focus is set on; Safety Principles overview, Management of functional safety, Hardware considerations and Application software. It is however strongly recommended that the *System 800xA Safety AC 800M High Integrity Safety Manual (3BNP004865*)* be read completely. The recommendations and requirements found in the safety manual must be considered and implemented during all phases of the life cycle of the AC 800M HI.

Safety Principles

The overall safety requirements of a process are identified by carrying out hazard and risk assessment. If the level of risk in the system is high the need for a Safety Instrumented System (SIS) may arise. For each identified safety function the Safety Integrity Level (SIL) must also be identified.

The AC 800M High Integrity is certified to handle safety functions with SIL1-2 or SIL3 requirements.

Management of Functional Safety

It is recommended to create a management strategy as a base for the safety work. Ensure the strategy is communicated within the organization and that required safety activities are implemented, monitored and revised.

Hardware Consideration

The AC 800M HI, including its I/O system can be designed for any optimization of system availability (system uptime) by individually being configured with redundant input modules, redundant CPU/communication modules and/or redundant output modules.

The AC 800M HI does not require any redundancy for safety integrity reasons.

If the safety functions of the plant require more than one AC 800M HI, for capacity or geographical reasons, they can interchange safety critical data by means of the certified peer-to-peer communication. For more information refer to the section covering Communication Interfaces in the *System 800xA Safety AC 800M High Integrity Safety Manual (3BNP004865*)*.

Application Software

Activities must be defined and planned for design, development and test/verification of the application software until it is integrated with the hardware and ready for installation and commissioning.

The table below gives a brief description of Safety considerations and activities for a selection of areas. More detailed information is found in *System 800xA Safety AC 800M High Integrity Safety Manual (3BNP004865*)*.

Table 5. Safety Considerations

Area	Considerations
Software Architecture	Rules and guidelines on how to utilize the features of AC 800M HI and the Control Builder M Professional to achieve a good design of the application software is described in the <i>System 800xA Control, AC 800M Configuration (3BSE035980*)</i> .
SIL Marking	<p>A SIL marked application must only contain SIL classified types and functions and only contain the language constructs allowed for the selected safety integrity level.</p> <p>This has to be considered also for upgrades and rollups.</p> <p>The AC 800M HI allows SIL classified functions and non SIL classified functions to be programmed and executed in the same controller if they are programmed in different applications.</p>
Programming Languages	<p>The international standard for the process industry sector, IEC 61511 defines the supported programming languages for up to SIL3 classified applications.</p> <p>Application programmers working with higher SIL, are referenced to IEC 61508.</p>

Table 5. Safety Considerations (Continued)

Area	Considerations
Libraries	An overview of certified library elements is presented in Appendix A, Certified Library, in the <i>System 800xA Safety, AC 800M High Integrity Safety Manual (3BNP004865*)</i> .
Control Builder M Professional - Settings and Restrictions	The IEC 61131-3 application code must be compiled before it is downloaded to the AC 800M HI. The compilation and download is performed with Control Builder M Professional.
Controller Settings and Restrictions	The internal diagnostic and fault reaction of AC 800M HI must be adapted to the actual process according to available settings on controller level and task level.
I/O Module Settings	Each High Integrity I/O must be configured according to specific directions and settings for the specific device.
Access Management Settings	The “maximum number of forces” property must be set based on the characteristics of each application and the operation philosophy of the plant.
Configurations Management	Procedures for Configuration Management must be established and used during the whole life cycle of the application software.
Software Module Testing	Testing of project specific Function Blocks and Control Module types must be performed to verify their specified functionality. Execution time and memory consumption must be verified in an AC 800M HI.
Software Integration Testing	Software Integration testing can be done by either using the AC 800M HI or the <i>Test Mode</i> or <i>Simulation</i> functions.
Integration of Application Software with the System Hardware	All software and hardware modules of an AC 800M HI must be verified that they function together and interface with each other as a complete system. Verify I/O configurations and other external interfaces and correct function of redundant components. These tests are performed using the AC 800M HI hardware.

Table 5. Safety Considerations (Continued)

Area	Considerations
System Integration Testing	Projects containing functions that are dependent of more than one controller, or other external systems must be verified in a complete system setup, comprising all involved sub systems. These tests can be done in a staging area or in the actual plant after installation.
Modification Testing	An impact analyze must be performed on modified hardware or software parts that already are tested. Also Functional and Regression testing have to be performed.

Reuse

Building Composite Objects

The 800xA System supports high level composite objects. When designing composite objects several system wide considerations must be done. The list below describes issues related to one aspect object in one library. The 800xA system supports all this functionality included in one object type with inheritance to all instances of this type. This is efficient engineering which differentiates 800xA from competitors.

The example below shows the workflow for the design of a batch equipment starting from the BatchAdvTemplatesLib:UnitTemplate.

Table 6. Batch Equipment Design

Step	Action	Affects
1	Creation of a new application specific library	Library Strategy
2	Definition of phase parameters including access level	Security Strategy
3	Definition of Alarm classes/severities	Alarm List Strategy
4	Definition of Alarm & Event settings	AC 800M hardware performance

Table 6. Batch Equipment Design (Continued)

Step	Action	Affects
5	Equipment control logic with drives, device etc.	Drives, device, Fieldbus Builder etc.
6	Define Display Elements	Overall graphics and navigation strategy
7	Define Faceplates, alarm indications (Alarm Control)	Overall graphics/Alarm Strategy
8	Data ToSimpleEvent for data logging	PDL/IM batch logging
9	Configure logs for Trend and Long time storage (IM)	PFC BMA Trend Collections
10	Define Trend Displays	Overall navigation strategy
11	Perform Additional applications calculations	Calculations/Softpoints
12	Provide documentation/SOPS etc.	File Viewer

Section 4 Application Design

Overview

This section describes design considerations for application parts that are not directly controller based:

- Reusable parts (libraries, object types, etc.).
- Graphics.
- Basic trending and history.
- Documentation and basic information access.
- Equipment diagnostic and data exchange.

Planning application design for alarm management, data management, external data integration, asset condition monitoring, and system configuration documentation are handled in separate, more detailed sections of this document.

Reuse

System 800xA supports concepts that allows you to organize reuse of engineering work within projects and across projects, the most important are:

- Versioned Object Type Libraries and Object Types/Object Type Groups.
- Aspect Categories and Aspect Templates.

Engineering and configuration tools use this base functionality to provide preconfigured reusable entities to start-up with, for example:

- AC 800M Connect provides versioned object type libraries for Control Builder M Professional. Extended also for use of Function Designer. In general Control Builder M requires control and hardware unit libraries to be versioned object type libraries.

- None-control object types can be put into versioned object type libraries, but if a managed version administration is no issue they can also be put in object type groups only. An example are the CBM_Signal object types brought into the system by Engineering Studio / Engineering Base.
- All aspect systems typically provide in aspect categories in the system, that can be used to create aspects in objects. Example: Graphics Display PG2.
- Some aspect systems provide additionally aspect templates. They can be used to pre configure the behavior of corresponding aspects in objects. Example: History Log Templates.

General advice:

- Reuse the existing object types / aspects (delivered with 800xA or available from former projects) in your application whenever possible.
- Engineer and configure application parts for reuse whenever it is beneficial in a project and across several projects.

Typical benefits of reuse are stable, well tested, and consistent application (solution) parts, lower effort to engineer and configure the total application (solution) and shorter time to delivery.

[Table 7](#) lists which principal reuse opportunities exist and how you can use it in a project to build corresponding application parts. For more details refer to [Reuse Instructions](#) on page 325, *System 800xA Configuration (3BDS011222*)* and additional tool or aspect system specific configuration instructions.



Additionally to the concepts listed in [Table 7](#) reuse features on file level or on tool level are available in some tools. Examples are project documentation templates of Control Builder M, files in Engineering Templates folder for Bulk Data Manager, (optional) Report Templates as files for Application Scheduler, and the Graphic Libraries in Graphics Builder.

Planning Considerations

1. Use versioned object type libraries when the requirement to deliver new major / minor versions and revisions is given and the need to keep track of deliveries. Especially when an internal or external customer is affected.
2. Object types in object type groups you can use if:

- a. if object type contents is stable and delivered once (version management is no issue).
- b. or scope of reuse is limited, for example to a small local project group working on one project.
- c. or controlled long term reuse is not an issue.

Table 7. Reuse Opportunities

No	Required in Project /Existing or additional...	Used For	Concept	Operation
1	Control (Extension) Library Existing or Add. (Extension) Library without control aspects	Standard Control Libraries, BU/BA Control Libraries	Versioned Library / Extension Library (of Object Types / Composite Object Types)	New Object (instantiate)
2a	Object Types / Composite Object Types	In versioned (Extension) Libraries: Process Object Types, etc.	Object Type / Composite Object Type	New Object (instantiate)
2b		In Object Type Groups: Control Application specific Control Object Types, Hardware Unit Object Types, Field Device Object Types, Asset Object Types, etc.		

Table 7. Reuse Opportunities (Continued)

No	Required in Project /Existing or additional...	Used For	Concept	Operation
3	Object Type to provide add. / modified Function Diagram templates	Function Diagram or Topology Diagram templates	Object Type providing a template	New (diagram)
4	Object Type to provide add. / modified Function Diagram skeletons	Function Diagram or Topology Diagram skeletons	Object Type providing a skeleton	New (diagram)
5	Object Types for generating variant instances	Object Types supporting variants, designed in Reuse Assistant Architect	Object Type providing a Reuse Instruction	New Object (instantiate). Additional substitution and selection input entered in Reuse Assistant Builder
6	Aspect Category. Either programmatic (Type / Category) or by configuration (Category)	Aspect System providing Aspect Type / Aspect Category	Aspect Category	New Aspect (instantiate)
7	Aspect category to provide add. / modified templates	Aspect System Structure: File Viewer, Parameter Manager, Graphics Builder	Existing Aspect Category as template	New Aspect (instantiate)

Table 7. Reuse Opportunities (Continued)

No	Required in Project /Existing or additional...	Used For	Concept	Operation
8a	Objects with aspects to provide add. / modified templates	Library Structure objects: Document Manager Templates, Alarm and Event Configuration templates, Trend Templates, History Log Templates	Aspect as template	Reference (by name)
8b		Scheduling Structure: Report Templates		
9	Aspect to provide skeleton (source to copy).	Basic reuse in projects without involving own aspect categories	Aspect as skeleton	Copy+Paste
10	Object instance to provide skeleton (source to copy).	Basic reuse in projects without involving own libraries / object types.	Object Instance/ Composite Object Instance	Copy+Paste

3. Do not invest in own object types (neither in libraries nor in object type groups) if existing object types are sufficient. Developing object types requires extended system skills.
4. If additions are necessary try to inherit given object types in own object types and add aspects. Remark: This is not possible with standard control object types. Else add project specific aspects to the given object type.
5. Carefully evaluate if you have to develop own additional control object types. If yes, develop them in a versioned object library loadable via system extension (Table 7, No.1 and No. 2a).
 - a. If the object types shall be used for Functional Planning include Function aspects, best into an extension library loadable via system extension.

Default symbol layout for new control object types are provided automatically by the added Function aspects.

- b. If the object types shall contain other additional aspects (Graphics, Documents, etc.) include them, best into an extension library loadable via system extension.
 - c. Only if the object types are really application specific use application-defined types (Table 7, No. 2b).
6. Carefully evaluate if you have to develop own additional (non-control) object types, for example for plant structuring or documentation purposes. If yes, develop them
 - a. as a versioned object library loadable via system extension (Table 7, No.1 and No. 2a)
 - b. or as a versioned object library importable as an archive (Table 7, No.1 and No. 2a)
 - c. or, if statement of Step a applies in your case here, as a object type group (Table 7, No. 2b).
7. If object types for new Hardware Units, Field Devices, Assets are required involve responsible ABB product management.
8. If customer / project specific templates / skeletons for Function Diagrams and / or Topology Diagrams are required: Copy, rename and modify the existing object types (Table 7 No. 3 and No. 4.).
9. If non-default symbols and / or additional component object types for Function Designer or Topology Designer are required involve responsible ABB product management.
10. For object types with standardized variants consider to use Reuse Assistant to develop and build instances of these object types (Table 7, No. 5). Regarding inclusion of these object types into libraries the same considerations as for other object types apply. Be aware that only substitution variable values for the instances can be set with Bulk Data Manager, answer selection must be done interactively.
11. Uniform templates and skeletons for documentation and configuration work should be defined and stable before bulk work in project begins.

12. Some aspect systems support to configure template aspect categories in the Aspect System Structure. Use this to provide basic reusable aspect contents for the project, for example preconfigured Graphic Elements or Display aspects or Parameter Manager aspects (Table 7, No. 7).
13. Some other aspect systems provide support to configure templates in objects of Library Structure or Scheduling Structure and reference the templates from the instances. Use this to provide basic reusable aspect contents for the project, for example preconfigured Alarm and Event Configuration templates or Document Manager templates (Table 7, No. 8a) or Report Templates (Table 7, No. 8b).
14. Special requirements not covered by standard 800xA aspect types / aspect categories and their template configuration possibilities you can consider to fulfill by developing an own aspect system. This you only should consider if there is no other way (Table 7, No. 6). Involve responsible ABB product management.
15. Reuse by copy (+paste) is supported for aspects and object instances, both in Plant Explorer and in Bulk Data Manager. Reuse by copy should be used in projects only if reuse of object types, aspect categories, and templates is not appropriate for some reason (Table 7, No. 9 and 10).

Documentation and Information Access

System 800xA provides various aspect systems that allow to store and manage documents and other kind of information in the system and to access it using Plant Explorer.

Table 8 lists principal opportunities for documentation and information access and how you can use it in a project to build corresponding application parts. For more details refer to Documents on page 293, Information Access on page 295, *System 800xA Configuration (3BDS011222*)* and additional tool or aspect system specific configuration instructions.

Table 8. Documentation and Other Information Access

No	Required in Project	Aspect System	Supports
1	<p>Dynamic engineering and maintenance documents:</p> <ul style="list-style-type: none"> . Automatically adapting to Structure and Object context . and / or requiring data base for meta data (document properties) management . and / or showing live process values <p>Examples: Function or Topology Diagram document meta data base (by default), electronic interactive SOP (standard operating procedure), tagsheets, ad hoc maintenance reports etc.</p>	Document Manager	<p>Dynamic Word, Excel and AutoCad documents (property references from within documents).</p> <p>Static documents for other formats.</p> <p>Document meta data in relational (MSDE) database, Bulk Data manger Excel user interface to meta data.</p> <p>Checkin / checkout, two document versions</p> <p>Templates / Skeletons.</p>
2	<p>Static documents for operators or maintenance persons without the need for document management. (By default used for report handling via Application Scheduler.)</p>	File Viewer	<p>Static documents only.</p> <p>Public document (.pdf, for operator) and work document.</p> <p>Checkin / checkout, versions for work document.</p> <p>Bookmarks into.pdf.</p> <p>Templates / Skeletons.</p>
3	<p>Integrate web pages or local html documents without the need for document management</p>	Web Page	Web page or html documents
4	<p>Basic third party application integration</p>	ActiveX Wrapper	ActiveX application integration
5	<p>Basic third party application integration</p>	Windows Application	Windows application integration

Table 8. Documentation and Other Information Access (Continued)

No	Required in Project	Aspect System	Supports
6	Support staff in operation / maintenance of complex application objects	Help Aspect	Specific online help for object types and instances
7	Engineering and configuration information and complementary object or aspect information for operation and maintenance. Example: Integrated article information database.	Parameter Manager	Parameter aspects, flexible parameter aspect categories definition including data entry support (data type checks, validation checks, pick-lists, default values, expressions). SignalParameter aspects of CBM_Signals engineering signals. Data sharing via property references. and/or (AES) Property Transfer aspects Default Bulk Data manger / Excel bulk data user interface. Data base for document management.
8	Engineering and configuration information and complementary object or aspect information for operation or maintenance.	General Properties	Information integration based on aspect directory: Properties aspects, template categories, data sharing via property references and/or (AES) PropertyTransfer aspects

Planning Considerations

1. Check requirements on documentation to be used at site during engineering, commissioning or operation/maintenance: Are dynamic document parts required or document meta data to be managed (Table 8, No. 1 and 2)?
 - a. If yes: Plan for Document Manager Document aspects to hold these documents
 - b. If no: Plan for File Viewer aspects to hold the documents.

2. If you use Function Designer decide on how to manage Function Diagram document meta data: Default solution via Function Diagram Document aspect or a substitution of this aspect by a Parameter Manager aspect or a General Properties aspect. The document aspect allows to provide the Function Diagram as a pdf- file additionally to the Function Designer view. In the other cases the Parameter aspect or General Property aspect has to be designed an the Function Diagram template object types have to be adapted ([Table 8](#), No. 1).
3. Corresponding decisions have to be taken for Topology Designer.
4. Refer to further documentation considerations in chapter System Configuration Documentation.
5. Plan for Web Page, Active X Wrapper and Windows Application aspects if there is a corresponding special requirements, see ([Table 8](#), No. 3, 4 and 5). Remark: If you have decided to primarily use Document Manager you can show web pages using Document aspects. No need then to use Web Page aspects.
6. If online help support for Application objects / object types is required, plan to use corresponding Help aspects and plan for the needed additional documentation activities and tools ([Table 8](#), No. 6).
7. Check requirements on additional / complementary information storage for in engineering, commissioning or operation/maintenance: High amount or low amount? Comprehensive or minor data entry support ([Table 8](#), No. 7 and 8)?
 - a. High amount, comprehensive data entry support: Plan for Parameter Manager aspects to hold the information
 - b. Low amount, minor data entry support: Plan for General Properties aspects to hold the information.
8. If requirements to include simulated / calculated process values / laboratory values exist additionally: Plan to fulfill these requirements with Softpoint and Calculation services ([Table 8](#), No. 9).

Section 5 Alarm Management

Overview

800xA handles alarms and events generated internally in the system, a controller or other hardware unit or in applications. Alarms and events are normally presented in the Operator Workplace and can be directly connected to steer a process. Alarms and events can also be transferred to user devices such as mobile telephones and e-mail accounts using SMS and e-mail Messaging software.

This section also describes the planning of:

- [Alarm Strategy](#).
- [Alarm Generation](#).
- [Alarm Retrieval](#).
- [Remote Users](#).

Alarm Strategy

A good alarm system should be designed to meet the users' needs and operate within their capabilities. The information the alarm system presents should:

- be relevant to the user's role at the time
- indicate clearly what response is required
- be easy to understand
- be presented at a rate that the user can deal with

To meet those requirements it is recommended to create an alarm strategy before setting up the alarm environment. The strategy typically describes alarm generation, retrieval and blocking principles as well as alarm priorities and definitions.

This section contains guidelines. For further reading we recommend *Alarm Systems - A guide to design, management and procurement, publication no. 191, ISBN 085931 76 0*, published by *The Engineering Equipment and Materials Users Association in 1999* and available at www.eemua.co.uk.

Alarms and Events

Events informs about changes. Alarms alert of an abnormal state and must be acknowledged.

There are three types of alarms in the system:

- Process alarms are generated from the process, such as failure in a valve or pump or an overpressure in a tank.
- System Alarms and System Events are generated from the 800xA System, for example network problems, file system errors or server errors.
- Maintenance Alarms/AO Alarms alerts the operator about equipment that needs maintenance or that parts of the plant is running in a non optimal way. These alarms are typically not time critical and should be displayed in a separate alarm list.

Alarm Generation

Process alarms can be generated in several ways:

- By using standard process objects in the AC 800M controller. These objects are located in the standard libraries and predefined in the system.
- By using specific alarm condition blocks in the AC 800M controller. Alarm condition blocks are used when adding complementary, non-default process alarms.
- By using Soft alarms expressions set in the Process Portal. Soft Alarms are controlled by the Scheduler and allow alarms to be triggered due to a defined expression, for example a calculation value or a specific event.
- By using Soft Points. Soft Points are object types and instances which are not tied to a controller but can be driven by applications such as Calculations and external applications.

More information about Alarm Generation is found in the:

- *System 800xA Configuration (3BDS011222*)*.
- *System 800xA Control, AC 800M Configuration (3BSE035980*)*.
- *System 800xA Control AC 800M Binary and Analog (3BSE035981*)*.

Alarm Priorities

It is recommended to prioritize alarms according to their severity. The default priorities in the system are presented in the table below. Each alarm is configured with an OPC-severity code between 1000 and 1. Value 1000 is normally mapped as most critical but this is configurable as long as the mapping is consistent throughout the system.

Table 9. Default Priority Levels for Process Portal Alarms and Events

Priority Level	Definition	OPC-Severity
1	Critical (Fatal*)	1000-751
2	High	750-501
3	Medium	500-251
4	Low	250-1
* AC 800M has a fifth Priority Level defined as Fatal. Fatal has a default OPC-Severity code of 1000 which makes it a level 1 priority		

Table 10. Examples of Triggers

Priority	Process Alarms	Process Simple Events	System Alarms	System Simple Events
1	Level alarm HHH	Level event HHH	Loss of Aspect Directory, Corrupt memory, 1131 task aborted	Only displayed together with a system alarm to extend the information about the alarm.
2	Level alarm HH	Level event HH	Loss of communication, Faulty I/O, Memory close to being full	Only displayed together with a system alarm to extend the information about the alarm.
3	Level alarm H Object Error for a motor	Level event H	Loss of redundancy, Summary alarm for Error Handler	Alarm definition failed
4	-	-	-	I/O Underflow/ Overflow

The table contains possible reasons for a specific alarm/event to be triggered. Triggers for Process Alarms and Process Simple Events are user defined. Shown here are Default Values for some library objects.

The most important System Alarms are described in the *System 800xA Control, AC 800M Configuration (3BSE035980*)*, Appendix B.

Alarm Retrieval

The retrieval for alarms and events can be configured in different ways. Some configurations are made online by the operator, others are configured in the controller by a user with an engineer role.

The functions for retrieval are:

- Filtering and Presentation functions inherent in the alarm list can be used to customize the presentation, for example setting the sort order or the size of columns. The functions are available to the operator and can be set online. More information about Operator workplace configuration is found in the *System 800xA Operations Operator Workplace Configuration (3BSE030322*)*.
- Alarm list layouts can be configured also on system level by creating new layouts in the library structure in Plant Explorer. For example it is possible to set up rules for filtering, to decide which columns to be included as well as setting presentation details such as alarm colors, date and time format. Configuration on controller level requires a user with the engineering role. More information is found in the *System 800xA Configuration (3BDS011222*)*.
- Area alarm structures can be set to restrict the alarm retrieval according to different process areas in the plant. Alarm lists (standard or user defined) can retrieve alarms from all process areas in the system, from the object, from the object and descendents in this structure or from objects and descendents in a specified structure. A common way to map alarms to process areas is to configure the parameter “Class” in a Control Module or a Function Block. Area alarm structure configurations require the engineering role.
- Alarm bands provide a summary display for selected alarm lists. The Alarm band provide a link to the alarm lists.

Alarm Blocking

Alarms can be blocked in the operator station or on Controller level. Some blocking functions can be configured by the operator, others require the role of the engineer. The tables below summarize the available functions.

Table 11. Alarm Blocking Functions in the Operator Station

Function	Description	Usage/Example
Faceplate Disabling	Alarms and events are disabled, from the faceplates, via interaction parameters.	Disabling from the Faceplates disables both the alarm and the control action. (Disabling from the alarm list only disables the alarm)
Hiding	Hiding of alarms based upon process state. The alarm is hidden in the alarm list.	Used on process areas or objects to hide alarms which are not of interest at a specific stage or area. The Hiding function can be manually or automatically activated.
Shelving	Alarms are removed from the main list for a specified time and are placed in the shelved list.	Shelving can be used to temporarily shelve nuisance or standing alarms for a specified time period. After the specified time has elapsed, the alarm reappears in the main list. Standard shelving and One shot shelving are two ways of shelving the alarms.

Table 12. Alarm Blocking Functions on Controller level

Function	Description	Usage/Example
Suppression	Suppression via control application	The Suppression function is user defined and built into the application. Suppression is used to block alarms which are not of interest at a specific stage or area.
Inhibit	When the control action is inhibited the system does not act on alarms or events. Although the alarms/events are still logged and presented to the operator.	Inhibit can be used to block the control action while still supervised. (For example, Level supervision of a tank - if inhibited the pump is still supervised but will continue to pump even if the tank level is achieved.)
Disabling	No alarms or events are generated in the system and no control action is taken. (No alarm/event generation, logging or control action is taken)	Disabling can be used to block deviated alarms, for example due to a faulty transmitter. (For example, Level supervision of a tank - if disabled the pump will continue to pump and no alarms will be sent even if the tank level is achieved.)

More information about alarm blocking functions is found in the:

- *System 800xA Operations Operator Workplace Configuration (3BSE030322*)*.
- *System 800xA Configuration (3BDS011222*)*.
- *System 800xA Control, AC 800M Configuration (3BSE035980*)*.

Remote Users

SMS and e-mail messaging provide a method for sending messages based on alarm and event information to user devices such as mobile telephones and e-mail accounts. It is possible to control sending messages by configuring a message schedule for each user. The message schedule allows one active paging time interval for each day of the week. The SMS and e-mail messaging function is a system extension to the 800xA System. For more information about the SMS and e-mail Messaging function, refer to the *System 800xA Operations Operator Workplace Configuration (3BSE030322*)*.

Section 6 Information Management

Overview

Information Management functions are inherent to ABB's Industrial IT System 800xA. Historical, process and business data is collected from available sources and stored securely.

History servers are logically linked together at a site, and across an enterprise, to provide a common historical data environment. This allows seamless data retrieval from all points within the system.

The following topics should be considered when planning to use Information Management functions:

- [Data Collection.](#)
- [Archive Strategy.](#)
- [Reporting Services.](#)
- [Scheduling.](#)
- [Information Presentation/Desktop Services.](#)
- [Application Services/Application Data Management.](#)

System Dataflow

There are two phases to consider for data planning:

- Logical phase.
- Physical phase.

The logical phase is determining what to collect. The physical phase is determining where to collect the data and how to plan the hardware for collection.

It is important to analyze the data collection requirement prior to determine the physical requirements. The logical requirements will determine the physical requirements.

The basics is the number of signals to be collected and the rates for collections, but also online storage for messages, production data and reports as well as spare capacity for other applications. Figure 11 below shows a logical view of a possible setup for Information Management.

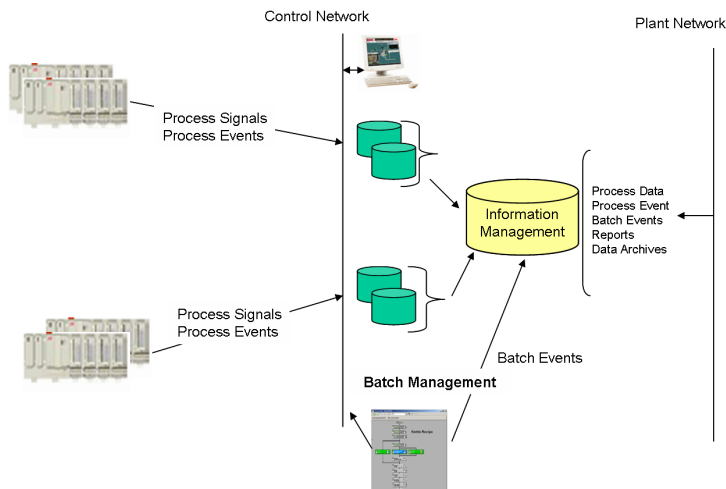


Figure 11. Overview of the System Dataflow

Planning Considerations for Dataflow

1. Consider how many Connectivity Servers will be used in the system.
2. Consider the amount of data to be collected in the connectivity servers and how much data will be distributed.
3. Consider if the 800xA Base system will be sufficient or if the system, for example also should handle Long Term storage, central archiving and advanced services, provided by one or several Information Managers? Table 5 below lists some of the feature differences between the Base system and the extended Information Manager system.

4. Consider the reporting requirements as part of the planning process. This is especially important for Production Management, since most of the work required is developed while creating the batch recipes.
5. Determine the primary data user. Is it operators for trending and even presenting, desktop users will the data be used for reporting applications.

Table 13. Key Feature Comparison

Base System	Information Manager/Optional Features
Short Term History Services <ul style="list-style-type: none"> • Process and event data collection 	Long Term storage for <ul style="list-style-type: none"> • Process Data • Event Storage • Reports
Reporting Services <ul style="list-style-type: none"> • Excel add-ins • Scheduling • Templates 	Structured storage for production data <ul style="list-style-type: none"> • Batch and manufacturing data using Production Data Log (PDL)
Application Services <ul style="list-style-type: none"> • Soft-points • Calculation engine (optional) 	Reporting Services <ul style="list-style-type: none"> • Additional Excel add-ins or Excel Data Access • Crystal Report
	Extended Archive Services
	Desktop tools <ul style="list-style-type: none"> • ODBC • Trending • Data Browsing

Data Collection

Process Data

Process data collection refers to the collection and storage of numeric values from aspect object properties. This includes properties for live process data, softpoint data and lab data.

Process data collection may be implemented on two levels in the 800xA system:

- The standard system supports operator trend data storage (short term) via Trend Logs.
- The Information Management History server function supports extended historical storage and permanent offline archiving (long term).

Log Templates

To implement process data collection, one or more History Log Templates must first be created. Each template serves as a model for a specific data collection scheme, for example rates for data collection and data storage. Log Templates are managed in the library structure.

Logs can be configured in two ways:

- online using the Plant Explorer Workplace. The Plant Explorer provides a graphical user interface for history configuration. This is the easiest method when adding log configuration aspects to just one or two objects.
- or offline via the Bulk Log Configuration Import/Export utility. The Bulk Log Configuration Import/Export utility uses Microsoft Excel to create a list of object properties, and then match object properties with their respective log templates. This tool is preferred when handling large number of object properties that use the same log template.

More information about basic Log Templates is found in the *System 800xA Configuration (3BDS011222*)*.

For information on Information Manager extensions to Log Templates refer to the *System 800xA Information Management Configuration (3BUF001092*)*.

Data Storage

Plan the layout of the disk drives to support the Information Management installation.

- It is recommended to use separate drives for program files and data files. Use separate drives for historical data storage.

Table 14. Disk Configuration

Disk No.	Content
1	System software only
2	Relational data, messages and production data (PDL)
3	Numeric process data (flat file)

- Consider which disk will be used for archiving (this can be a portion of an existing disk).
- Consider RAID disks for key data storage and backup.
- Remember to size the disks so that they will be 25% empty for defragmentation purposes.
- Multiple disks are recommended for better performance.

Make sure that Disk, memory, CPU and network can handle the planned configuration. After the configuration is worked out, reevaluate the hardware for integration purpose.

Planning Considerations For Trend Logs (Short Term Logs)

Storage space and loading calculations are performed according to several parameters. Consider the following when planning for a numeric log:

- Storage rate.
- Percentage of change.
- Storage type.
- Log period.
- Log size.

- Number of logs.
- Total disk size.

The configuration has to be planned so that the short term storage handles down time for the long term storage. This will allow the short term storage to back fill data that was collected by the trend log while the long term storage is offline.

- Decide if redundancy is needed. The trend log supports redundancy when parallel connectivity servers are configured.
- Calculations can be performed by desktop tools during data retrieval. However, reporting capabilities can be improved by storing calculated values rather than having the calculation performed on retrieval.

Planning Considerations For Long Term Logs

1. Consider the number of different properties to be logged.
2. Decide where the primary log will be located and how many logs will be serviced by that connectivity server.
3. Determine the aggregate to be collected. Reporting performance will be improved if common aggregates are already calculated.
4. Decide the rate of data change.
5. Decide for how long the data must be available online in the Connectivity Server.
6. If Information Manager is used, decide for how long the data will be online in the Information Manager server.
7. Determine if Dual logs are needed. The same Trend Log can feed two history logs on two different History Server nodes. Dual logs may be required when your application cannot wait for history data to be back-filled in the event that a History Server goes off line. For example, if shift reports are required to be executed at the end of each 8-hour shift, and you cannot wait days or weeks for the data to be backfilled.
8. Review how the stored data will be used and how long the data needs to be retained.
9. Name the log according to its function, to make it easy for users to find.

Event Data

Planning for Event data comes down to estimating the number of messages to be stored online. Depending on the number two options are possible:

- 800xA System Message Server collects and stores all alarm and event messages for the 800xA system, including Process, Operator, and Audit Trail messages. This provides a short-term storage facility with the capacity to store up to 10,000 messages per message category. The messages can be organized into filtered lists for viewing using the standard operator display event viewer, filters and Microsoft Excel.
- If Information Management History Server is installed, the messages stored by 800xA System Message Server may be forwarded to an Information Management message log for extended online storage. The messages are stored in a relational database and can be accessed by commercial tools for viewing. This message log can store up to 12 million messages.

Production Data

Production Data Logs (PDL) is used in conjunction with Batch Management. PDL organizes critical process data such as operator interventions, alarm and events, equipment usage and task start/stop and duration times. PDL provides special data structures for storing logically organized production data. It is typically used for batch records and is used in conjunction with Production Management (Batch Management).

PDL data is stored in a relational database and is accessible by standard SQL queries. The information is available to Crystal Reports, other network-based report packages, and applications such as Microsoft Access and Excel. Information from PDL is used as the data source for batch-to-batch displays and for standard batch reports.

Planning Considerations for Production Data

- Determine the number of batches to be stored online.
- Decide what actions should occur when the PDL is full.
- Plan an archive strategy for the PDL.
- Have in mind that the size of the message log will impact the need for memory and query response time.

Memory for Event and PDL Storage

While most of System 800xA run in 32-bit mode, Information Management uses the 64-bit version of oracle. This means that oracle has full access to memory above 2GB. For very large Event and PDL storage applications, IM performance can be improved by purchasing additional server memory and increasing the oracle memory usage. If one or more of the following conditions are true, additional memory should be considered for the server.

1. High steady state rate event storage/PDL storage that is close to the system performance limit for the system.
2. The maximum event capacity configured for the IMMSGLOG and PDLMSGLOG.
3. Have significant on-line storage of Production Data Log information.
4. Rely on reports to other access mechanisms to access all the data stored in oracle and desire the fast performance possible.

The actual additional memory required depends on all these factors. If additional memory is utilized, an oracle parameter must be modified to use the additional memory. For more information, refer to *System 800xA Information Management Configuration (3BUF001092*)*.

Retention and Recovery

Archives, Database backup (hsbar) and Connectivity Server configuration determine how much of an outage your system can recover from.

Connectivity Server Data Retention Online

Connectivity Servers provide the first level of data storage. The long term storage log configured for storage of the information feed from the data collected in the Connectivity Server and will backfill from this server after downtime.

The amount of data to be stored in the Connectivity Server should be based on plant operations needed to recover from a hardware failure in the Information Manager or for standard maintenance.

The archive rates should be set at a rate which is faster than the potential loss of data in the Information Manager as well as in the Connectivity Server.

Information Manager Server Retention Online

The Information Manager Server retention is based on the Log Templates configuration. This configuration decides how you want to archive.

Archive Rates

- Process Data Streams - determine groups and the archive rate for the group.
- Message Logs - determine groups and the archive rate for the group.
- Production Data - determine archive rules, after batch complete, delays. Determine when production data can be removed from the system after archive.

Recovery Strategy

The Database backup tool (hsBar) provides a base snapshot of the running history database. A backup of the runtime database should be taken periodically to augment the archives and the data storage duration of the Connectivity Servers. The archives will contain all the data, but requires the data which has been removed to be published back to the system, if a system failure occurs.

The best method to recover from a hard failure back up utility for example GHOST, is to restore a snapshot created by the hsBar tool. The data will then be backfilled from the connectivity servers.

It is recommended to test the backup routine occasionally to make sure that the recovery can be handled properly.

Archive Strategy

The archive function supports permanent offline storage for:

- Numeric process data stored in history logs
- Finished reports scheduled and executed via Scheduler, and stored in report logs, or stored as File Viewer aspects.
- Production data from batch control applications and stored in Production Data Logs (PDLs).
- Event, system and- Audit trail messages and generated by 800xA system applications and stored in message logs.

Archiving is managed by one or more archive device aspects which are configured in Node Administration Structure. An archive device is a logical entity that defines where and how archive data is written. Every Magnetic/Optical or disk drive used for archiving must have at least one archive device aspect configured for it.

- To prevent automatic overwriting of stored data it is recommended to archive more sensitive data through a separate device which is specifically configured to prevent overwriting.

Moving Information to Secure Offline Storage

There are different configuration possibilities for the archiving:

- Archiving may be scheduled to occur on a periodic or event-driven basis through the Scheduling Services.
- Manual archive operations on an ad-hoc basis can be executed on demand.
- Production Based archiving according to Batch execution defined as part of the Batch recipe.

Making Archived Data Available

Archived data is available for desktop tools like Display, Data Direct and Desktop Trends and for 800xA operations such as operator trends, event filters and reports.

Numeric logs, which are the most common, can be published directly from their current location. Event and Production data must be restored.

Archive Guidelines

Plan the archive function to ensure reliable and predictable operation. Consider the quantity and storage rate for History logs and size of archive media.

1. Determine how often the archive media is to be changed
2. Determine the Archive destination, Magneto Optical Platter or Disk. Disk Archive is recommended for most situations.
3. Determine the size of the archive device and the number of volumes. The volume consumed is based on the logs to be stored as well as on the rate of data change.

4. Plan the configuration of archive groups. The key to determining the correct number of archive groups is calculating the optimum size of an archive entry. The amount of data archived at any one time should be substantially less than the maximum capacity of the archive media. This will compensate for the possible accumulation of archive data in the event that a reasonable number of archive operations are missed. On the other hand the amount of data should not be so small that the archive media is used inefficiently.
5. Determine what will happen when the archive destination is full, for example wrap when full, stop etc.
6. Determine if shadowing will be used to move the data to a networked media. Shadowing is recommended.

Reporting Services

Reports are created using Crystal Reports or Data Direct which is a plug-in for Excel. A report is scheduled to run using the Application Scheduler and the Report action plug-in.

In addition to the Data Direct plug-ins for Excel, Excel as well as Crystal Reports can also be used with ODBC to communicate to the Information Manager. A custom ODBC interface has been provided to access process data as well as numeric logs. A standard Oracle ODBC interface has been provided to access message log and production data.

Report Templates

Example Report Templates for both Excel and Crystal Reports are provided in the system. The following reports are provided:

Table 15. Standard Report Templates

Type	Features
Trip	20 pre-trip and 10 post-trip values with 10 seconds between values
Trend	Hourly - one minute values Shift - eight one hour values Daily - one hour values per day Monthly - daily values
Event	One hour time span with filtering
Snapshot	Standard - 20 signal values from the historian Real-time - 20 current signal values

Report Destinations

Completed reports can be presented in several ways, for example they can be printed, sent by e-mail, saved in history or saved as a Windows file.

Scheduling

Versatile scheduling within the 800xA system makes it possible to automate not only repetitive jobs, but also initiate reactions to exceptional circumstances without the need of manual intervention.

A variety of scheduling techniques are provided:

- Event-driven.
- Cyclic.
- Time-based.
- On-demand.

The Scheduler lets the user schedule and run a variety of tasks including:

- Data Collection.
- Calculations.
- Reports.
- Archiving.
- Message log and PDL Consolidation.
- System Back-up.

Information Presentation/Desktop Services

The Desktop Services provides access to data in the system. The following services can be used:

Table 16. Desktop Services

Service	Description/Used For What Info?
Trends/Trending	Process Portal users can see all real-time and historical data using the standard operator trend displays.
Display Service	Desktop users process the ability to create process mimics as well as custom user defined forms.
Tag Ticker	Desktop users are provided with a user configurable tool to display current value information in a stock ticker format.
Tag Explorer	Desktop users are provided with a user configurable tool to manage tags and look at current values. The data managed in this tool can be exchanged with other desktop tools.
PDL	Desktop ad-hoc query browser designed for production data.
Event Browser	Desktop ad-hoc query browser designed for event data.
SQL Browser	Desktop ad-hoc query browser designed for access to various database tables.

Table 16. Desktop Services (Continued)

Service	Description/Used For What Info?
Batch to Batch Displ.	Desktop ad-hoc query browser including trending capabilities for production data. Provides batch to batch comparison on a neutral time horizon.
Add-ins for Excel	Excel Data Access add-ins provide tools for reading and writing process values, history values, reading alarm/event messages, reading Production data from PDL tasks for 800xA Batch Management, Batch 300, or TCL and for doing SQL queries.

Application Services/Application Data Management

Application services provides additional capabilities for end user application deployment.

Calculations

Calculation Services let you configure and schedule calculations for aspect objects. Typically calculations are applied to real-time database objects, including both softpoints and actual process points.

Calculations can be triggered by changes to the input, or be scheduled to execute cyclically or at a given date and time. A calculation aspect may be applied to any aspect object such as a unit, vessel, pump or Soft point. Input can be any aspect object property and output can be any writable point in the system.

Calculation logic is written in VB Script.

For details on Calculations services, refer to the *System 800xA Information Management Configuration (3BUF001092*)*.

Softpoints

Softpoint Services extend the system's basic capabilities. Softpoints are executed in an application server and not in a controller. Softpoints behave like any other control object in the system and can have alarms, events, ranges and limits.

Once configured, the Softpoints may be accessed by information management applications as if they were actual process points. For example, Softpoint values may be stored in property logs in History Services.

Softpoints can be used for presentation in reports together with Excel or Crystal Reports. Also desktop tools such as Desktop Trends and Data Direct can read from and write to Softpoints.

The Softpoint Services software may run on any number of servers within an Aspect System. Each server can have up to 2500 objects and each object can have up to 100 signals. The total number of signals cannot exceed 25,000.

For details on Softpoint services, refer to the *System 800xA Information Management Configuration (3BUF001092*)*.

Section 7 Batch Management

General

Batch Management contains the batch process related functions. Batch Management is used to set up the batch process control system.

Batch Management Preparation

The first step in Batch Management configuration is for the process engineer to list the functions of the (plant equipment) hardware used for batch processing. The engineers segment the process, breaking it into units and shared equipment modules (SEM). A unit performs a function, such as filling a tank or mixing ingredients. The equipment has the capability to execute phases identified in the procedure. An example of a process diagram displaying unit and SEM boundaries is shown in [Figure 12](#).

Planning Considerations

A SEM also consists of hardware and software that work together to perform an operation. However, a SEM can not run phases, and there are no equipment phases associated with it. A SEM can be acquired by more than one unit, causing the batch manager software to treat it differently than a unit. An example of a SEM is a set of pipes that serve more than one tank.

Once the process engineers have decided how the process will be organized, a name is assigned to and a corresponding configuration is created for all batch equipment in the controller. The equipment database contains information about the equipment, including what phases can be executed using each unit, the names of the parameters that would be used by the phases, and attributes of the equipment that may be needed when determining which will be used by a particular batch.

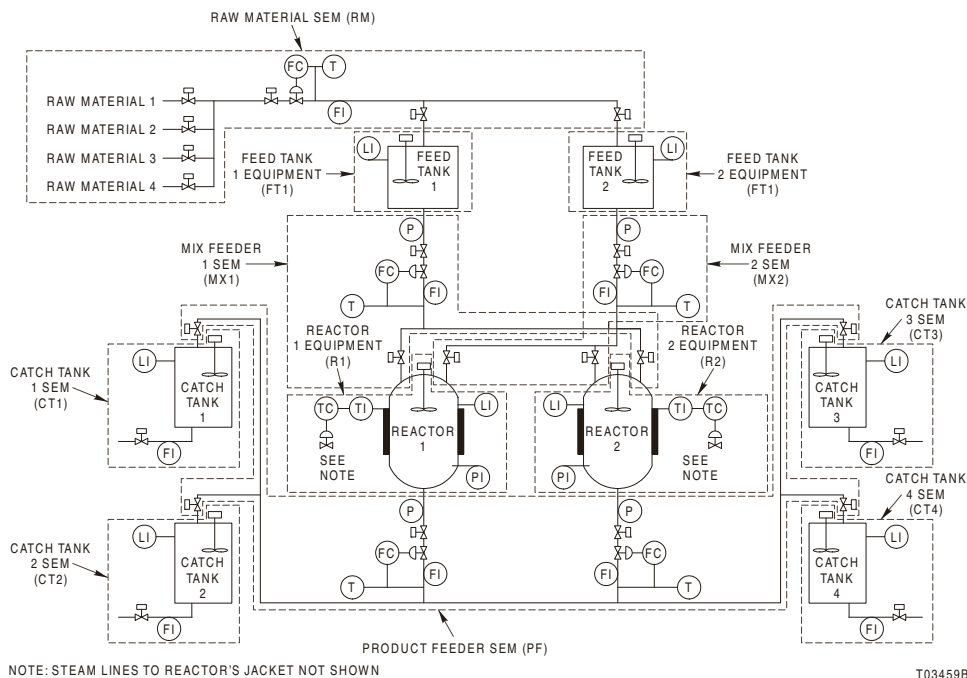


Figure 12. Process Diagram Example

A unit can be used by only one batch at a time. However, a SEM may be used by more than one batch simultaneously, depending on how it is defined in the equipment database. If the maximum number of users is one, the SEM is an exclusive use SEM. If two or more users can share it, the SEM is a shared use SEM.

The equipment database can also contain definitions for pseudo equipment. Pseudo equipment can be some resource required for batch processing, even though the resource is outside the control system. There are no phases associated with pseudo equipment. A procedure can be configured to reserve use of pseudo equipment, preventing another batch from running simultaneously if both batches require the same resource. For example, a system can be configured so the pseudo equipment represents a human operator.

Each unit must have an equipment phase logic associated with it residing in the controller. At run time, the names and parameters of the phase are downloaded to the controller by the batch manager.

The process engineers write the equipment phase logic required to accomplish tasks using the unit. For example, equipment phase logic can:

- Open valves.
- Transfer raw material from a feed tank to a reactor.
- Measure amount of material delivered to the reactor.
- Close valves.

Once the equipment database has been configured, the chemist or other product specialist can take over configuration. Procedures are created using graphical symbols that represent operations, phases, and batch manager actions. These symbols are combined into procedure function charts (PFC) as shown in [Figure 13](#). A PFC is a flow chart that depicts the batch's progress.

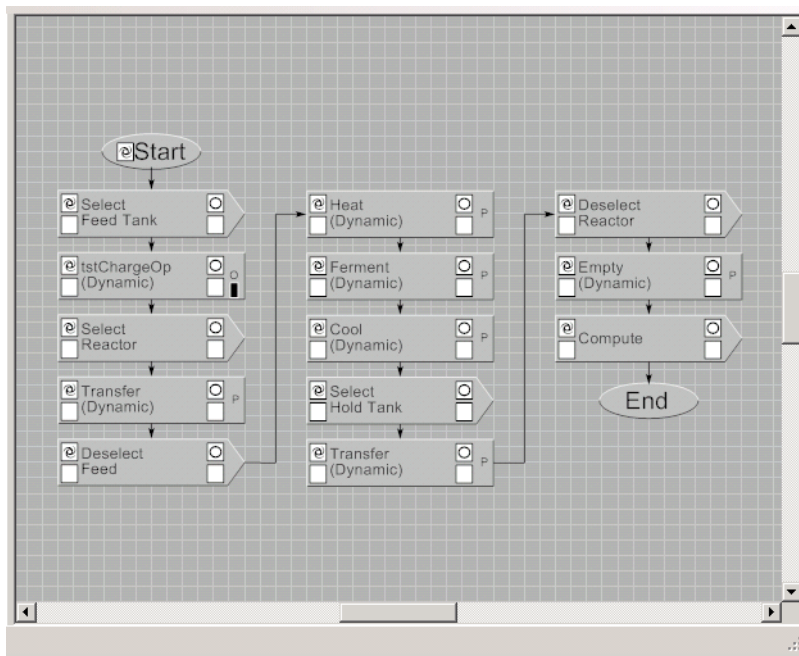


Figure 13. Procedure Function Chart

The main concept of procedure design in the batch manager is to create procedures out of reusable components. The process engineer configures phases that perform certain tasks such as moving material from a feed tank or heating a reactor. The product specialist can use these phases repetitively in a variety of combinations. In addition, the PFCs representing procedures can themselves be reused (alone or inside another procedure).

At run time, a selected master recipe is copied to be used as a control recipe. The appropriate phase parameter values are passed to the equipment phases in the controller subsystem.

Section 8 Device Management

General

The engineering of the Device Management is typically divided into a preparation phase, a configuration phase and a commissioning phase as it is also known from other parts of the system.

Throughout this section several documents are referenced. The following documents are available in the ABB Library:

- *AC 800M PROFIBUS DP Installations (3BDS009029*)*.
- *PROFIBUS - Refer to System 800xA Installation and Upgrade Getting Started (2PAA111708*)*.
- *FOUNDATION Fieldbus - Refer to System 800xA Installation and Upgrade Getting Started (2PAA111708*)*.
- *Field IT - Linking Device Integration Pepperl+Fuchs Power Hub (2PAA102122*)*.
- *RLM 01 Redundancy Link Module for PROFIBUS DP/FMS, Data Sheet (3BDD011641*)*.
- *Field Barrier and Power Conditioner System - Transition to Limited (2PAA104172*), Product Update*.
- *System 800xA System Guide Technical Data and Configuration (3BSE041434*)*.
- *System 800xA Device Management FOUNDATION Fieldbus Configuration (3BDD012902*)*.
- *AC 800M PROFIBUS DP Configuration (3BDS009030)*.
- *AC 800M FOUNDATION Fieldbus HSE (3BDD012903*)*.

- *System 800xA Device Management PROFIBUS and HART Configuration (3BDD011934*)*.
- *System 800xA IEC 61850 Configuration (9ARD171387*)*
- *System 800xA Asset Optimization Configuration (3BUA000118*)*.

The document AG-181 - System Engineering Guidelines, Fieldbus Foundation can be freely downloaded from the official Fieldbus Foundation website. (<http://www.fieldbus.org>)

Preparation

In the preparation phase at first it is necessary to decide which field protocol or a combination of HART, PROFIBUS and/or FOUNDATION Fieldbus - to be used. After this decision the field topology has to be planned. The installation of the hardware as well as the necessary engineering environment follows. Field Device Libraries with predefined object types for field devices reduce the engineering efforts enormously.

Decision on Protocol Type

Within System 800xA the usage of HART, PROFIBUS as well as FOUNDATION Fieldbus is supported. In most cases the customer has already decided which fieldbus protocol to use. However, the following table lists a couple of criteria in dependency of the protocol and its weighting.

Table 17. Decision on Protocol Type with System 800xA

	HART	PROFIBUS	FOUNDATION Fieldbus
FDT/DTM	Yes	Yes	No
OPC Access	Yes	Yes	Yes
Speed	Low	Fast	Fast
Intrinsically Safe	Yes	Yes	Yes
Control in the field	No	No	Yes

Table 17. Decision on Protocol Type with System 800xA (Continued)

	HART	PROFIBUS	FOUNDATION Fieldbus
Asset Optimization	Yes	Yes	Yes
Calibration	Yes	No	No
Redundancy	Yes	Yes (not PA segments, but redundant power supply modules possible)	Yes (not H1 links)
Staff Skills	normal	special Training	special Training

All three protocol types can be used in parallel at System 800xA. If the decision is made for FOUNDATION Fieldbus also PROFIBUS Remote IOs are often used for binary inputs and outputs. Most combinations of FOUNDATION Fieldbus, PROFIBUS and HART installations are used in process industries.

Topology Planning

The topology planning is based on piping & instrumentation (P&I) diagrams, tag lists and process descriptions. Also information about the site architecture is required to determine the cable length, usage of routers etc.

The main criteria to design the fieldbus topology is the possible amount of devices connected to the fieldbus segments. Sure, there are also process related requirements but these cannot be considered in this document.

There are different topologies for the individual fieldbus protocol.



Attend trainings to get familiar with PROFIBUS and FOUNDATION Fieldbus technology.

PROFIBUS Topology

Figure 14 shows an example PROFIBUS topology.

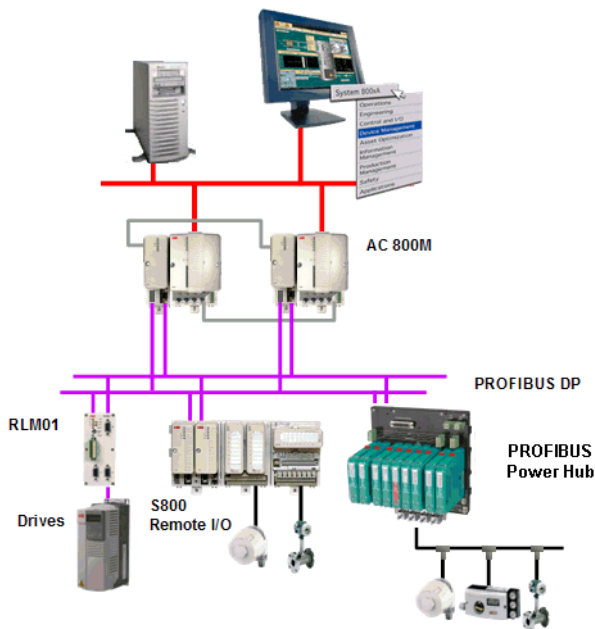


Figure 14. PROFIBUS Topology

PROFIBUS DP

A PROFIBUS line is built by using the PROFIBUS-DP/V1 interface module CI854(A). Redundancy by using two of them is possible. PROFIBUS can handle the address range from 0 to 126 at each line. At address 1 the PROFIBUS master is default assigned and it can be changed. Address 0 is reserved for a master class II. If PROFIBUS devices are not ordered with a preconfigured device address, they leave the factory with the default address 126. This address cannot be used for the standard operation but for connecting new devices to the bus. It has to be changed after bus connection. So, finally 124 field devices (address range from 2 to 125) can be connected to one PROFIBUS DP line.

If redundant devices should be connected a fixed addressing for the redundant device is used. The redundant device gets the address from its primary plus 64.

Example: Primary device address: 20
 Redundant device address: 20+64= 84

For later extensions ensure that the address of the planned redundant device is not already used by another device. In this case the address of the existing device has to be changed. The amount of free addresses will be reduced accordingly.

If more than 32 field devices are to be connected, the PROFIBUS line has to be separated by means of building segments with usage of repeater. A maximum of 8 DP segments can be built. Each connected PROFIBUS Power Hub acts as a gateway between DP and PA segments (Figure 15).

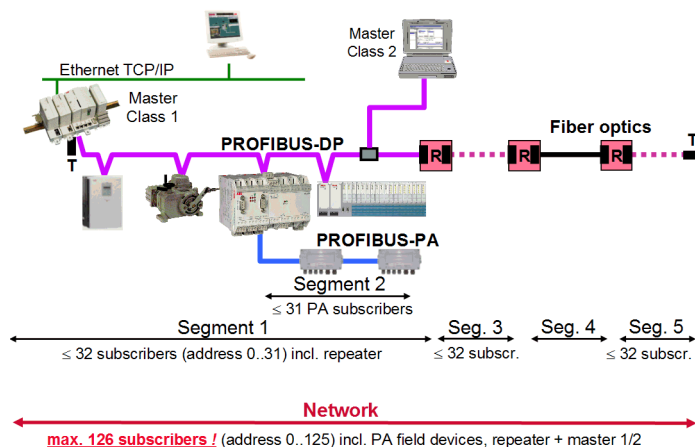


Figure 15. PROFIBUS Segmentation

The site architecture allows to calculate the required segment length to reach the connected devices. There exist a dependency of segment length and transmission rate. The transmission rate together with the amount and type of devices are the basis for calculation of the cycle time, which can be reached for a specific PROFIBUS segment (Table 18).

Table 18. Dependencies of Bus Length and Transmission Rate for Cable Type A

Transmission rate in kbit/s	9.6	19.2	93.75	187.5	500	1500	12000
Range in m/segment	1200	1200	1200	1000	400	200	100

Devices which transmit lots of information need more time for communication. The PROFIBUS master triggers every device according to the bus address. If there are lots of devices the cycle time will also be slowed down. The consequence is to increase the transmission rate which on his part shortened the segment length. Then routers has to be employed to refresh the PROFIBUS communication.

The possibility to switch to the next higher/lower transmission rate depends on the ability of the connected PROFIBUS devices: not all devices support all transmission rates. This and further communication parameters are defined in the GSD file of PROFIBUS device. It is ASCII readable and will be automatically interpreted by the PROFIBUS master to calculate the bus parameter. The slowest device speed will be taken.

To simplify the process of designing the PROFIBUS topology without having a running PROFIBUS master the PROFIBUS Layout Tool enables the user to calculate the cycle time in a PROFIBUS-DP/PA network and provides accessory lists, data sheets and help text. For more information on PROFIBUS installations and configuration refer to the:

- *AC 800M PROFIBUS DP Installations (3BDS009029*)*.
- PROFIBUS Installation suggestions - 10/63-0.40.
- *AC 800M PROFIBUS DP Configuration (3BDS009030)*.

Redundancy can be established through the whole DP architecture. If there are devices with non-redundant DP interfaces and have to be connected to a redundant PROFIBUS DP network the PROFIBUS DP/FMS Redundancy Link Module RLM01 has to be connected in between. For more information, refer to *RLM 01 Redundancy Link Module for PROFIBUS DP/FMS, Data Sheet (3BDD011641*)*.

PROFIBUS PA

PROFIBUS PA is a specific PROFIBUS profile. Its characteristic is the fixed transmission rate of 31.25 kbit/s and bus-powered 2-wire field devices such as transmitters and actuators. It corresponds to IEC 61158-2 - the new naming is MBP Transmission Technology (Manchester Coded, Bus Powered) - and can also be applied to intrinsically safe (IS) installations (EEx i) in Zone 1 / Div. 1. Refer to [Table 19](#).

Table 19. Limits of MBP Networks

	Intrinsically Safe	Non-intrinsically Safe
Measurement voltage	12,6 V ... 13,4 V	24 V ... 26 V
Measurement current	100 mA	400 mA
max. PROFIBUS PA slaves	10	32
max. line length	1000 m	1900 m

For more information, refer to the product documentation of the new solution of field barrier and power conditioner system as referenced in *Field Barrier and Power Conditioner System - Transition to Limited (2PAA104172*)*.

HART Topology

HART field devices can be connected to ABBs Remote I/O S800 and S900 as well as to the S800 local I/O, placed direct to the AC 800M Controller.

In principle the amount of HART devices connected to the above mentioned I/Os is not limited. When FDT/DTM or Asset Optimization functionality is used, consider the limitation of number of devices (refer to *System 800xA System Guide Technical Data and Configuration 3BSE041434**).

The access on HART device internal data is performed via OPC Server PROFIBUS/HART. This server has to be installed on a connectivity server. The maximum number of Connectivity Servers with OPC Server for PROFIBUS/HART or FOUNDATION Fieldbus installation is limited and depends on the maximum possible number of connectivity servers in total. This as well as the maximum

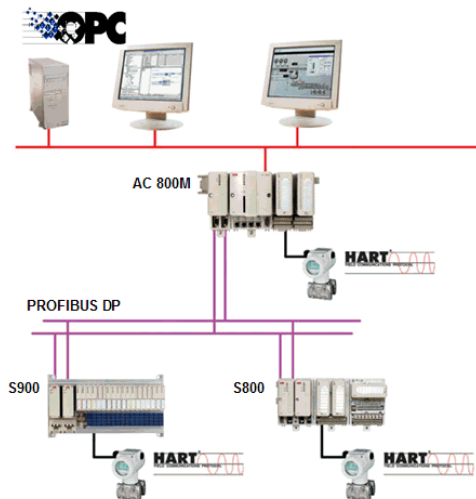


Figure 16. HART Devices, Connected to Remote and Local I/O

number of field devices per Connectivity Server differ for combined and separated node types.

Please check details listed in the *System 800xA System Guide Technical Data and Configuration (3BSE041434*)*, chapter Configuration Rules Summary, Control and I/O Rules.

Physically the Connectivity Server with OPC Server PROFIBUS/HART is linked to the Control Network and the Client/Server Network. In the configuration the OPC Server PROFIBUS/HART is assigned as default to the Control Network to collect all data from devices which are configured below this network. In the case, the fieldbus structure below a Control Network node does include more devices than supported by the OPC Server (refer to the *System 800xA System Guide Technical Data and Configuration (3BSE041434*)*, the OPC Server PROFIBUS/HART, together with all other OPC Servers (example AC 800M OPC Server) assigned to the Control Network, can be moved to the underlying Control Project node. The OPC server can be assigned to AC800M only up to the project node.

It is only allowed to run one OPC Server PROFIBUS/HART on a single Connectivity Server. So, the number of Connectivity Server will increase in this case. Above mentioned limits have to be considered.

FOUNDATION Fieldbus Topology

Within System 800xA a FOUNDATION Fieldbus subsystem is built by HSE subnets (HSE = High Speed Ethernet) based on standard Ethernet physical layer (Figure 17).

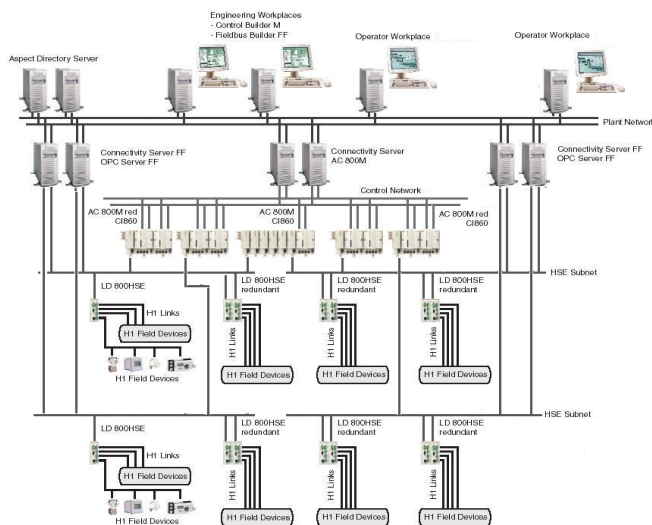


Figure 17. LD 800HSE in the FF topology

ABB's LD 800HSE Linking Devices serve as gateways between the HSE subnets and the slower H1 links where finally the FOUNDATION Fieldbus devices are connected. If necessary the interface module CI860 provide the linkage of the AC 800M controller to the HSE subnet.

The FOUNDATION Fieldbus specification is uniquely different from other networking technologies in that it is not only a communication protocol but also a programming language for building control strategies which are distributed into the field devices.

Typically the FOUNDATION Fieldbus devices perform simple regulatory controls. The AC 800M controller may perform discrete logic and sequence control as well as advanced control. In this case the number of controller can be drastically reduced and in some cases eliminated altogether.

Field-level Topology

The field-level topology is mainly driven by the capability of the Linking Device LD 800HSE. It has one HSE link and four H1 links. The maximum number of Linking Devices connected to one HSE subnet as well as the maximum number of field devices connected to the H1 links is limited.

Please check details listed in the *System 800xA System Guide Technical Data and Configuration (3BSE041434*)*, chapter FOUNDATION Fieldbus Configuration Rules.

Every field device can transmit one or more process signals. Maximum 100 signals can be physically transmitted between one H1 link and HSE.

The maximum throughput is limited to 40 signals per second. Considering the function block runtimes and H1 Link bus parameter settings this limit will only be reached using, for example, multiplexer devices providing 8 or more process signals each with a short H1 macro cycle.

The Fieldbus Foundation issued a System Engineering Guideline (refer to AG-181 - System Engineering Guidelines, Fieldbus Foundation). This guideline has been prepared under direction of the End User Advisory Council (EUAC) of the Fieldbus Foundation. The chapter 6.4, *FOUNDATION Fieldbus Segment Risk Management*, gives some suggestions how to design the FF topology to minimize single points of failure. One recommendation is that as default not more than 12 devices should be placed in one H1 segment and not more than 4 valves.



Refer to the chapter 6.4 ff of AG-181 - System Engineering Guidelines, Fieldbus Foundation to get familiar with FOUNDATION Fieldbus Segment Risk Management

Controller-level Topology

In such cases where the FOUNDATION Fieldbus is used as a pure I/O bus or if a process signal exchange between controller and HSE subnet is required, for example, for advanced control the FOUNDATION Fieldbus HSE Interface CI860 is necessary. This module provides channels for analog and digital inputs and outputs to the HSE subnet as well as the capabilities to provide acyclic client/server communication for access to FF block contained parameters. For more information, refer to *AC 800M FOUNDATION Fieldbus HSE (3BDD012903*)*. Check the limits

and the maximum throughput of the CI860 in *System 800xA System Guide Technical Data and Configuration (3BSE041434*)*, chapter Fieldbus, table *FOUNDATION Fieldbus basic layout rules*.

If the amount of signals or the possible throughput exceeds the limits further CI860 can be used. Maximum 12 CI860 can be connected to one controller. CI860 redundancy can be established by simple using two of them. A redundant pair of CI860 count as two of the 12 maximum possible.

IEC 61850 Connect System Topology

The IEC 61850 Connect integrates the IEC 61850 network with 800xA system. The solution is based on the Standard Connectivity functionality in 800xA where the subsystems are integrated to the 800xA system using the OPC Servers (Data Access and Alarm and Event). The IEC 61850 Connect uses the IEC 61850 OPC Server.

This section gives an overview of IEC 61850 Connect. The IEC 61850 object type library contains the logical node objects that are specified in the IEC 61850 standard. It also includes additional generic objects and functional objects such as breakers and transformers, which represent the substation functional view in the 800xA. These object types act as a database for creation of Communication and Substation section of SCD file in 800xA. The Uploader creates Control and Functional Structures in 800xA system automatically by reading the Communication and Substation section respectively in the SCD file. The IEC 61850 OPC Server provides data from IEC 61850 network to 800xA.

Refer to *System 800xA System Guide Technical Data and Configuration (3BSE041434*)*, for performance and configuration limitation of IEC 61850 Connect System Topology.

Figure 18 shows the IEC 61850 Connect System Topology.

The IEC 61850 Connect facilitates the vertical communication as defined in the IEC 61850 standard. So, only the MMS (Manufacturing Message Specification) based signal on the IEC 61850 bus is read by the IEC 61850 Connect and given to the 800xA client. For more information about the IEC 61850 Connect, refer to *System 800xA IEC 61850, Configuration (9ARD171387*)*.

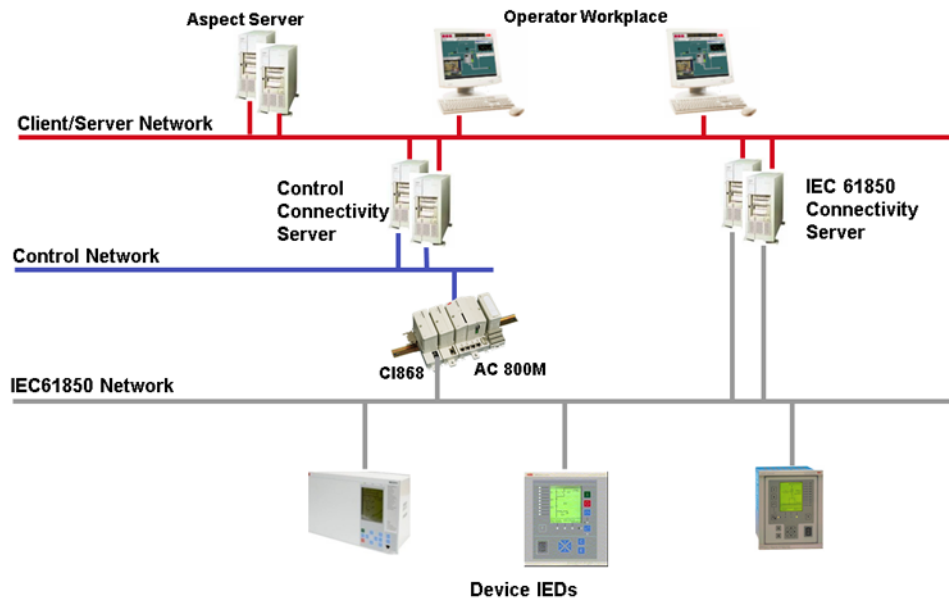


Figure 18. IEC 61850 Topology

Installation

Hardware

90% of the problems occurring in the fieldbus area are based on hardware installation and wiring mistakes. It is strongly recommended to read the *AC 800M PROFIBUS DP Installations (3BDS009029*)* and *PROFIBUS - Installation suggestions - 10/63-0.40*, as well as *AG-181 - System Engineering Guidelines, FOUNDATION Fieldbus*.

Software

Refer to the installation guide to install the relevant software for the Device Management on the different node types.

The technology of digital fieldbuses (PROFIBUS, FOUNDATION Fieldbus, and also HART) allows to have access on field device internal data and information. This access can be used for Asset Monitoring, calibration, faceplates, DTMs, and EDDL views for configuration and diagnosis, and so on. For efficient engineering the field device objects have to be well prepared to offer such functionality as well as for CMMS interfaces and access on manuals.

The installation DVD contains field device libraries for all three protocol types provided by the Device Integration Center (DIC). For a specific project not all field device objects are required. Only those which are needed have to be installed with the Device Library Wizard whereas only the relevant aspects according to the installed system extensions will be considered. The installation of the required field device objects has to be done on all nodes where access on field devices with for example FDT/DTM is necessary.

If a relevant system extension has to be added later on (for example for Asset Optimization) the already installed field device objects has to be updated via the Device Library Wizard. In this case the missing aspects will be added to the existing field device objects.

The installation DVD reflects the status of device integration at the date of issue. In addition newer field device objects can be downloaded from the ABB Library as well as from the ABB SolutionsBank.



Lists of currently integrated field devices are available at www.abb.com/ProductGuide/.

Follow the link, **System > 800xA > Device Management ... > Device Integration Center**.

Release Notes to every field device object are available where the result of integration is described. It is recommended to check this first to ensure that the integrated field device meets the project needs. The amount of integrated field devices is continuously growing. If a required device is not yet integrated send a mail to DIC@in.abb.com to inform the Device Integration Center (DIC). DIC will take this information to put this into the list of integration plannings.

Configuration

Usage of Preconfigured Devices

Intelligent field devices are capable to be adapted for different process needs. To adapt devices to the specific process need the device parameter has to be adjusted.

Typically devices will be ordered and delivered from the factory already well parameterized. There is no need to configure devices offline by means of device DTMs in the planning phase. Later, when the controller configuration is loaded and the field devices are connected to the 800xA system, the parameter upload is necessary to have these preconfigured device parameter available in the engineering system environment (refer to [Usage of Extended Status and Diagnosis](#) on page 137).

Designation of Field Devices

To be able to communicate with PROFIBUS devices unambiguous node addresses are needed at the PROFIBUS segment. The node address is specified when the field device is created. If a system alarm occurs the alarm message contains beside the error code also the device path built by the node addresses of all parent objects and the own node address. This is sufficient for error tracking via for example Control Builder M.

PROFIBUS devices are created by Control Builder M or Plant Explorer. In both cases the node address is defined with the name property of the Name aspect. The Control Builder Name aspect finally holds the node address as reference and will automatically get the value from the Name aspect during creation. Directly after creation both aspects have the same name property value.

The name property of the name aspect should be renamed. Otherwise messages originated by for example Asset Optimization displayed via alarm and event lists, are not unambiguous when more than one fieldbus segments are used. The device path built by the node addresses of all parent objects is missing. Only the Name aspect is displayed. The Name aspect should be renamed with the target tag of the device. The Control Builder Name aspect still holds the node address. The same has to be done for PROFIBUS device modules where for example Asset Monitors (Channel Asset Monitors) are attached.

FOUNDATION Fieldbus devices will be created by Fieldbus Builder FF (FBB FF) or directly in Plant Explorer, for example, through Bulk Data Manager (BDM).

The given device name will be taken to generate automatically the names for the resource and transducer blocks. The following example shows the naming pattern:

- Device name - TI4711
- Resource Block name - TI4711_RB
- Transducer Block 1 name - TI4711_TB1
- Transducer Block 2 name - TI4711_TB2

The function block names and the short texts are displayed in FF faceplates. Names and its descriptions can be changed in the Plant Explorer, but they will be overwritten during the next automatic Fieldbus Builder FF upload.

HART devices will get their names during the creation process. This is the normal object handling within the Plant Explorer without any difficulties/problems.

Usage of Bulk Data Manager

HART, PROFIBUS, and FOUNDATION Fieldbus devices can be created with the Bulk Data Manager (BDM).

HART devices can be instantiated in the control structure (and simultaneously in other structures) with a defined name and the HART tag, if it is not write-protected by the DTM. Channel assignment needs to be done either by the Fieldbus Management aspect manually or via BDM sheet.

PROFIBUS devices consist of a slave object for address assignment and communication modules. The initial instantiation of PROFIBUS devices must be done in the Control Structure but can be inserted in other structures afterwards. A BDM sheet can be used for instantiation and for the renaming process as described in [Designation of Field Devices](#) on page 134).

FOUNDATION Fieldbus Topologies can be created with the BDM. This includes all FF objects such as, HSE subnets, HSE Host CI860, Linking Devices, H1 Links, H1 Devices, H1 Schedules, and Function Block Application as copies from Templates. Bulk Data Manager can also be taken for parametrization of, for example, the range or alarm settings of Function Blocks or the network settings of FF topology objects. Additionally, the assignment of function blocks to FF devices and signals to HSE Host CI860 modules can be done with BDM. For more

information, refer to the *System 800xA Device Management FOUNDATION Fieldbus Configuration (3BDD012902*)*.

Usage of Function Designer

Further increase of efficiency can be reached by the usage of Function Designer. The usage of Function Designer presumes to be able to create objects out of the Plant Explorer. This is currently not the case for FOUNDATION Fieldbus devices. Furthermore it presumes to initially create the objects in the Functional Structure. PROFIBUS and HART devices can be efficiently engineered with Function Designer.

FOUNDATION Fieldbus devices can be inserted in Function Designer but requires additional configuration steps.

Usage of Extended Status and Diagnosis

The main differentiator for the usage of fieldbuses against conventional I/O is to have more information available via the extended device status and device diagnosis.

For HART, a device diagnosis is available via acyclic communication provided by DTMs. Asset Optimization takes this information via OPC server for monitoring the device conditions.

For PROFIBUS devices a device status (HWStatus) and an extended status (Extended Status) exists. Based on the specification done in the hardware definition file (HWD) diagnosis is available. The predefined device types coming from the Device Integration Center provides the most common used diagnosis information. The release notes coming along with the downloadable device object type in the ABB library provides the information at which bit position which diagnosis is available. Within Control Builder M the current diagnosis information are displayed in the Unit Status tab after opening the Hardware Editor window. In the Connections tab a variable can be assigned to the Unit Status to make the information available in the IEC 61131 control application.

Each PROFIBUS signal provided by the modules consists of two elements: the value and the status. The status element contains additional validation information about the value. In opposition to PROFIBUS DP, where currently no standards exist the status for PROFIBUS PA devices is standardized. It is divided into three parts:

the quality, the quality substatus and the limit condition. The quality is the general validity of the value: good, uncertain or bad. The quality substatus depends on the quality status and explains the situation with a finer granularity. The limit condition indicates if the value is high-, low- or not limited or is fixed as a constant. If the status is greater or equal to the value 128, then the process value has a good quality. After assigning a variables to the status it can be used in controller applications.

FOUNDATION Fieldbus signals contain value and status information as well. The status composition is the same as for PROFIBUS PA signals: divided into the same three parts with quality, quality substatus and limit condition. The difference to PROFIBUS PA is that the FF status is mapped into one single process variable and can be used with already defined FF specific control modules and function blocks. For more information, refer to the *AC 800M FOUNDATION Fieldbus HSE (3BDD012903*)*.

The FOUNDATION Fieldbus device diagnosis is available through the Fieldbus Builder FF supporting enhanced EDDL. The OPC Server FF provides these information for Operation and Asset Monitoring.

Section 9 Connectivity to External Data Sources

Common Data Integration

Integrating data into the system 800xA can be made in several ways, each way has its own complexity and functionality. This section is intended to work as a guideline for which method to choose, but also provides useful information of how to best use each integration method.

The following three methods of integration are supported and described. They can be divided into two major sections, direct and indirect integration.

Direct integration to 800xA:

- [PLC Connect](#).
- [Generic OPC](#).

Indirect integration to 800xA:

- [Controller as Gateway](#)

The direct integration handles larger data amounts and incorporates them directly into the System 800xA. The indirect integration requires different pieces of hardware in the process which incorporates the data. The data is then visible for 800xA through this device (controller or panel).

Methods Overview

PLC Connect is the most powerful way of integration which fully incorporates third party data in the system. The system extension PLC Connect can be engineered to fit any process structure and since a real time data base is added it is designed to handle massive amounts of data. PLC Connect requires an initial configuration phase for type design if the upload function is not used. Data can be

modified and altered to suit the need of operation by configuration of aspects. PLC Connect can handle serial-, OPC-(DA and AE) and dialled communication

On the operator level, there are very few differences compared to an AC 800M integration. The main strength is the flexibility and the rich set of functionality, making it compatible with most makes of PLCs'. All solutions are reusable between projects thanks to the object type design. PLC Connect is also the preferred way of integrating older system offerings like SB2 controllers and Satt.

Generic OPC is a built in function in the 800xA base system and is another way of direct integration. A sophisticated OPC client enables direct upload of data from OPC servers. Data can be modified and altered to suit the need of operation by creating expressions, in a simple expression language, for each data property. The Generic OPC is only usable together with OPC servers and does not comply where serial communication is needed. On operator level the Generic OPC shows very few differences compared to an AC 800M integration. The main strength is that it is very simple to incorporate static OPC DA and AE data into the system.

The **AC 800M** controller can be used as a gateway for communicating with devices that have support for the same communication protocols, for example remotely distributed RTU's or PLCs. It can be used for combining typical distributed installations into an integrated DCS system. By using the AC 800M controller as gateway between the remote RTU's objects are presented in the system as if they would have been integrated into the system in an AC 800M controller.

PLC Connect

Overview

PLC Connect is a connectivity option to the Industrial IT 800xA that makes it possible to connect and integrate any type of remotely or locally installed PLC, RTU or other type of device. The information from the connected unit is treated in the same way as information from other 800xA System connectivities. PLC Connect is an object oriented PLC-server with built in functionality for alarm detection (Boolean, integer and real), scaling and pretreatment of values. Besides OPC, serial communication protocols are also provided (COMLI, MODBUS and Sattbus). As an option, dialled communication for remote units can be used.

PLC Connect offers the highest level of integration of external data and enables system 800xA to work as a hybrid DCS/PLC system. Whatever appearance the external data have, it can be mapped into a user defined object structure in 800xA. Two functional ways are possible, either define object types and map the object structure to external data or upload all data automatically (only possible for OPC servers).

For more information on engineering refer to the *System 800xA PLC Connect Configuration (3BSE035041*)*. For the basic setup, refer to the step-by-step instructions in the *System 800xA Post Installation (3BUA000156*)*.

Strengths and Weaknesses

Strengths

- PLC Connect is highly scalable and can cope with all kind of system sizes. With a fully object oriented architecture the system can handle changes in an efficient way. A change in type will affect all object instances of that type automatically.
- It is compliant with any device which follows the communication standards for the provided drivers (OPC, COMLI, MODBUS (TCP and RTU), Sattbus or IEC 60870).
- It handles efficient and flexible upload of OPC server contents into the Control Structure. All items are automatically mapped to a matching object type (OPC items can also be manually associated with a type).
- PLC Connect can be defined and configured and handle a heavy, constant Alarm and Event load.
- It is easy to configure alarm- and signal logic via aspects. No code is required.
- It supports high reusability of object types from one project to another.
- It can create a logic Plant Explorer structure independently of the appearance of the connected device.

Weaknesses

- Serial communication cannot use the upload functionality. This can be solved by using an OPC server with a serial far end.

When and Where to Use

PLC Connect can be used for any type of external data integration. It is especially powerful during following circumstances:

- Dialed- or serial communication needed.
- The connected data is not object oriented and not well structured.
- Full redundancy is required for external data.
- Heavy, continues alarm- and event loads are expected.
- Pretreatment of signals are required.
- PLC Connect works just as well with or without an available OPC DA/AE server.

How to Use

It is important to spend time to think through and engineer the basic object types in an optimal way. A finished type library can easily be used in the next project requiring only a minimum of reconfiguration.

Engineering Considerations:

- Start configuring signal object types before any object types are configured.
- Always use the composite type, even if only one object type level is required.
- Name the signals of an object type according to OPC Server contents. This makes the type association more efficient for a filtered upload.
- Always use the PLC Connect provided faceplates.
- It is easy to change an already defined object type. A finished object type library can easily be reused for another application.
- Always strive to use the filtered upload function to populate the control structure.

- Bulk Data Manager can be used where the uploader is not applicable. Bulk Data Manager enables the system engineer to drag a structure from Plant Explorer to an Excel sheet. Inside Excel all necessary changes can be made, such as addressing signal objects and changing inheritance. It is also possible to create and delete objects, which is useful for populating or managing other structures in Plant Explorer.
- Redundancy in PLC Connect can be used both with OPC and with serial communication. The System Version 6.0 supports full redundancy on both server and communication level.
- When using redundancy with OPC servers, it is important to use synchronized OPC servers that are intended for redundant use. If synchronized servers are not used timestamps might differ from one server to the other which can cause differences in alarm and event lists between two connectivity servers.
- Carefully think through the communication failover criterion. The communication status supervision will order a failover if as little as one communication link fails to answer, even though all other links work correctly. If the failure is depending on a communication line break, the other connectivity server will resume communication.
- If there is only one PLC connected and this is broken, the Connectivity server will start oscillate. Refer to A: in [Figure 19](#).

Using serial protocols together with redundancy often requires Serial Port Servers to expand the number of communication ports of the server. The serial ports must be of the brand Digi RealPort^R in order to work correctly with PLC Connect. All usage of other third party hardware is not supported and software based port sharing between the two Connectivity Servers is not allowed as such a solution would inflict a communication error after failover ([Figure 20](#)).

Generic OPC

Overview

Generic OPC is a base function in Industrial IT 800xA. Generic OPC allows connection of any OPC DA and AE server to 800xA. The information from the connected unit can be treated in the same way as information from other 800xA data

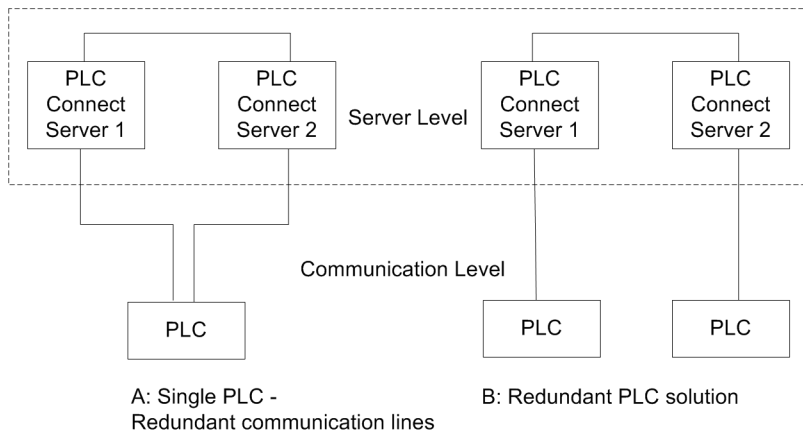


Figure 19. Different Redundancy Levels

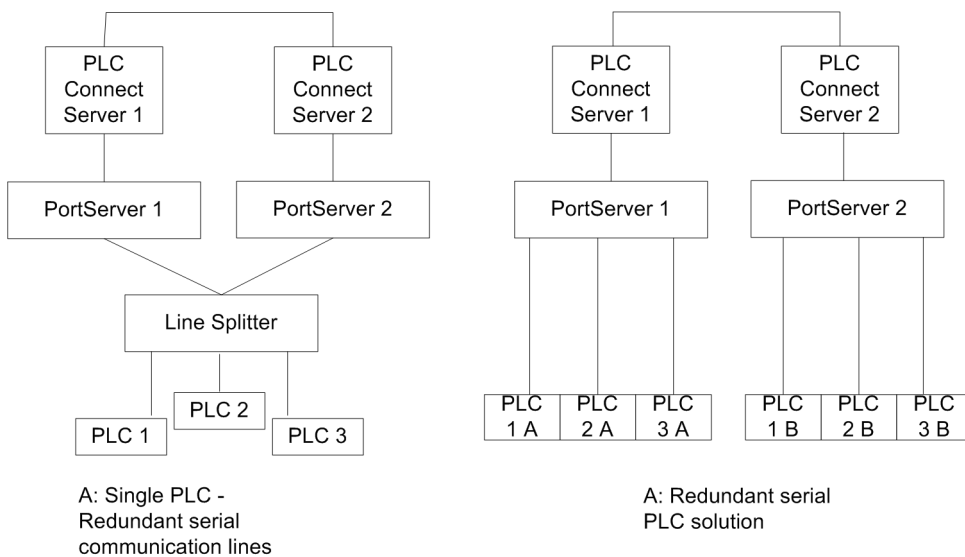


Figure 20. Serial Redundancy with Port Servers

sources. Standard 800xA aspects can be used to build project specific libraries containing Process Graphics and Face plates. Logs and trend displays can also be configured as for other 800xA connectivity.

The Generic OPC support in 800xA contains aspect objects for creation of an Generic OPC Server Network and a Generic OPC Object type.

All basic tools as Graphical builder, Bulk data manager and Document manager can be used with Generic OPC. The main tool for configuration in 800xA is Plant Explorer.

For specific information about the Generic OPC refer to the *System 800xA Configuration (3BDS011222*)*.

Strengths and Weaknesses

Strengths

- Generic OPC is an alternative for well structured OPC servers which requires little or no need for changes of the present structure.
- Handling and configuration work are very much the same as for AC 800M Connect.
- When single items of the available information in the OPC server will be used, predefined control connections and object types must also be used. General handling of Generic OPC is common with most connectivities in 800xA.
- Generic OPC works well together with OPC AE.

Weaknesses

- It requires a well structured OPC server contents to avoid that Plant Explorer gets unorganized and hard to work with.
- It requires to upload the entire OPC structure.
- A change in the structure requires a new complete upload of the OPC server contents.
- Configuration of signals and definition of alarms requires the use of a simple expression language.

When and Where to Use

Generic OPC can be used for most devices connected via OPC.

- The information in the OPC server is well structured and have internal connections between OPC DA and OPC AE via for example common source names.
- Other OPC servers that are engineered separately or as part of the device engineering benefits from using Generic OPC. Both these alternatives give a minimum of configuration in 800xA and the engineering can be further optimized if the OPC servers has a structure that can be directly mapped to 800xA Object Types.
- The Object types must be built for the device. In those cases the Functional structure can be built in advance and the OPC items can be mapped to this structure direct at upload.
- Generic OPC is most suitable when you have a few data sources and you can manage a larger effort to handle the connectivity.

How to Use

The benefits of using Generic OPC is very much dependent on the OPC server.

- Generic OPC gives more benefits when the engineering work of the OPC server can be reused in the 800xA, for example if the structuring and naming is done as part of the device and OPC server engineering. To do repeated uploads without destroying applications, the Functional structure can be built in two ways:
 1. By inserting uploaded objects from the Control structure into the Functional structure or be built on Object types complying with the objects in the OPC server.
 2. By joining the OPC items to the object in the Functional structure via name resolving. Name resolving is done on the Object name or the Name defined in the Controller name aspect on the Object in the Functional Structure.

Controller as Gateway

Overview

The AC 800M controller can be used as a gateway for communicating with devices that support the same communication protocols as the AC 800M. This paragraph contains an overview. For a detailed description refer to the *AC 800M Communication Protocols (3BSE035982*)*.

Possible protocols to use are:

- Advant Fieldbus 100.
- COMLI (Serial RS-232 interface).
- DeviceNet.
- EtherNet/IP.
- IEC 61850.
- MasterBus 300.
- MOD5-to-MOD5.
- MODBUS RTU (serial RS-232 interface).
- MODBUS TCP.
- MMS on TCP/IP.
- PROFIBUS-DP, the AC 800M controller can only act as a PROFIBUS-DP master.
- PROFINET IO, both cyclic and acyclic.
- SattBus on TCP/IP.
- Siemens 3694R (Serial RS-232 interface).
- Self-defined UDP.
- Self-defined TCP.

Strengths and Weaknesses

Strengths

- By using the AC 800M controller as gateway, RTU or PLC type of equipment can be integrated and presented into an 800xA system as any other objects, without the need for PLC Connect.
- Alarm and Event functions available in the AC 800M controller can be used to create timestamping in the controller for this data. It is also supported to create simple coding of time information and pack into registers in the serial protocols for giving the timestamp to the event block directly from the PLC/RTU.

Weaknesses

- The maintenance work, to make changes and additions, is slightly higher compared to a PLC Connect based system.

When and Where to Use

The AC 800M controller can be used for combining typical distributed installations into an integrated DCS system. When having a number of remotely distributed RTU's or PLC's these can be connected to the 800xA system by using the available communication protocols in AC 800M. The AC 800M will in this case serve as communication server and basically route the data from the remote protocol to the AC 800M Connectivity Server.

By using the AC 800M controller as gateway between the remote RTU's objects are presented in the 800xA system as if they would have been integrated into the system in an AC 800M controller. The data comes into the AC 800M controller very similar as to if it would have been I/O variables, and needs to be bundled together in objects to be presented in 800xA Operations.

How to Use

Most often this type of configurations requires some kind of converters. From RS-232 to RS-485 or even using real modems. The RS-232 type of interface is only recommended for peer-to-peer connections over short distances, while RS-485 can

be used for multi drop connections and even over distances reaching up to around 1.000 meters, depending on the baud rate.

When using PROFIBUS-DP, a coupler acting as slave in both ends is necessary, since it's uncommon that RTU's or PLC can act as PROFIBUS-DP slaves. PROFIBUS-DP couplers are available on the open market.

Section 10 Maintenance Workflow Strategy

Overview

When determining a Maintenance Workflow Strategy for 800xA consider the following questions:

- What Asset Optimization Functionality is needed?
- What is the Maintenance Concept?
- What are the Environmental conditions?

This section identifies usable planning considerations and asks questions to help analyze the requirements and make decisions when defining the Maintenance Workflow Strategy.

Maintenance Concept

Planning Considerations

Follow the planning considerations below when defining for the maintenance concept:

User Concept

1. What user groups shall access the maintenance data?
2. Define the hierarchy for Maintenance Users.
3. Define the needed User Groups.
4. Define the needed User.

Workplace Concept

1. Is a dedicated Maintenance Workplace required?
2. Identify the number of 800xA Client Licenses needed for the Maintenance Workplaces.

Structuring of Assets

1. Define the desired structure that shall contain all Assets.
2. Define the way of grouping the Assets.
3. Define the Asset groups in the Asset Structure.
4. Define the Assets for each group.

Asset Optimization Reporting Requirements

1. Define the alarming concept for maintenance alarms.
2. Define the paging concept for alarm receiver.
3. Define needed reports for Asset Data.
4. Identify if Scheduler is needed.
5. Identify if Information Management is needed.
6. Identify if SMS and e-mail Messaging is needed.
7. Define the needed reports.
8. Define the SMS sending device and the PC is shall be attached to.

Environmental Conditions

Use [Table 20](#) when defining the environmental conditions.

Defining these environmental conditions helps to define the Calibration Management functionality.

Table 20. Decisions of the Environmental Conditions

Environmental Conditions	Yes	No
Regulated Industry		
Enhanced Access Control		
Hazardous environment		
SIL Level		

Asset Optimization Functionality

Asset Optimization functionality includes Asset Health Condition Monitoring for predictive maintenance, CMMS (Computerized Maintenance Management System) Integration, and Calibration Integration to the 800xA System. Define the Asset Optimization functionality needed and follow the links for further definition.

Table 21. Areas for Asset Optimization Functionality

Area	No	Yes	Further Reading
Asset Health Condition Monitoring			If Yes, continue reading the paragraph "Condition Monitoring"
CMMS Integration			If Yes, continue reading the paragraph "CMMS Integration"
Calibration Management			If Yes, continue reading the paragraph "Calibration Management"

Asset Health Condition Monitoring

The Asset Health Condition Reporting system provides the infrastructure that reports asset status/condition information to notify operators and maintenance personnel when an abnormal condition calls for a maintenance action. Asset Optimization provides asset condition reporting via the Asset Viewer and Asset Reporter. The Asset Tree, visible in the Asset Viewer, shows the status of associated plant objects (assets) based on Plant Explorer hierarchies. Assets can be control system hardware

components, control system networks, control system devices, fieldbus components, machines, pumps, motors, process equipment (boiler, reactor), etc.

Condition Monitoring systems can report accessed asset conditions into the Asset Health Condition reporting infrastructure, Asset Viewer, and Asset Reporter. Condition Monitoring systems include Asset Monitoring.

If Asset Health Condition Monitoring functionality is needed, determine which of the following types of condition monitoring functionality are required:

- IT Asset Monitoring.
- Field Device Asset Monitoring.
- Process Equipment Monitoring.

Planning Considerations for IT Asset Monitoring

1. IT Asset Monitoring requires one PC, Network, and Software Monitoring (PNSM) license per system.
2. Decide the types of IT Assets to be monitored:
 - PC Nodes
 - 800xA Software Applications
 - Network Components

Check if each IT Asset is defined in the PNSM Library.

3. Is Alarming of IT Assets issues needed?
 - Define the number of IT Assets to be monitored.
 - Determine the number of “100 Asset Monitors” licences needed.
 - Define the AO Server that will be running the IT Asset Monitors.
 - Define the update rate of the Asset Monitor per IT Asset.
4. Define where the IT Assets are located:
 - Same domain.
 - Same physical network.
 - Define the network access to the IT Assets.

5. Determine how the different physical networks are connected:
 - Using a router.
 - Using a workstation.

A PNSM Server installation is needed on each workstation that connects two physical networks. Define the workstation nodes where the PNSM Server have to be installed.

Planning Considerations for Field Device Asset Monitoring

1. Define which Field Devices should be monitored:
 - HART Devices on AC 800M. HART Device Integration licenses are required.
 - HART Devices on HART MUX. HART Multiplexer licenses are required.
 - PROFIBUS devices. PROFIBUS Device Integration licenses are required.
 - FOUNDATION Fieldbus devices. FOUNDATION Fieldbus Device Integration licenses are required.
2. Identify the size of the system: small, medium, large.
3. Define the Field Device Network Structure:
 - Identify the number of HART Connectivity Servers and the number of devices in each server.
 - Identify the number of FF Connectivity Servers and the number of devices in each server.
 - Identify the number of PROFIBUS Connectivity Servers and the number of devices in each server
4. Define the Device Types. Check if all Device Types are defined with an asset monitor in the Device Library. If the Device Type does not have an asset monitor defined, a default or generic asset monitor can be used for the Device Type.
5. Identify the total number of devices with Asset Monitors.

- Define the number of Asset Monitor licenses needed. Refer to the *System 800xA System Guide Technical Data and Configuration (3BSE041434*)* for the more information.
 - Estimate the number of asset conditions (number of devices * 12) and define the number of Asset Monitoring Server (less than 30.000 conditions per server).
6. Identify the size of the system.
 - Define the system node types.
 - Define the Asset Monitoring Server.
 7. Define the needed Asset Monitor Data Sources and the asset monitor update rate.
 8. Define the Asset Monitoring Server for each device.

Planning Considerations for Process Equipment Monitoring

1. Define the process equipment needing condition monitored.
2. Define what functionality should be monitored for each process equipment type:
 - Is an Asset Monitor already available? If so, is a specific Asset Monitor license required?
 - Can a basic Asset Monitor be used?
 - Is asset monitor development needed?
 - Define the Asset Monitor types per Process Equipment.
3. Identify the size of the system: small, medium, large.
4. Identify the data sources needed to provide data to the asset monitors.
5. Identify the total number of Process Equipment with Asset Monitors.
 - Define the number of Asset Monitors licenses needed. Refer to the *System 800xA System Guide Technical Data and Configuration (3BSE041434*)* for the more information.

- Estimate the number of asset conditions and define the number of Asset Monitoring Servers (less than 30.000 conditions per server).
6. Identify the size of the system.
 - Define the node types.
 - Define the Asset Monitoring Server.
 7. Identify the needed data sources and define the network connection to the sources.
 8. Define the needed Asset Monitor Data Sources and the asset monitor update rate.
 9. Define the Asset Monitoring Server for each Equipment.
 10. Define the Asset Monitor configuration per object type.

Asset Monitor Configuration

The asset monitors delivered by the Device Integration Center have preconfigured severity settings for the asset monitor condition and subcondition. Each asset monitor alarm condition has a severity defaulted to 1000. Refer to the *System Asset Optimization Operation (3BUA000150*)* for more information about alarm severities.

The severity is a configurable asset monitor parameter, on either an aspect category or individual instances. Asset monitor parameter changes can also be implemented using Bulk Data Manager.

The asset monitor severity is used to notify a maintenance engineer that a device is in an alarm situation in the Alarm and Event List. Asset Optimization alarms can be filtered from process alarms and system alarms. For each asset monitor, the asset monitor subcondition severity level should be evaluated for process needs. More information can be found in the *System 800xA Asset Optimization Configuration (3BUA000118*)*.

CMMS Integration

CMMS Integration brings maintenance management to the operator environment to give a single window interface for multiple systems. Context menus on process graphics, the Alarm and Event list, etc. provide access to several views and actions

for the specific CMMS item. This allows plant personnel to collect, compare, and monitor field device data to accurately assess equipment conditions in real time.

Planning Considerations

1. Review what CMMS is in use?
 - SAP/PM 4.7.
 - MAXIMO 6.2.
2. Define the requirement of the CMMS integration
 - Will the CMMS information be accessible in the operator environment?
 - Will work order information be displayed?
 - Will Equipment status be displayed?
 - Will Maintenance Schedule be used?
 - Should it be possible to submit fault reports from DCS to CMMS?
 - Will CMMS specific work from 800xA workplaces be supported?



If the above functionality is required, add related CMMS aspects to the object types.

3. If Maximo Integration or SAP/PM Integration functionality is needed, the Asset Optimization Maximo Integration software license is required.
4. Define where the CMMS is located.
 - Define the network structure so that the AO Server can connect to the CMMS workstation.
 - Identify the main AO Server.
 - Identify the Network interface.
5. Define the user strategy:
 - Define the 800xA user that shall have access to CMMS information.
 - Identify the CMMS credentials for the 800xA user.
6. Define the user rights:
 - 800xA user that can view CMMS information.
 - 800xA user that can submit fault reports.
7. Determine what equipment in the CMMS shall be linked to 800xA:

Define the tags and related equipment IDs in the CMMS.

Calibration Management

Calibration integration is an engineered solution. The solution is based on standard technologies like DBMS, OPC, XML and Web Services and allows integration between System 800xA and a third party calibration application. The horizontal - across features, and vertical - details of a feature, level of integration is determined by the openness of the calibration application.

Planning Considerations

1. Define what calibration functionality is needed?
 - Zero point adjustment.
 - Initial Device Calibration.
 - Periodic calibration verification.
2. Define the system interface of the devices?
 - mA devices without HART interface.
 - mA devices with HART interface.
 - PROFIBUS devices.
 - FOUNDATION Fieldbus devices.
3. Define the device types (measurement):
 - Temperature (sensor type).
 - Pressure (Abs.-, Diff-, Over-, pressure).
 - Flow.
4. If mA, PROFIBUS, or FOUNDATION Fieldbus devices are used:
 - Verify if Device Calibration System software supports calibration for such device types (measurement).
5. If mA, PROFIBUS, HART, or FOUNDATION Fieldbus devices are supported:
 - Define Device Object types for mA devices in 800xA.

- Verify the needed device types available in the 800xA device library.
 - Verify if the needed device object supports calibration.
6. Define the number of calibrators needed. Base the decision on the state of the environment is hazardous or not:
 - Select Intrinsically Safe or not Intrinsically Safe version of calibrator.
 7. Select the needed number and type of calibration modules. The sensors are intrinsically safe and can be used in non-intrinsically safe environments.
 8. Define the SIL Level for all devices.
 9. Define the Calibration procedures for each device type.
 10. Define the Calibration cycle for each device.
 11. Define the date of the first calibration.
 12. Define the user concept for the calibration work on the choices below:
 - Will Calibration information be exposed to operators?
 - Will a hierarchy between technicians be used?

Usage of Asset Optimization

CMMS System Setup and Configuration

System 800xA supports two CMMS, Maximo, and SAP/PM Plant Maintenance. Only one CMMS system is supported per 800xA system.

In order to make CMMS Connectivity operational, it is necessary to map the 800xA System object to the CMMS Equipment.

Maximo Integration. Maximo Integration includes a Maximo Equipment ID aspect, Maximo Credentials aspect, and CMMS Views aspects. The Fault Report Submitter aspect (part of the Asset Optimization Server system extension) makes it possible to submit fault reports to the Maximo system.

The Maximo Equipment ID provides information for mapping the System 800xA object to the Maximo equipment in the CMMS database for a particular Maximo Server. Multiple Maximo Equipment IDs can be associated with the same asset thus

providing the ability to map to more than one piece of equipment defined in Maximo.

Maximo Credentials contain the Maximo user credentials used to access the Maximo Server by the specific 800xA user.

CMMS Views consist of the following:

- Active Work Orders View.
- Work Order History View.
- Equipment Status View.
- Preventive Maintenance Schedule View.
- Spare Parts/Availability of Spare Parts Views.
- Maximo Portal Views.

For every entry in the CMMS view, it is possible to navigate to the related Maximo portal view. The Maximo portal view is a web view that allows direct interaction with the Maximo data (if the user has the appropriate rights). Changes to the data are possible.

SAP/PM Integration. SAP/PM Integration includes a SAP Equipment ID aspect, SAP Credentials aspect, and CMMS Views aspects. The Fault Report Submitter aspect (part of the Asset Optimization Server system extension) makes it possible to submit fault reports to the SAP system.

The SAP Equipment ID provides information for mapping the System 800xA object to the SAP equipment in the CMMS database for a particular SAP Server. Multiple SAP Equipment IDs can be associated with the same asset thus providing the ability to map to more than one piece of equipment defined in SAP.

SAP Credentials contain the SAP user credentials used to access the SAP Server by the specific 800xA user.

CMMS Views consist of the following:

- Active Work Orders View.
- Work Order History View.
- Equipment Status View.

- Preventive Maintenance Schedule View.
- SAP/PM Portal Views.

For every entry in the CMMS view, it is possible to navigate to the related SAP/PM portal view. The SAP/PM portal view is a web view that allows direct interaction with the SAP/PM data (if the user has the appropriate rights). Changes to the data are possible.

For PROFIBUS or HART devices, the configuration mapping can be performed using BDM with first instantiation of the devices or later on. For FOUNDATION Fieldbus devices, the configuration mapping can be performed using BDM after uploading the FF configuration into the Plant Explorer.

Plan for Network Supervision

To be able to supervise the correct function of the 800xA nodes and the system network there are tools that are always available but there are also some tools that need to be installed and configured to provide good supervision information.

RNRPs low level node and network supervision functions are always available when RNRP is used. Events and alarms are generated based on the information from RNRP. These will be presented in Alarm lists and Event lists. Consider how to use the lists and what functions to supervise where.

The System Status viewer is always available for all node object in the Node Administration Structure. You may consider to add the System Status Viewer aspect also to objects in other structures, for example the functional structure.

The *PC Network and Software Monitoring* (PNSM) package provides monitoring of the IT asset status. Additionally, it provides supervision of single node network utilization, node to node bandwidth utilization, and supervision of network interface. An IT asset object needs to be created and configured in order to monitor the IT asset.

When used in conjunction with Asset Optimization, the Asset Monitoring alarms are generated based on error conditions and hence, PNSM serves as a powerful tool for monitoring the network load and equipment.

Section 11 Commissioning the Solution

System Commissioning Considerations

This section describes the tools that should be used and items that should be checked before commissioning a solution:

- Diagnostics Collection Tool.
- Control checks (I/O and task tuning).
- Logs.
- Field devices.

Diagnostics Collection Tool

The ABB Diagnostics Collection Tool (DCT) is used to collect diagnostics information for analysis from a local or remote node in a distributed LAN system. The data is packaged into compressed cabinet (.cab) files that are attached to the reported problem.

The purpose of the DCT tool is to unify the information gathering process for all ABB IndustrialIT products from installations at customer sites. This speeds up the problem resolution process performed by the ABB support organization. The ABB DCT enables diagnostics information gathering from several IndustrialIT products in a consistent way. The main functions of the tool are:

- Collect diagnostics data from nodes in a distributed system.
- Explore the contents of the collected data.
- Analyze relevant parts of the collected data.

For more information about the plug-ins available with DCT, refer to *System 800xA Tools (2PAA101888*)*.

Control

I/O Checks

As the first step in the commissioning phase it is recommended to perform I/O checks. This can be done before the applications are fully configured. For this purpose, a setting, in Control Builder M, called Copy Unconnected Channels can be used (Figure 21).

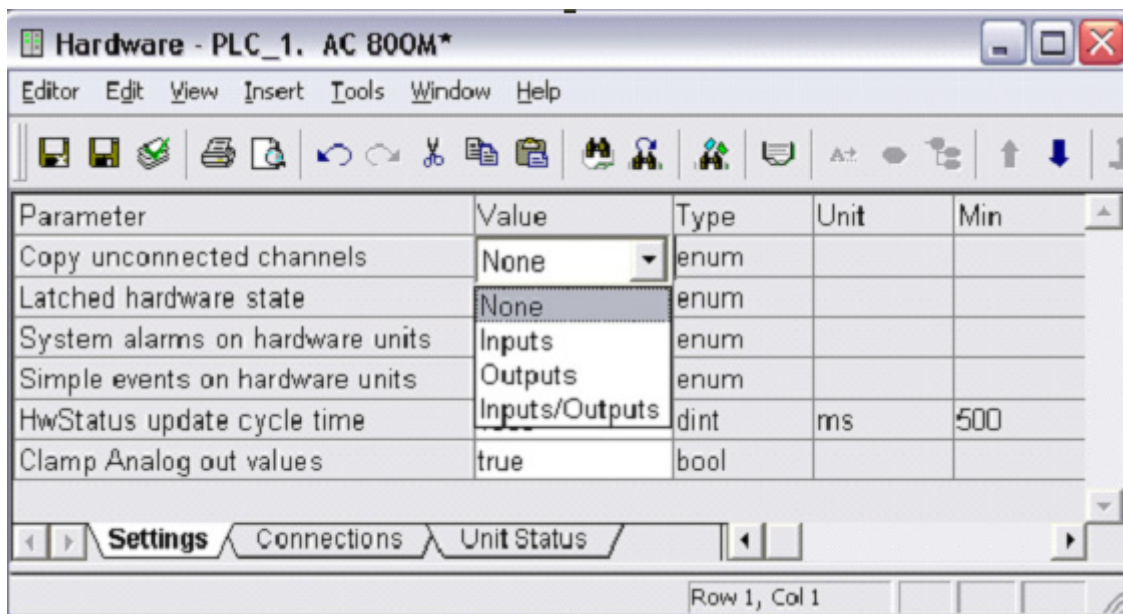


Figure 21. Usage of the Setting Copy Unconnected Channels.

If the value is set to any value other than None, I/O copying is started even if there are no I/O connections to the applications in the controller. In this state, input values are automatically shown in online hardware editors. Outputs can be set by forcing the channels in the online hardware editors.

After the check it is very important to set the value back to None to avoid a high CPU load in the controller.

Task Tuning

All tasks in the controller needs to be tuned in respect to priority and offset, refer to the *System 800xA Control AC 800M Planning (3BSE043732*)*. The tuning should be performed after the I/O check but before the actual application commissions. It can not be done in an office environment since the execution time for I/O copy will then not be present.

An example of a typical situation and what actions to be taken in order to tune task execution is found in the *System 800xA Control AC 800M Planning (3BSE043732*)*.

Logs

It is recommended to check for errors in all system logs before a site is left. Some of the products have manuals describing logs and log handling. Refer to the following manuals concerning Control and Information Management:

Table 22. References to Information on Logs and Log Handling

Area	Reference
Control	<i>System 800xA Control, AC 800M Configuration (3BSE035980*)</i> .
Information Management	<i>System 800xA Configuration (3BDS011222*) and System 800xA Information Management, Configuration (3BUF001092*)</i> .

Field Devices

After physically connecting the field devices to the fieldbus segments a couple of actions have to be made to get in touch online with the devices.

Field Device Assignment

HART devices are working immediately after powering up at the connected I/O channel. To communicate acyclic via DTM the channel was already configured within the Fieldbus Management aspect in the configuration phase.

PROFIBUS devices have to get the right addresses which is previously configured in the configuration phase. Per default PROFIBUS devices are delivered with the address 126. The right address can be assigned via the web interface opened with the IP address of the corresponding controller. Please have a look at the *System 800xA Device Management PROFIBUS and Hart Configuration (3BDD011934*)* for further information. This cannot be done with Bulk Data Management.

FOUNDATION Fieldbus devices will be assigned to devices listed in the live list via the Fieldbus Builder FF. After assignment the devices have the right address and physical device tag. Please have a look at the *System 800xA Device Management FOUNDATION Fieldbus Configuration (3BDD012902*)* for further information. FBB FF does not support Bulk Data Management.

Upload device parameter

In most cases devices will be delivered with a preconfigured set of device parameter. In this cases it is recommended to upload the parameter before continuing with commissioning work.

For PROFIBUS and HART devices the upload is performed via the Fieldbus Management aspect. It is possible to upload one or more devices selected at a specific fieldbus segment.

For Foundation Fieldbus devices the upload is performed by means of the Fieldbus Builder FF.

Check Device Parameter

An upload of device parameters for PROFIBUS and HART devices will overwrite the previously offline configured parameter. FBB FF offers a list where the configured parameter and the uploaded parameter will be listed in parallel columns. The user can decide which data should be taken.

Load OPC Server and Asset Monitor Server Configuration

After finishing the checks described above, configure the OPC Servers PROFIBUS/HART and load the Asset Monitor Servers. Please check the Control Connection aspect in case of HART and FOUNDATION Fieldbus devices if there is no empty Property View tab. For PROFIBUS devices check the Property View tab in

the Hardware Unit aspect. If it is empty, operation application like e.g. faceplates or graphic displays cannot display values and asset monitors shows bad quality for the needed OPC items.

Loop Check

If the device parameter have been adapted to the process needs they can be downloaded back into the field device. For PROFIBUS and HART it is performed via Fieldbus Management aspect. The download for FOUNDATION Fieldbus devices is performed via FBB FF.



Please consider the commissioning procedures required by the specific field devices

The loop checks can be started with simulating e.g. transmitter outputs or actuator setpoints. There are different ways to make simulation. The most convenient way is to do this via DTMs and via FBB FF for FF devices. The simulated value can be checked if it is right displayed in Control Builder M and/or in 800xA, if the signal way and the ranges are OK. Configured limits can be checked as well as if asset monitors give the right feedback at e.g. sensor failures etc.

Section 12 System Configuration Documentation

This section describes considerations on 800xA system functionality that helps to produce the system configuration documentation of an engineering project. This documentation describes how the 800xA system has been configured to fulfill the functional requirements of the customer. It shall enable the plant staff to run and maintain the software configuration of the 800xA system.

System configuration documentation is just a part of the complete project documentation to be delivered to the customer. Complete project documentation also includes documents not directly related to control system like Terminal Connection Tables, Cable Route Network, Loop Diagrams, Cable Tables and the 800xA product documentation. What actual documents have to be produced is to a large extent dependent on the industry and the product/service being provided and typically is determined by contract with the customer.

Depending on the specific aspect systems different functionality is available that helps you to produce the system configuration documentation, refer to [Table 23](#).

Table 23. System Configuration Documentation Support

No	Required in Project	Aspect System	Supports
1	Function Diagrams as documents and control application configuration means.	Function Designer	System wide Function Diagram design

Table 23. System Configuration Documentation Support (Continued)

No	Required in Project	Aspect System	Supports
2	Templates describing document frame and Document Manager based meta data. Can be adapted project specific.	Function Designer (continued)	Function Diagram Templates
3	Printing out Function Diagrams of a subtree or of the whole Functional Structure. Also on Function Designer made Object Types in libraries.		Printing / bulk printing of Contents Diagrams (table of contents) and Function Diagrams
4	Document data base to Function Diagrams and can be used to add a description file (Word, Excel, ...)		Function Diagram Document aspect, provides default Document Manager meta data
5	Topology Diagrams as documents (and control hardware configuration means)	Topology Designer	System wide Topology Diagram design
6	Templates / skeletons describing document frame and Document Manager based meta data. Can be adapted project specific.		Topology Diagram Templates and Skeletons
7	Printing out Topology Diagrams of a subtree or of the whole Control Structure		Printing / bulk printing of Contents Diagrams (table of contents) and Topology Diagrams

Table 23. System Configuration Documentation Support (Continued)

No	Required in Project	Aspect System	Supports
8	Document data base for Topology Diagrams and can be used to add a description file (Word, Excel, ...)	Topology Designer (continued)	Topology Diagram Document aspect, provides default Document Manager meta data
9	Generation and printing of control project configuration documentation as Microsoft Word document. Library, application and hardware unit documentation.	Control Builder M Professional	Control Project Documentation (templates available and adaptable)
10	Generation and printing of FF configuration documents	Fieldbus Builder FF	Documentation job definition and generation in Fieldbus Builder FF (templates definable)
11	Integration of dynamic documents of Microsoft Word, Microsoft Excel and AutoDesk AutoCAD drawings as well as arbitrary non-dynamic documents of Mircosoft Office and other tools	Document Manager	Document aspect object integration.
12	Printing multiple Document Manager documents	Bulk Data Manager /Engineering Templates	Bulk Data Manager Document Batch Print template
13	Retrieve configuration data of other aspect systems for documentation purposes. Specific worksheets / data sheets configurable.	Bulk Data Manager	Configuration data retrieval and document design with Bulk Data Manager based on Microsoft Excel.

Table 23. System Configuration Documentation Support (Continued)

No	Required in Project	Aspect System	Supports
14	Data retrieval and printing with Property Editor or Bulk Data Manager user interfaces. For example: Engineering signals (CBM_Signals).	Parameter Manager	Storage of extended configuration data in Parameter aspects. Configurable aspect categories.
15	Integrating non-dynamic documents of Microsoft Office tools and others.	File Viewer	Document aspect object integration.
16	Printing out Graphic Displays (graphic and / or VB code)	Graphics Builder	VB printing functions
17	Printing out Graphic Displays / Trends	Process Graphics	Print functionality
18	Printing out Graphic Displays / Trends	Information Management / Display Services	Print functionality
19	Printing out process values	Information Management / Data Direct	Print functionality

Generally you can use Microsoft Office tools to write and produce the documents not generated by 800xA tools. Microsoft Windows Print (screenshot) functionality can be used to include bitmap pictures into documents written/produced with Microsoft Office tools. Documents can be converted with PDFMaker to Adobe PDF standard format from within the Microsoft Office Tools.

Produced document files you can integrate into aspect objects, for example placed or created in Documentation Structure, using Document Manager (in original format or PDF format) or File Viewer (in PDF format). This makes documentation accessible on your workplaces.

Document Manager and File Viewer aspects can get electronic signatures that enable you to verify if the documents they hold have been changed.

Planning Considerations

1. Check for requirements on system configuration documentation. Most items / questions listed below are also valid for complete project documentation.
 - a. Which toolset to use?
 - b. What documents to produce?
 - c. Contents / layout standards to comply with?
 - d. Guidelines regarding kind and depth of content?
 - e. Which templates to use / to configure?
 - f. Printed and / or electronic format?
 - g. What is the needed document approval cycle?
 - h. Shall documents be electronic documents?
 - i. How electronic documents shall be stored and accessed?
 - j. Shall electronic documents be secured by electronic signatures?
2. Map the requirements to the provided documentation functionality (see list above) of 800xA to determine how system configuration documentation tasks have to be solved in total.
 - a. Tasks supported by 800xA tools.
 - b. Tasks to be solved with non-800xA tools.
 - c. Define the whole document production process.
3. In detail plan for the documentation tasks
 - a. Infrastructure.
 - b. Resources.
 - c. Schedule.

Section 13 Basic Concepts

Data Access

Data Access Based on OPC

Overview

The 800xA System OPC Data Access functionality consists of two main functions, OPC Connect and OPC Support.

OPC Support is the functionality provided when the 800xA system acts as an OPC Data Access Server. Clients can subscribe to data from the OPC Data Access Server.

OPC Connect provides connectivity to OPC Data Access Servers. The OPC Data Access Server of System 800xA is a generic switch which can access other servers referred to as Data Sources.

The Data Sources can consist of:

- OPC Data Access servers.
- Specialized adapters via the DA Connector component.
- Aspects that provide OPC properties.
- 800xA service applications that have been extended with the DA Injector in order to provide OPC access to the service application data.

An overview of the Data Access concept is shown in [Figure 22](#).

The boxes filled with grey denotes Data Access related Components or Aspects.

The OPC items represents connections to the Data Sources of the OPC server. In the OPC Data Access Server of System 800xA each item refers to a Data Access

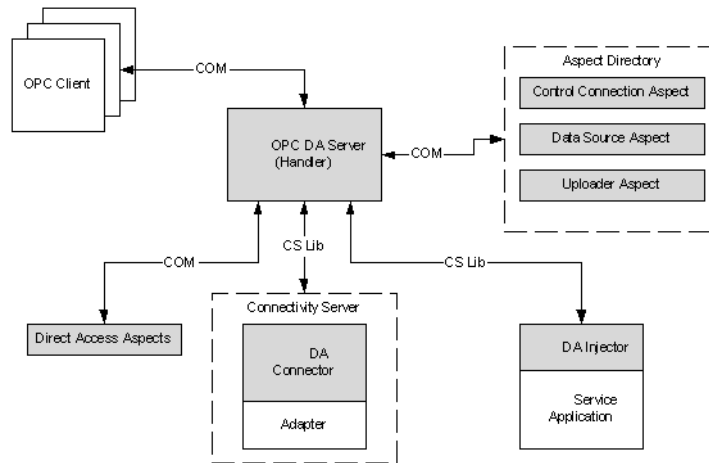


Figure 22. Block Diagram of the Data Access Concept

property that belongs to an object in the Aspect Directory. The properties of the object are defined by one or more Property Definition Aspects.

In System 800xA OPC Data Access there are different types of Property Definition Aspects available.

- **Direct Access Aspects:**
Defines Data Access properties implemented by the aspect system. The System 800xA OPC Data Access Server will access these properties by directly invoking these types of aspects.
- **Control Connection Aspects (CCA)**
Defines Data Access properties that exist in an external device and are accessed via a System 800xA DA Connector/Adapter.
- **Global Properties Aspects (GPA)**
Defines Data Access properties that are common for all objects in the 800xA system. A system service application extended with the DA Injector handles these properties in System 800xA.

OPC Connect

OPC Connect provides connectivity to third party OPC Data Access Servers. The connection can be made in different ways. The recommended solution is to run the third party OPC Data Access Server in a Connectivity Server node as shown in [Figure 23](#). In this manner, all the benefits of 800xA network failure detection are utilized.



Several OPC Servers may run in the same Connectivity Server but, depending on load and vendor specification, they may have to run in separate servers.

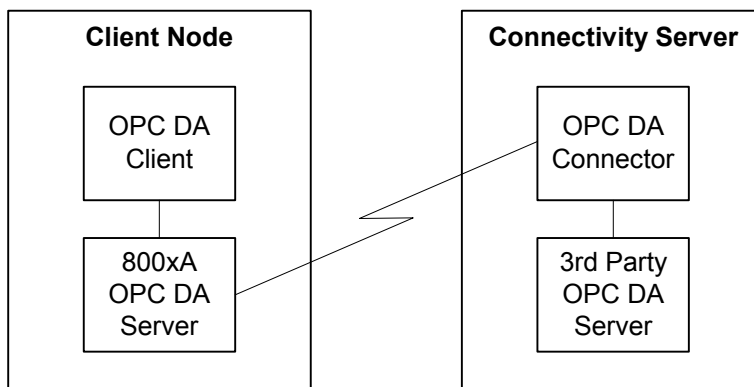


Figure 23. OPC Server in Connectivity Server

If the third party OPC Data Access Server runs in a non-system node, the connection is made via DCOM. This is shown in [Figure 24](#).

DA Connector/Adapter

The System 800xA DA Connector process is a System service of System 800xA controlled by the System service manager of System 800xA. It can only run in a server node in the system. The server node is called a Connectivity Server. In the Connectivity Server an Adapter is specifically designed to access data from a particular type of Data Source. The Adapter is a component that is hosted by the DA Connector process.

The Connectivity Server communicates with the clients via the CSLib/System framework. CSLib is a standard communication protocol in the 800xA system. The

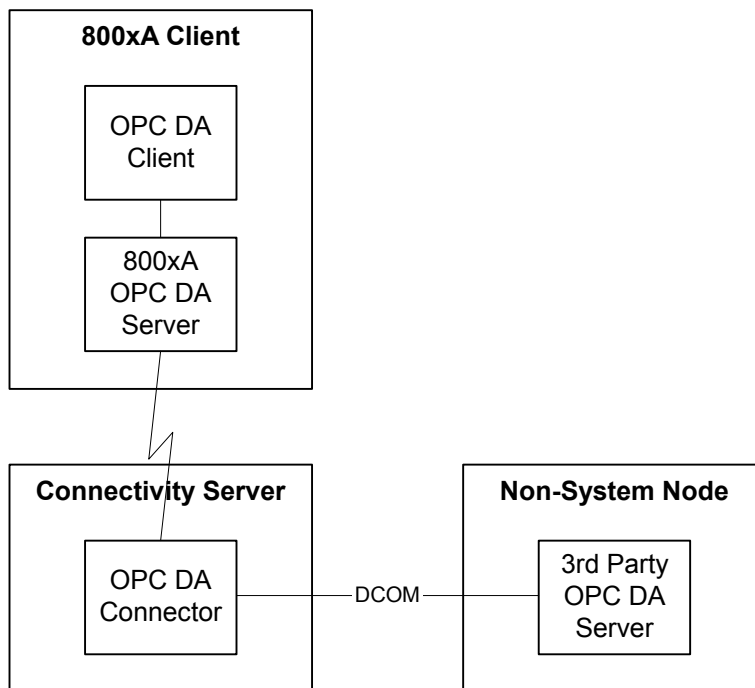


Figure 24. OPC Server in Non-System Node

CSLib component allows redundant, asynchronous or synchronous client/server communication.

The System 800xA OPC Server Toolkit provides a special System 800xA OPC Server Adapter to access third party OPC servers. If the third party OPC server is running in the same node as the DA Connector, COM is used for communication. Otherwise DCOM is used.

DA Injector

System 800xA DA Injector is used as a component in a System Service application of System 800xA to expose data to the System 800xA OPC Data Access Server via CSLib. The System Service application hosts the DA Injector component. The OPC Data Access Server will handle the service application as if it was a DA Connector.

Redundancy

Data from a redundant Service Provider is processed by the System 800xA OPC Data Access Server according to provider, quality, timestamp and value. Redundant Service Providers form a logical unit called a Service Group.

When the 800xA system is redundant, the OPC Data Access Server can realize two types of redundancy:

- Failover.
- Parallel.

Failover Redundancy

Failover redundancy is the default configuration of redundancy. A subscription is reestablished to a redundant service provider at failure. Load balancing in the Service Group decides which Service Provider the client will be connected to.

If a Data Source fails or the contact is lost due to network failure a new connection is established to another service provider in the same service group (failover). The new connection is kept until the original Data Source has been restarted or the network failure has been handled. When the original Data Source is operational again a reconnect is made to the first Service Provider.

[Figure 25](#) illustrates a redundant connection between client A, B and process control objects AI117, AI118, and AI119 respectively.

Process control objects AI118 and AI119 have a Data Source that is referring to Service Group 2 containing two service providers. This means that redundancy can be provided by access through either Server B or Server C. Process control object AI117 is represented by Service Group 1 which contains only one Service Provider. Hence redundancy is not provided in this case.

Parallel Redundancy

When parallel redundancy is applied, subscription is always established to two redundant service providers, OPC servers. Service affinity is not applicable when parallel redundancy is used.

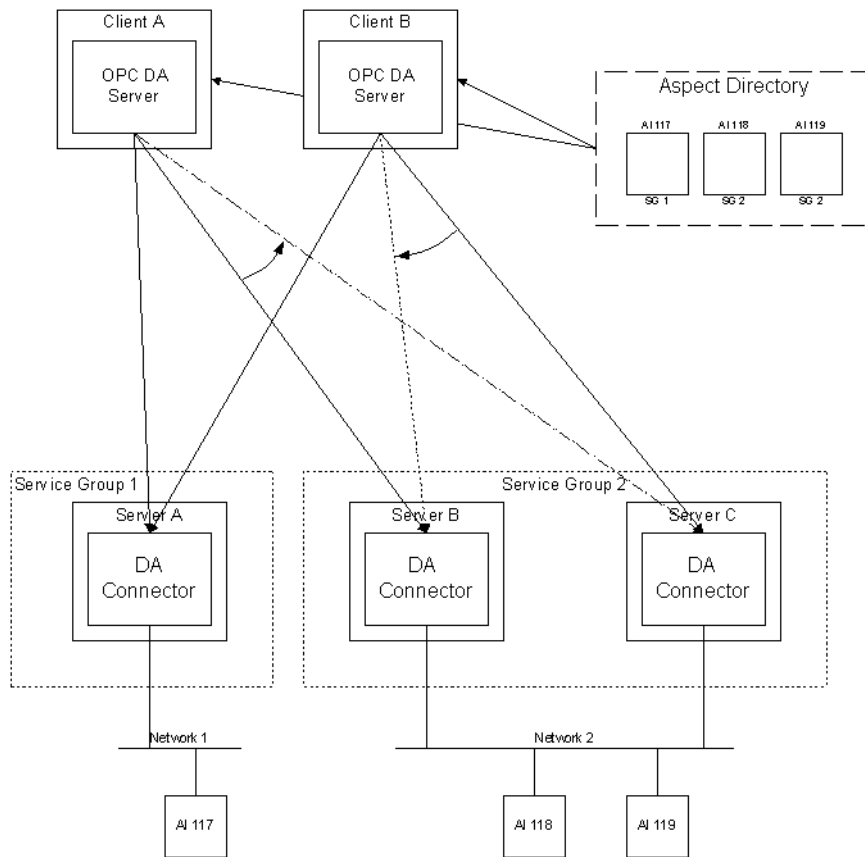


Figure 25. Redundancy Concept

Parallel redundancy has the benefit of no loss of data if one service provider fails. Selection of good quality data is performed when data quality differs between the Service Providers.

It is possible to configure each OPC DA Connector Service Group if parallel redundancy is allowed. The default is to allow parallel redundancy.



Parallel redundancy might increase Controller Network load. Check connectivity documentation for restrictions regarding parallel redundancy.

Upload

The uploader function browses an OPC DA Server for all branches and leafs, and creates a reflection of them in the Control Structure. Branches are appended as Aspect Objects in the Control Structure. Leafs are declared in Control Connection aspects as aspect object properties.

Authenticated Write

Authentication of write operations to certain object properties can be performed before the actual access to the property. This is to protect the object properties from unauthorized writing or writing by mistake. Write operations to properties configured for authentication are only allowed if:

- The write operation is asynchronous.
- The OPC Client is running in an Operator Workplace application process.

Each property is, in the Property Info tab of the Control Connection aspect, configurable if it should be:

- Not authenticated.
- Re-authenticated.
- Double authenticated.

Every property in the write operation must be granted access for the authentication to succeed. If the authentication is successful, the operation is passed on to the corresponding Data Source, otherwise it will be rejected.

Confirmed Write

Confirmed write may not be supported by all 3rd party OPC Servers.

To ensure that an operator writes correct values to correct properties in a controller a confirmed write function is available. This is particularly useful in safety critical applications where online operations are performed. The Confirmed Write function supports user confirmation and data protection of OPC write operations.

Write operations to properties configured for confirmation are only allowed if:

- The write operation is asynchronous.
- The OPC Client is running in an Operator Workplace application process.

A write performed by a System Service of System 800xA will not trigger the Confirmed Write function. The accessed device may in this case stop the write access. Write operations from third party OPC clients will be terminated unconditionally.

To reduce the risk for unintended write operations the following functions are added:

- A confirmation dialog box shows the property and the write value before the actual write operation is performed. The user is prompted to proceed or reject the operation.
- A data transfer check is realized by means of Cyclic Redundancy Check (CRC) or other methods to ensure that data value is written correctly. This is to protect against transmission or program failures. The feature is optional and is activated if any Aspect on the object that holds the CCA supports it.

Confirmed write is configured for each property on the Property Info tab of the Control Connection aspect.

For configuration information on data access based on OPC, refer to the *System 800xA Configuration (3BDS011222*)*.

Configuration Change and Access Management

Introduction

Configuration change and access management of System 800xA allows you to assure that the manufacturing process the system is controlling consistently produces results according to the specifications. It helps you to build, run and maintain applications that comply with FDA cGMP and 21 CFR Part 11. Refer to [Section 12, System Configuration Documentation](#) for compliancy planning.

The 800xA system provides means to manage versions of the configuration data of a whole system, so called system versions, via a backup / restore mechanism. It further supports user controlled synchronization of a Production System with configuration data changes done on an Engineering System using Import / Export.

Regarding authorization and access management the 800xA system allows managing of the user's access to entities down to the single property of an aspect object.

Changes on aspects of a certain category can be secured by requiring re-authentication or double authentication. All changes done in an Production System or an Engineering System can be audit trailed. Audit Trail messages can be archived.

Regarding health of process data the 800xA system for example provides signal quality support and regarding health of the hardware devices system status support. SFCs in Control Builder M Professional or SPL diagrams in Function Designer and corresponding SFC displays on the Operator Workplace can be the base for checking the permitted sequencing of steps and events.

Security

In general data security of a plant installation requires taking care of risk management, physical security, computer security and personnel security.

Base for personnel Authorization and Access Control is the Security functionality provided by the 800xA system.

The System Security model of System 800xA is based on extensions to Windows security model. The extensions make it possible to set permissions for users or user groups on an 800xA system, a structure or part of a structure, or an aspect object. The security setting can also be limited to a single node, or all nodes. This feature makes it possible to require an operation to be performed from a node with a visible view of the area it controls.

Connected to the security is the usage of roles. The roles adapt the user interface for different user groups. Some operations require an application engineer or system engineer role to be performed.

However, having the correct role does not give the user the permission to perform the operation. The permission is completely controlled by the security configuration of the system.

The security and audit in an 800xA system are set with the Security Definition aspects, which are added to Aspects Objects. You set the security by changing the Authority Range, Permissions (related to User Identity/Group and Node) and

Search Option of a Security Definition aspect. By these settings you can define a wanted security level for your plant. You set the Audit function more or less in the same way as the security. In the audit log you can follow what happens in your plant from a security point of view - for example attempts to access secured objects etc.

Security for an Engineering environment is controlled with Engineering Security Definition aspects.

The general principle when configuring System Security of 800xA systems is to keep it as simple as possible. A complicated security configuration is, in it self, a security problem, since the overview is lost.

For a small installation it may be enough to change the Default Security Definition (an aspect in the 800xA system). A larger installation may have several operator groups with allowed permissions on different areas in Functional Structure. Only exceptionally should Security Definition be put on individual Aspect Objects. It is also recommended to work with groups instead of individual users.

Detail information on security configuration of the 800xA system are found in the *System 800xA Administration and Security (3BSE037410*)*.

Authorization

Security controls a user's authority to perform different operations on Aspect Objects, depending on several parameters:

- The user's credentials, as provided by Windows.
- The node where the user is logged in. This makes it possible to give a user different authority depending on where he/she is located, e.g. close to the process equipment, or in a control room.
- The operation the user wants to perform.

Access Control

Basic access to the system is controlled based on user's credentials entered during Windows Logon or logover.

Advanced Access Control provides re-authentication and double authentication.

Authentication

For process critical operations, an aspect category or OPC property may be configured to require an explicit authentication operation for the operation to be performed. Two different authentication operations are supported:

- Re-authentication is used to guarantee that an operation is performed by the correct person. To require a re-authentication immediately before a change can be performed guarantees that nobody has the possibility to use a workplace when an operator temporarily leaves it.
- Double-authentication is used for operations critical to the quality of the product or required by regulation. It is used where the knowledge of an operator is too limited or where there must be another operator that verifies the change before it is done.

Re-Authentication

The re-authentication dialog box is used to guarantee that the correct person performs an operation. When a change is made to an OPC property or aspect that requires a re-authentication, the dialog box pops up. You must type your User ID and your password and select a reason from the **Reason** drop-down menu before you can press **OK**. It is optional to type a comment in the **Approval Comment** text field.

Double-Authentication

The double-authentication dialog box is used to guarantee that the correct persons perform an operation. When a change is made to an OPC property that requires double-authentication, the dialog box pops up. You must type your User ID, password and select a reason from the **Reason** drop-down menu in the **First Approval** area before pressing **Apply**. It is optional to type a comment in the **Approval Comment** text field.

If the information provided by you as the primary authenticator is accepted the **Second Approval** area is activated.

In the **Second Approval** area a secondary authenticator, which must have Approve permission granted on the object, shall type his/her User ID, password and reason

before pressing **OK**. It is also here optional to type a comment in the **Approval Comment** text field.

Log Over

The log over function enables a fast switch between users in a running workplace. For example if an operation requires a permission not held by an operator, another user (e.g. a system engineer) that holds the required permission, can log over to perform that operation. The log over changes the permissions and user roles but keeps all open windows with their present contents. The permitted actions in the open windows are controlled by the permissions of the logged over user.



The log over only affects the System permission. Windows security is still the same as the user logged in. This means that the access to files is still controlled by the user logged in.

To change a user:

1. Right click on the user name in bottom line of your workplace window and select **Change User**.
2. Enter User ID and password of the new user and click **OK**.

Digital Signature

The digital signature function allows you to electronically sign an aspect (typically representing a document) according to the requirements from different national authorities. The Digital Signature is also used for verification of who signed the aspect, when it was signed and if the aspect is unchanged since it was signed. To be able to sign an aspect you need to have the permission configured for the First and/or Second Signature operations.

Configuration Change Management

The 800xA system supports you in maintaining System Configuration Versions of a Production System using the system's Backup/Restore function and in transferring engineering changes between systems and environments.

Transfer of engineering changes from an Engineering System to the Production System is by the Import/Export tool.

Transfer of engineering changes from Engineering environment to Production environment is supported by the Deploy tool. Transfer of engineering changes from Production environment to Engineering environment is supported by the Update tool and/or the Replace tool.

Audit Trail allows to log the engineering / configuration changes.

System Configuration Version

A version of the system is taken by producing a complete (full) backup of the system configuration data and by assigning it a version number and a label that identifies it as a System Configuration Version. Default for a full backup is to include all aspects, objects and all services that support Backup/Restore.

Restore of a System Configuration Version can be executed from Configuration Wizard's Restore System item.

For more details System Configuration Version and Backup / Restore, refer to the *System 800xA Administration and Security (3BSE037410*)*.

Synchronizing Changes Between Systems

Import/Export provides the ability to synchronize two separate systems with each other. This is done based on entities and eventually additional single aspects with or without manual added dependencies. Before transferring the changes the differences between the target system and the archive to import can be detected. After transfer the consistency can be checked.

Synchronizing Changes Between Environments

Deploy provides the ability to synchronize the Production environment with the Engineering environment.

Update and Replace provide the ability to synchronize the Engineering environment with the Production environment.

Synchronization can be done based on entities and eventually additional single aspects with or without manual added dependencies.

Before transferring the changes the differences between the environments can be detected. After transfer the consistency of the target environment is checked by default.

Initial synchronization of the Engineering environment with the complete Production environment can be done using the Refresh engineering environment function of the Configuration Wizard.

Using the Load-Evaluate-Go procedure against the controller allows to evaluate the changes online before making them active. This is done based on entities.

Archive

The archive function of Information Management History Services supports permanent offline storage for historical data collected in property, message, and report logs, as well as the Operator Workplace alarm/event message buffer, including audit trail messages. When a history log becomes full, the oldest entries are replaced by new entries, or deleted. The archiving mechanism copies the contents of selected logs to a designated archive media.

For further information, refer to the *System 800xA Information Management Configuration (3BUF001092*)*.

System and Device Checks

The 800xA system provides comprehensive functions that allow determining the health of the process and of the controlling control system and of the allowed sequences of steps and events.

Examples are: Regarding health of process data the 800xA system for example provides signal quality support and regarding health of the hardware devices system status support. System status can also be displayed in the context of topology diagrams using Topology Status Viewer. Asset Monitoring provides a way to access the status information for IT assets / IT devices used in the context of the control system. SFCs in Control Builder M Professional and corresponding SFC displays on the Operator Workplace can be the base for checking the permitted sequencing of steps and events.

Multisystem Integration

800xA Multisystem Integration makes it possible to supervise and operate several 800xA systems from one central control room. The 800xA systems can be in the same Windows domain, but do not have to be. The supervised system can be without any local workplaces, or be a complete system with its own local control room and is called the Provider because it provides the supervising system with data. The supervising system is called the Subscriber because it subscribes to the values from the provider. Two services implement the communication, the Remote Access Server running in the provider system, and the Remote Access Client running in the subscriber system.

The network between the subscribers and the providers can be anything from a high speed LAN 100 MBit/s down to modem connection with a speed of 512 kBit/s. A password and encryption can be used to secure the connection between the provider and the subscriber.

There are no special hardware requirements 800xA Multisystem Integration other than that required for the 800xA core system software.

The time difference between a Remote Access Client and Remote Access Server should be reasonable (normally less than a minute). Additional external equipment may be needed to achieve this time synchronization.

Section 14 Engineering Workflow

Introduction

This section introduces you to [Engineering Tasks](#) and an [Overall Project Workflow](#).



Note that both engineering tasks and project workflow are introduced on a certain abstraction level. You probably have to adapt them to the concrete requirements of your engineering project.

Detailed examples of configuration workflow steps are described in the *System 800xA Configuration (3BDS011222*)*.

Engineering Tasks

The engineering and configuration of an Industrial IT System 800xA consists of a number of engineering tasks and incremental steps within those tasks. Many of these engineering tasks can be performed in parallel and others can be executed in a different order.

When your experience grows you will also realize where you can execute them in parallel or in another order and where not. Which aspects to configure depends on the aspect systems and the combination of aspect systems in your concrete instance of a 800xA system.

A sketch of Functional Structure built early, for example derived from P&IDs, helps to identify typical solutions to implemented as Object Types/Composite Object Types.

After you have verified the existing Object Types/Composite Object Types in available standard libraries and object type groups and you have designed and implemented the remaining non-standard object types you can start to build final Functional Structure and the further structures.

Every entity that exists more than once could be made as a Composite Object Type. A (Composite) Object Type might not just be a motor or valve. It can be tanks, process units or even whole production lines.

There are four major reasons for using a type object based application design:

- Better application quality in the first place, since less coding is needed.
- Less validation work in validated installations, since less coding is needed.
- Less application maintenance costs, since less code exist.
- Less memory need in the controller, since many instances refer to the same code (type).

Analysis and Design

Before you start the engineering work, you need to make an analysis of the system you are going to build. This analysis can be seen split into six parts:

- Domain analysis.
- Identifying Aspect Objects.
- Identifying Structures.
- Identifying Object Types / Composite Object Types (Typical Solutions).
- Identifying Security Settings.
- Identifying users and groups.

If the analysis and design to come to a Plant model is done during the Plant Design phase or in the early steps of the Configuration (Software Production and Test) phase depends on the concrete project.

Domain Analysis

The intention of the domain analysis is to identify entity objects and structures in the domain. This analysis should be based on existing standards for how systems are composed into different structures, for example IEC61346. Refer to International Electrotechnical Commission's home page (www.iec.ch) for details.

Normally, you start with the top object in the functional structure and start to decompose it according to functionality, thus creating a functional structure of

objects. A sketch of functional structure built early, for example derived from P&IDs, helps to identify typical solutions to be implemented as Object Types / Composite Object Types.

Identifying Aspect Objects

The found domain entity objects are mapped into aspect objects. The aspect objects are used as container models for the domain entity objects. Their aspects will correspond to the functionality of each respective entity.

Identifying Structures

The aspect objects are put into structures. Try to identify the plant structures of the real objects in order to make models of them in the Plant Explorer tool.

Identifying Object Types/Composite Object Types

If applicable, the found domain entity objects with a functionality that reoccurs in several parts of the plant are mapped into existing (Composite) Object Types available in standard libraries and object type groups. Composite Objects Types typically are used as big building blocks (typical solutions). You should also identify if new object types need to be developed.

Identifying Security Settings

When you plan for the data security of a plant installation, you have to take the total security into consideration. It is not enough to make a strong fire wall if people have free access to the operator's room in the plant.

Establish a detailed risk management program which covers the complete spectra of risks. Make a catastrophe plan. Make plans for how to recover and how to restore.

You also need to have procedures for testing, that your security plans function in a number of situations. You have to inform and educate your employees about the security plans for the plant. Update the plan requirements once per year. For further information read the *System 800xA Administration and Security (3BSE037410*)*.

Building of Objects

After you have verified existing standard object types and developed the remaining non-standard object types you can continue to build final Functional Structure and further applicable structures.

- Each (Composite) Object Type that can be used repeatedly will save engineering work.
- You get a methodology with which you can break the process down into smaller parts that are easier to grasp and handle.

Many of such object types are provided initially when you install the libraries provided with the 800xA System product.

In principle you have to create / build

- the Plant Structures (Functional, Control, Location, Documentation),



Which structures to use in an engineering project depends on the concrete requirements. The recommended minimal set is to use Functional Structure and Control Structure. Functional Structure then contains objects with Graphic Displays and, if the Functional Planning approach is used to configure Control Applications, Function Diagrams. For certain projects it might be sufficient to work in Control Structure only.

- the Overview Drawings / Function Diagrams / Graphical Displays,
- the Control Applications,
- and the remaining applications (covering Alarm and Event-, Trend-, History-, Asset Optimization-, Information Management-, and Batch Management-functionality).

Overall Project Workflow

In the following sections an overall project workflow for configuration of the 800xA system shown in [Figure 26](#) and additionally explained in [Table 24](#) is referenced as a guideline.

This workflow is typically executed on an Engineering System (either on-site or off-site). When executed on a Site Engineering System it is included into an extended workflow for Site Engineering, described in [Figure 30](#) and [Table 25](#).



Please be aware that this workflow is an example reference workflow you can, - and probably have to -, adapt to the needs of your concrete engineering project.



The generic term Process Function used in [Figure 26](#), [Table 24](#) and in the following subsections represents a building block implemented by an object type or composite object type. Granularity can vary, for example a whole process section, a tank, a drain system or a single PID loop.



Both [Figure 26](#), [Table 24](#) and the following subsections do not mention iterations in the engineering workflow. Such iterations of course exist and have to be passed if a step leads to non satisfactory results and the causes have to be corrected in the step(s) before.

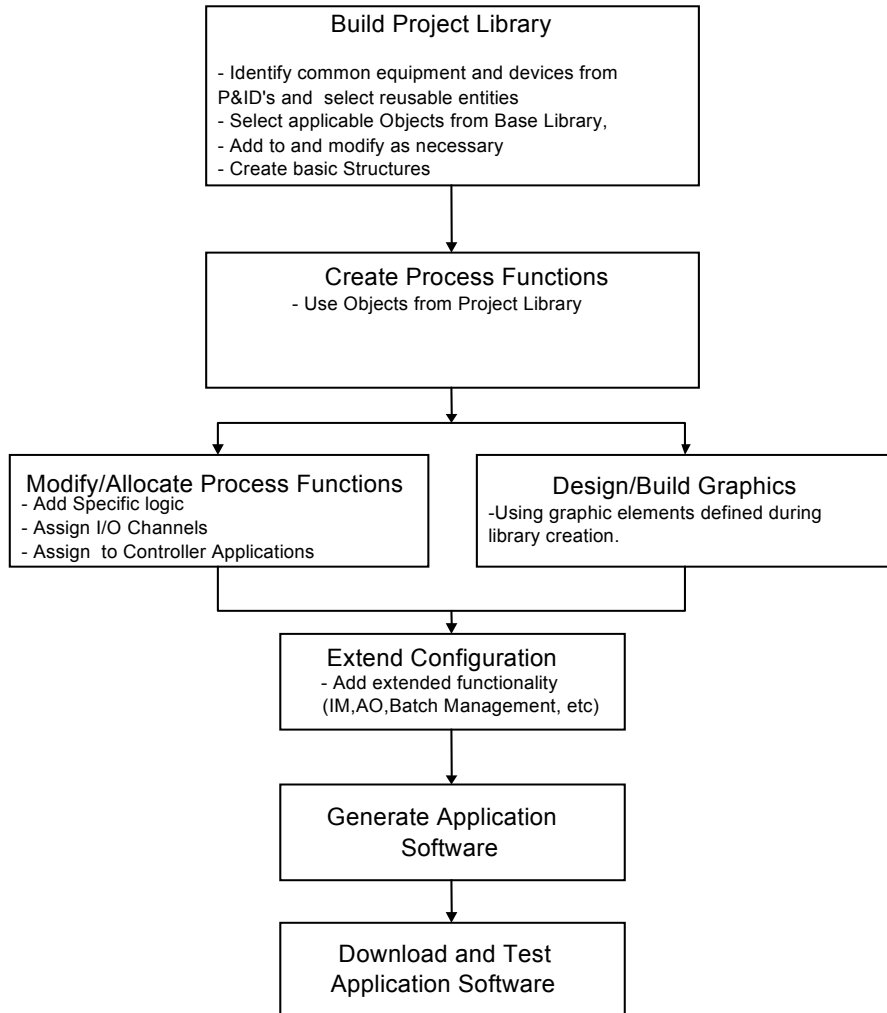


Figure 26. Overall Project Workflow

Table 24. Workflow Activities

Workflow Step	Activities
Create Project Environment (Remark: Not mentioned in workflow chart)	Create System Select and Load Standard Object Type Libraries Load / Import other reusable entities and standards Customize Project Environment
Build Project Library	Identify common equipment and devices from P&IDs and select reusable entities Select applicable Object Types from Standard Libraries Complete Project Library: - Build New Object Type Libraries - Adapt Standard Object Type Libraries - Write Specifications / Guidelines
	Create Basic Structures - Create Upper Level Plant Structures (Functional, Control, Location, Documentation) from Object Types
Create Process Functions	Build Lower Level Functional Structure from Standard Object Types
	Build Lower Level Functional Structure from Adapted Object Types
Modify / Allocate Process Functions	Add Specific Logic Assign I/O Signals Allocate to Applications in Control Structure Allocate I/O Signals to Boards in Control Structure
Design / Build Graphic Displays	Finalize Graphic Displays from Object Types Design and Build New Upper Level Graphic Displays Using graphic elements defined during library creation

Table 24. Workflow Activities (Continued)

Workflow Step	Activities
Extend Configuration	Add Production Management Functionality (Batch Management) Add Information Management Functionality Add Asset Optimization Functionality Finalize Lower Level Location Structure Finalize Documentation Structure
Generate Application Software	Generate Controller Code
Download and Test Application Software	Deploy Graphic Displays Download Controller Code, if required use Load-Evaluate-Go procedure Load other Run-time Configuration data Test complete Configuration

Build Project Environment

Create System

At first you have to create the standard environments on the aspect server and, if available, on the client(s). This requires, that all programs and libraries, necessary for the execution of a project, are already installed on the server and on the clients.



From System Version 6.0 onwards, a simplified procedure of creating a system using the System Configuration Console is available. Refer to *System 800xA Tools (2PAA101888*)* for more information about the System Configuration Console tasks.

1. On the aspect server create a system using System Configuration Console. (A system contains all data of a project, therefore we will further talk of a system / project or, short, of a project.)



System Configuration Console must be accessed with a IndustrialITAdmin account.

2. Introduce the users for that system/project via Windows tools and System Configuration Console.
3. Configure access from the client workplaces to the system / project on the server using System Configuration Console.

For more information on system creation and related tasks to perform with System Configuration Console, refer to the section on Create System in the *System 800xA Installation Update and Upgrade Getting Started (2PAA111708*)*.

Load Libraries and Standards

1. Install and load the libraries for the target control system. For AC 800M the object type libraries are loaded with the AC 800M Connect system extension.
2. Install and load basic standard object type libraries such as Basic Object Type Library.
3. Install and load further standard object libraries and typical solution libraries needed for the project and depending on libraries installed and loaded in step 1 and 2. For example for AC 800M signal engineering you need the CBM_Signals object type group loaded with the AC 800M Signal Extension system extension.



Depending on the library you include it into your project either via loading a corresponding system extension or via import using the Import/Export tool (AfwImport/Export).



Load Libraries can also be done by using Application Change Management (ACM). For more information, refer to *System 800xA Engineering Application Change Management (2PAA108438*)*.

Available standard libraries for AC 800M are briefly described in the *System 800xA AC 800M Binary and Analog (3BSE035981*)*.

Customize System

During and after system creation some settings/actions are required to ensure that the system can be used by application engineers in the engineering project effectively. A collection of such settings and actions is described in the *System 800xA Post Installation (2PAA111693*)*.

Build Project Library

Assumption is that the basic analysis and design is already done and the results are available to be used as data for example in tabular formats (for example in data base or spreadsheet tables) and overview drawings (such as P&IDs).

You have identified common equipment and devices from P&IDs and you have selected reusable entities.

Whenever possible you map the reusable entities to applicable Object Types / Composite Object Types from existing standard libraries. Only if no applicable object type is available, you implement these reusable entities as their own object types / composite object types of their own versioned libraries or as extensions to existing object types / composite object types.



First task to perform when doing engineering within the 800xA system is to obtain an enough powerful tool-box of reusable objects, the object types / composite object types you need to build the automation solution.

These objects may be derived from several different sources: Some are already included in the standard libraries of AC 800M Connect / Control Builder M Professional, some are included in libraries made by ABB End User Divisions or OEMs, some may come from previously made projects, and some you need to build yourself.

Whenever possible base these objects on the objects delivered with the standard libraries of System 800xA.



Whenever possible also Extended Configuration issues should be taken care in building the object types / composite object types of the Project Library. In general as little instance specific additions as possible should be done.

Complete Project Library

In this working step you do adaptations and introduce extensions to the standard project environment regarding the solutions to reuse. In general you finally specify and build the common base for reuse of solutions in your engineering project:

- Versioned Object Type Libraries in the Library Structure.
- Object Types / Object Type Groups in the Object Type Structure.
- Reusable elements in the Library Structure:

- Document Manager Templates.
- Alarm & Event Configurations.
- History Log Templates.
- Trend Templates.
- Scripts.
- Function Diagram Templates in the Object Type Structure.
- Topology Diagram Templates in the Object Type Structure.
- Reuse Instructions in the Reuse Design Structure and corresponding Object Types in the Object Type Structure.

Versioned Object Type Libraries

Whenever applicable you create your own new and adapted object types and composite object types in your own versioned object type libraries. This gives you the path for controlled introduction of new versions of object types supported by Control Builder M Professional.

Steps and examples how to create a versioned library and how to switch to a new version is described in the subsection on Libraries in the *System 800xA Configuration (3BDS011222*)*.

Build New or Adapt Existing Object Types

How you build new Object Types and Composite Object Types or adapt existing object types in general is described in the subsection on Building Object Types in the *System 800xA Configuration (3BDS011222*)*.

How to build object types in Control Builder M Professional is described in the *System 800xA Control, AC 800M Configuration (3BSE035980*)*.

Any of the five IEC 61131 languages are preferably used when building the control logic in the Composite Object Types.

Function Designer can be used to build control module types too if Function Diagrams on type level are required.

Examples how to build your own composite Process Object Type with Signal objects of AC 800M Signal Extension / object type group CBM_Signals that supports functional planning are given in the *System 800xA Configuration (3BDS011222*)*.

Specifications and Guidelines

Another work part for completing the project library is to collect and/or write and adapt the specification, description, guideline and template documents for your project.

At this stage of the project work you can store all your project documents and document templates on a work directory on your local disk.

The templates you can introduce as Document Manager templates in the Library Structure.

If the templates shall be used in object types of your Project Library you extend these object types with corresponding document aspects.

After the basic Documentation Structure (refer to [Create Basic Structures](#) on page 204 below) has been built the specification, description, guideline documents can be made available in folder/binder objects of Documentation Structure using Document Manager aspects.

Typically there are a lot of project documents. The objects with the document aspects can be created/imported using Bulk Data Management functions.

Corresponding example instructions can be found in the *System 800xA Configuration (3BDS011222*)*.

Create Basic Structures

In this working step you create the upper level plant structures (Functional, Control, Location, Documentation) from object types of your project library.

Depending on the design of the library the objects on upper level are of a descriptive / structuring kind or they already have a relevant process function part.

Typically a plant is modeled in three structures:

- Functional Structure.

- Control Structure
Be aware that the main structuring of the Control Structure in the default case for a 800xA system is given by AC 800M connectivity.
- Location Structure.

The documentation of a plant can additionally be modeled in Documentation Structure.



Which structures to use in an engineering project depends on the concrete requirements. The recommended minimal set is to use Functional Structure and Control Structure (Figure 27). Functional Structure then contains objects with Graphic Displays and, if the Functional Planning approach is used to configure Control Applications, Function Diagrams. For certain projects it might be sufficient to work in Control Structure only.

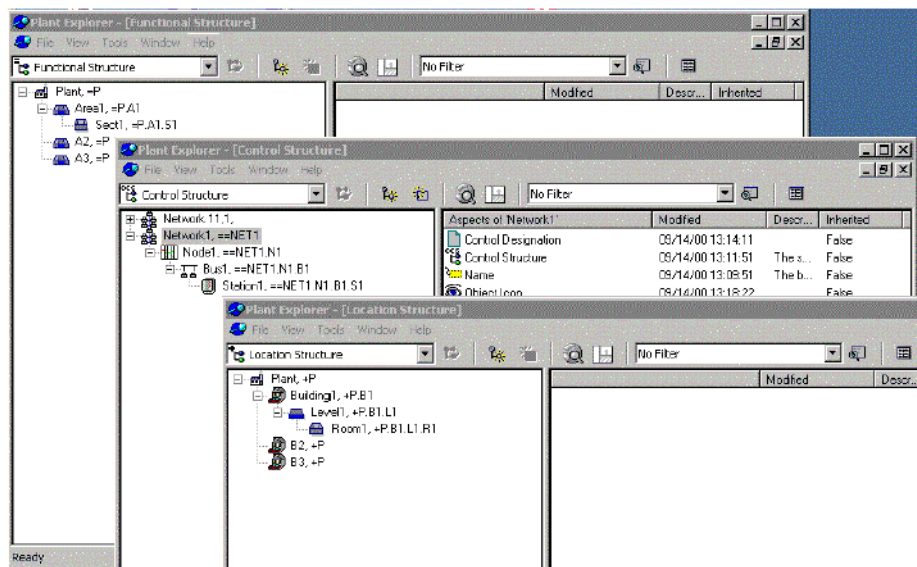


Figure 27. Basic Structures: Functional, Control and Location

Building *Basic Structures* in this context means always to specify the hierarchy of the upper level for the relevant structures by use of Aspect Objects.

The *Basic Structures* provide the base for adding design information in later stages:

- Add information as additional aspects.
- Add information by additional Process/Location/Control (System) /Document Objects.

Identification of Aspect Objects

Each level in the *Basic Structures* (represented by Aspect Objects) should be identified unambiguously in accordance to the naming / designation system used by customer. Refer to [Names and Designations](#) on page 316.



Be sure to include needed designation aspects already in the object types of your project library.

Principles of Reference Designations, usage of Function Reference Designations, Control Reference Designations, Location Reference Designations, and Documentation Reference Designations are described [Appendix B, Structured Plant Modeling](#).

Importing Design Information

Result of the analysis and design process is a certain amount of design information describing aspect objects and aspects. Typically the design information are / can be made available as data in relational databases or spread sheets as lists.

The tools of the Engineering Workplace supports you in solving the problem to bring in the design information into the structures / objects of your system / project.

The data representing the design information can be entered into Plant Explorer manually object by object: Create object from object type and enter the property values.

Or, which is much more efficient when an lot of objects result from the analysis and design process, the data available in relational databases or spread sheets are used to create the aspect objects from object types and setting the object properties automatically using Bulk Data Management functions.

Starting data entry in Plant Explorer manually, for example just for the first two structure levels, does not exclude switching to list based configuration with Bulk Data Management when it comes to real bulk work. You can easily extract the

manually entered data to a Bulk Data Management worksheet, refine it, extend it and then write it back to the system / project.

Functional Structure

The example in [Figure 28](#) shows a basic Functional Structure imported using a Bulk Data Manager worksheet. How you can configure such a worksheet is described in *System 800xA Engineering Engineering Studio (3BDS011223*)*.

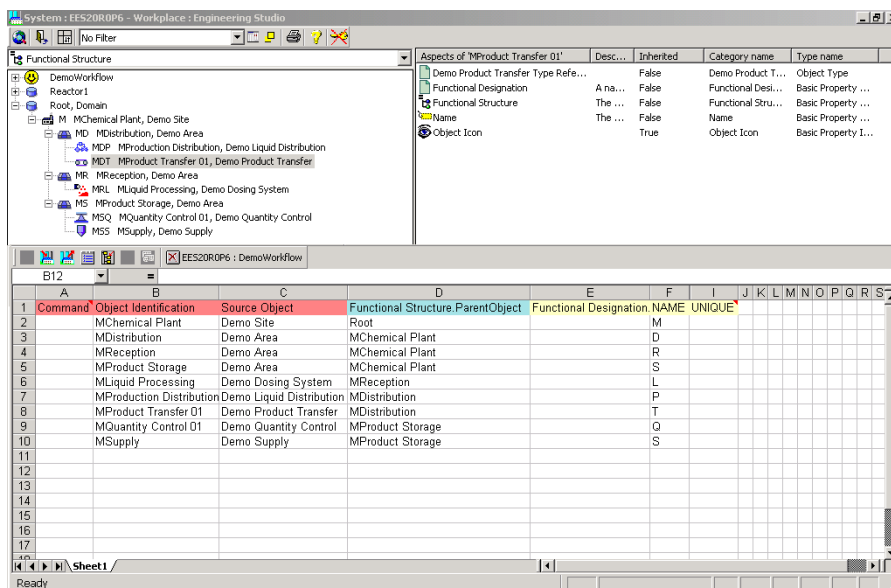


Figure 28. Basic Function Structure Imported via Bulk Data Manager

Importing design data via Bulk Data Manager as shown in [Figure 28](#) can be applied to all structures. If available, functional layout diagrams and descriptions are added to the objects via Document or AutoCAD_Drawing aspects.

Control Structure

At an early stage of the project work Control Structure might be specified by basic analysis and design considerations. Below the Control Network there are several Control Projects splitting the configuration work into reasonable part projects.

Below the Applications folder of each Control Projects there are a number of Applications assumed. The Applications each are connected to at least one Controller of the controllers under the Controllers folder. (More often several applications are connected to only one controller than one application is distributed in several controllers.) The number of controllers needed to run the applications are estimated. It might be that the Hardware units of the controllers are already estimated. You can create basic Control Structure in Plant Explorer or in Control Builder M Professional object by object or you can create it from a Bulk Data Manager worksheet.

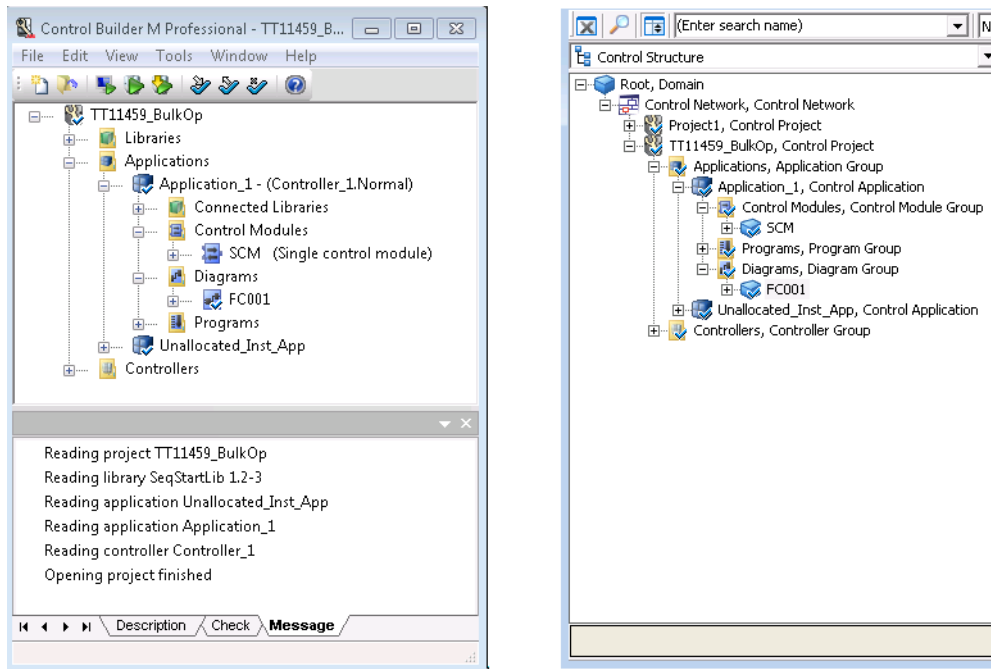


Figure 29. Basic Control Structure in Control Builder M Professional and Plant Explorer

How to design basic Control Structure depend on the concrete project requirements and on the experience how applications can be run on controllers. Some design hints are given in the *System 800xA Control AC 800M Planning (3BSE043732*)*.

Location Structure

The basic Location Structure in this context includes at least the topographic locations like buildings, floors, rooms,....

It should be detailed down to the level that is necessary for clarification in this stage.

If available, topographic layout diagrams and descriptions are added to the objects using Document or AutoCAD_Drawing aspects.

You can create the basic Location Structure in the Plant Explorer object by object or you can create it from a Bulk Data Manager worksheet.

Documentation Structure

The documentation of the project is modeled by folder/binder and sub folder/grip/package objects and document objects.

Already at this stage you can build the basic folder/grip/package structure of your project's Documentation Structure. You can create the basic Documentation Structure in the Plant Explorer object by object or you can create it from a Bulk Data Manager worksheet.

Next you can create documentation objects within the structure that contain the standard documents and templates in your project. You can do this by configuring a Bulk Data Management worksheet that creates the Objects and Documentation aspects and imports the document files.

Additionally you can insert the objects of basic Functional Structure, Control Structure and Location Structure holding Document or AutoCAD_Drawing aspects beneath corresponding folder/grip/package objects in Documentation Structure.

You can build up Documentation Structure to hold the whole user-created documentation base for the 800xA system, for the engineering project work and for the plant operation and maintenance. This gives you quick and structured access to all needed information contained in documents.

Create Process Functions

Build Lower Level Functional Structure

What is considered to be the start of the lower levels in Functional Structure depends on the granularity of the objects you deal with to build the process function part of your project.

Presuming the object types delivered with AC 800M connectivity of System 800xA you deal here with Process Objects (for example of the ProcessObjExtLib), Control Loops (for example of the ControlBasicLib) and corresponding Signal objects provided additionally by the AC 800M Signal Extension / object type group CBM_Signals of Engineering Workplace.

You instantiate the object types object by object using Plant Explorer or Control Builder M Professional or using Bulk Data Management worksheets. The latter is recommended when you have to create many objects, have to parameterize their properties or if you expect several change cycles.

If you have built composite Process Object types or Control Loop object types including the required Signal objects as shown in the corresponding examples in the *System 800xA Configuration (3BDS011222*)*, the instantiation results in a complete Process Object or Control Loop including the function components for the Function Diagram of the process function.

To build a Function Diagram of a process function in an object of the next level above some Process Objects/Control Loops you create a Function aspect in this next level object and configure a new diagram within Function Designer. The function component representations for the objects of the Process Objects/Control Loops below are available in this Function Diagram immediately and only have to be connected and placed.

Build Lower Level Control Structure

Assuming you have already created Control Structure including the level of application objects and including the Controllers you now create the Communication Boards, I/O Stations, I/O Boards from Object Types of the Project Library as required. You do this either object by object Plant Explorer or Control Builder M Professional or Bulk Data Management worksheets. The latter is

recommended when you have to create many objects, have to parameterize their properties or if you expect several change cycles.

Build Lower Level Location Structure

You create location objects like Floor/Room/Coordinate from Object Types of the Project Library. You do this either object by object using Plant Explorer or Control Builder M Professional or with Bulk Data Management worksheets. The latter is recommended when you have to create many objects, have to parameterize their properties or if you expect several change cycles.

Modify/Allocate Process Functions

Add Specific Control Logic

In the Function Diagrams of the process functions you can add glue and interlock logic at your requirements. Refer to the *System 800xA Engineering Engineering Studio Function Designer (3BDS011224*)*.

Assign I/O Signals

If you have used object types as shown in corresponding examples in *System 800xA Configuration (3BDS011222*)* the needed engineering Signal objects are already available and in place. If not you have to create them (could be you have done it during [Build Lower Level Functional Structure](#) on page 210) and insert them beneath the process object in Functional Structure. Both can be done using Bulk Data Management worksheets.

Allocate Process Functions to Applications in Control Structure

The function components (including the control logic) of the process functions have to be assigned to allocatable groups. How you can do this in detail in Function Designer or in Plant Explorer is described in the *System 800xA Engineering Engineering Studio Function Designer (3BDS011224*)*. By default the Function Diagram of a process function object corresponds to an Allocatable Group (the object contains an Allocatable Group aspect).

An object with an Allocatable Group aspect has to be inserted below the Control Modules folder of an Application object in Control Structure. This results in generation of Control Builder M Professional, containing all functions, function blocks and control modules contained in the Function Diagram and covered by the Allocatable group. The Application object in turn is connected to a Task of a Controller.

Allocate I/O Signals to Boards in Control Structure

The I/O Signal objects (of AC 800M Signal Extension / object type group CBM_Signals) have to be allocated to corresponding channels of I/O boards. This you easily can do with the smart I/O-Allocation function and the **Write Allocation to CBM** function both provided in the object context menu of Engineering Workplace. If the Signals have already been allocated before the Function Designer itself executes the **Write Allocation to CBM** function part. Refer to the *System 800xA Engineering Engineering Studio (3BDS011223*)*.

Design/Build Graphic Displays

Finalize Graphic Displays from Object Types

If you have instantiated object types on the process function level with preconfigured graphic displays you now can finalize these displays using graphic elements defined in the Process Object, Control Loop and other object types from your Project Library. How you do this in detail is described in the *System 800xA Engineering Process Graphics (3BSE049230*)*.

Design and Build New Upper Level Graphic Displays

If you do not have already preconfigured graphic displays in the process function objects you create Graphic Display aspects and build the corresponding displays from scratch. You use graphic elements defined in the Process Object, Control Loop and Signal object types from your Project Library. How you do this in detail is described in the *System 800xA Engineering Process Graphics (3BSE049230*)*.

Extend Configuration



Whenever possible also Extended Configuration issues have to be taken care in building the object types of the Project Library. In general as little instance specific additions as possible should be done.

Add Production Management Functionality

Refer to the *System 800xA Configuration (3BDS011222*)* for principal information how to add Batch Management functionality.

Add Information Management Functionality

How to handle documents in Documentation Structure already is described in subsections on Documentation Structure.

Refer to the *System 800xA Configuration (3BDS011222*)* on information how to add information management functionality.

Add Asset Optimization Functionality

Refer to the *System 800xA Configuration (3BDS011222*)* for principal information how to add Asset Optimization functionality.

Finalize Lower Level Location Structure

You do final updates of the already existing location objects and add or insert additional objects in Location Structure. Depending on the size of the plant at least a location overview of the most important hardware units and process functions should be reached.

Finalize Documentation Structure

You do final updates of the already stored documents and add additional document aspects or document objects with document aspects. In general all needed project documents and all needed operation / maintenance documents should be accessible via Documentation Structure.

Generate Application Software

As Control Builder M Professional works on one Control Project in Control Structure at a time the application software for the controllers has to be generated Control Project by Control Project.

For one Control Project: When Signal objects (of AC 800M Signal Extension / object type group CBM_Signals) have been allocated to their I/O boards in Control Structure and all allocatable groups of Function Diagrams have been allocated to Application objects in Control Structure the application software for the Control Project is already generated.

Before downloading to a real AC 800M controller you can test the application software using the Test Mode in Control Builder M Professional or by activating Simulate Hardware and downloading to SoftController. Refer to the *System 800xA Control, AC 800M Configuration (3BSE035980*)* and *System 800xA Engineering Engineering Studio Function Designer (3BDS011224*)*.



A restriction exists for the hardware simulation functionality which is intended for usage in a separate testing Control Network. Do not run hardware simulation controllers on a Control Network containing actual production controllers.

Download and Test Application Software

Deploy Graphic Displays

After development of the Graphic Displays they have to be deployed to be ready to be used in the system. Refer to the *System 800xA Engineering Process Graphics (3BSE049230*)*.

Download Controller Code

You download the control application software per Control Project using Control Builder M Professional **Tools > Download Project and Go Online**. This also the way to load changes. They are detected and loaded automatically.

If a changed Control Application has to be evaluated online you use the Evaluate option during download. This enters the Load-Evaluate-Go (LEG) procedure. Refer to [Load-Evaluate-Go](#) on page 236 for more information.

Refer to the *System 800xA Control, AC 800M Configuration (3BSE035980*)*.

Load other Run-time Configuration Data

Depending on your 800xA system's hardware and software configuration you might have to load or introduce additional configuration results on servers / clients to be ready to test. Refer to the configuration instructions of the corresponding functional area.

Test Complete Configuration

You use Control Builder M Professional and Function Designer in Online mode to verify the functionality of the control application software running on the controllers. You use the given 800xA system's hardware and software configuration to test all functions of your application software: From the I/O hardware to the displays, printers, archives, alarm devices, etc. and back. Depending on the test process simulation you have available/ have developed you include also process simulations in your test. Refer to the *System 800xA Control, AC 800M Configuration (3BSE035980*)* and *System 800xA Engineering Engineering Studio Function Designer (3BDS011224*)* and further corresponding configuration instructions.

Site Engineering Workflow

In an (on-)site engineering scenario with Production System (PS) and Engineering System (ES) the project workflow extends as shown in [Figure 30](#). Import/Export can be used for synchronizing changes.

Also refer to [Configuration Change Management](#) on page 188 and [Import/Export](#) on page 272 and the corresponding subsections in the *System 800xA Configuration (3BDS011222*)*.

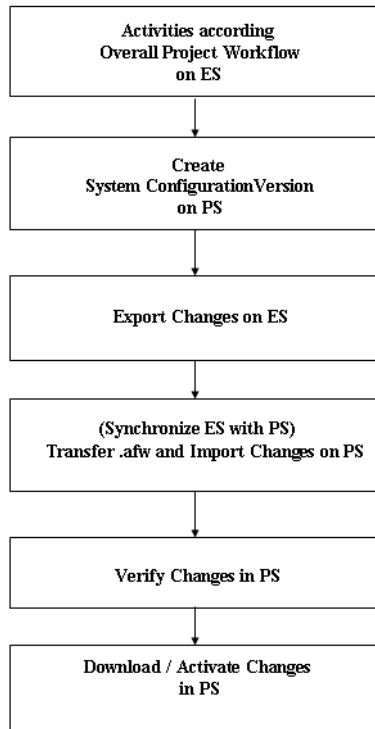


Figure 30. Extended Project Workflow for Site Engineering

Table 25. Workflow Activities Site Engineering

Workflow Step	Activities
Activities according overall project workflow on ES Refer to Figure 26 , above.	Refer to Table 24 .

Table 25. Workflow Activities Site Engineering (Continued)

Workflow Step	Activities
Create System Configuration Version on PS	Define System Configuration Version in System Configuration Version Definition object /aspect in the Maintenance Structure. Execute Creation of the System Configuration Version (full backup).
Export on ES	Export known changes: Entities with or without additional dependencies.
(Synchronize ES with PS) Import on PS	Transfer archive with changes. Import Archive. If Configure-Deploy or Load-Evaluate-Go are going be used on PS: Import Archive into Engineering Environment.
Verify Changes in PS	Verify trees/objects/entities changed by synchronization. If OK: Continue. Else: Do corrections either on PS or ES.
Download / Activate Changes in PS	If applicable: Download/activate changes. If online evaluation required: Use Load-Evaluate-Go during download on the controller.

Section 15 Configuration Tools

Automation Solution

In an 800xA system Automation Solutions are composed of:

- Controller software parts (control logic).
- Graphic displays, faceplates and trend displays for process operation.
- Alarm and event functionality and related functionality such as logging and messaging.
- Controller/ I/O and Fieldbus/Field device communication parts.
- Field device control software (if it can be configured).
- Graphic displays for diagnostic and test.

These parts together implement an Automation Solution.

The 800xA system allows you to treat these parts and even more parts together in Aspect Objects representing the control solution.

Control Logic

The tool to configure control logic for an 800xA system with included AC 800M controller connectivity is Control Builder M Professional.

Besides flexible basic control configuration functions based on IEC61131 languages and the additional Control Module and Diagram concept, Control Builder M Professional provides you with Function Designer.

This tool enables you to configure control logic according to a Functional Planning approach.



Control Builder M Professional additionally is the certified tool to configure SIL certified applications. SIL stands for Safety Integrity Level, as specified in the standard IEC-61508. To run SIL certified AC 800M applications, you need the SIL certified AC 800M High Integrity controller.

Function Designer is not certified for building SIL-certified applications. It can be used to configure non-SIL-certified applications for the AC 800M High Integrity controller.

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

SIL-application related configuration procedures to perform with Control Builder M Professional are described in the *System 800xA Control, AC 800M Configuration (3BSE035980*)*.

With Control Builder M Professional including Function Designer you can follow three different approaches to configure control logic:

- [IEC61131 Function Block/Program Programming](#) on page 223.
- [Control Module Programming](#) on page 223.
- [Diagrams](#) on page 224.
- [Functional Planning with Function Designer](#) on page 226.

Base languages in each of these cases are the IEC61131 control languages.

While the first approach typically is used for PLCs, the second approach (involving deep Control Module structures) extends the first approach in an ABB-specific way for DCS-like systems. It was introduced with the ABB SattLine system.

Diagrams are suitable for creating logic in a graphical editor by mixing function blocks, control modules, diagrams, and functions. The inserted objects can be connected graphically through ports, to create the logic. This represents a complete graphical overview of the whole logic.

System-wide Functional Planning is a typical DCS configuration strategy not limited to control logic. Regarding control logic it bases on the first two approaches.

Which approach to follow in a project depends on the customer's requirements, on technical design issues and, to a certain extent, on preferences and experiences of the project team.

Some basic information on configuration of control logic according the two first controller or task oriented approaches you find in [Function Block/Program Programming](#) on page 223 and [Control Module Programming](#) on page 223. Comprehensive information on these topics is contained in the *System 800xA Control AC 800M Planning (3BSE043732*)*.

The latter manual among other topics describes the basic design issues to take care for Function Block, Control Module, and Diagram Type programming. These basic design issues also apply to control logic configured using Function Designer.

Functional Planning allows you to work in comprehensive Function Diagrams that can cover control logic, engineering signals, field devices and even representations of information items of other aspect systems important for the plant functionality.

Function Diagrams allow you to configure, commission, maintain and document the control applications of a plant from a functional perspective. To configure control logic within Function Diagrams you insert Function Components representing Control Modules and / Function Blocks or Functions from AC 800M Connect libraries or industry specific libraries, connect them graphically and generate them into diagrams allocated to a control application in Control Builder M Professional.

The main engineering project requirements that can be fulfilled by using Function Designer are:

- Configuration driven by Functional Structure (functional decomposition) of the plant.
- Control application independent planning in early planning stages. Control logic allocation to application can be done in a late stage.
- Support for IO allocation tool: Allocation of engineering signals to hard ware units can be done in a late planning stage.
- Simplified graphical Function, Function Block, Control Module and Sequence Programming in Function Diagrams.
- Extended Name Handling:
 - Non-IEC61131 compatible object names can be used.

- Name synchronization between Object Name, Control Builder Name and Name property of Function Block or Control Module.
- Rename during copy.
- Electronic Function Diagram documentation (offline and online) to ease commissioning and maintenance.
 - Diagram reference based horizontal navigation following the signal flow.
 - Diagram page reference based sheet navigation in a diagram following the signal flow.
 - Diagram based vertical navigation following the diagram hierarchy in the Functional Structure, going from parent diagram to child diagram and back.
 - Live data display and value forcing in online mode.
 - Context sensitive aspect navigation from within diagrams.
- Bulk printing of Function Diagram documentation including generated contents documentation and generated hidden items documentation.
- Automatic creation of connection variables.
- Automatic control logic and variable generation for Inter Application Communication (IAC) using communication variables.

Function Components are not restricted to control logic elements as provided by AC 800M Connect libraries and industry specific libraries. By default Engineering Signals and Field Devices are supported additionally. The ability to configure own Function Components allow to represent other elements in Function Diagrams if needed.

Basic concepts of Functional Planning are described in [Functional Planning with Function Designer](#) on page 226, more details are described in the *System 800xA Engineering Engineering Studio Function Designer Getting Started (3BDS100968*)* and *System 800xA Engineering Engineering Studio Function Designer (3BDS011224*)*.

Function Block/Program Programming

The IEC 61131-3 standard describes programs, function blocks, and functions as Program Organization Units (POUs). These units help you organize your control project into code blocks, to minimize code writing, and to optimize code structure and code maintenance.

A POU can be described as an object type that contains an editor, where you can write code and declare parameters and variables. All POUs can be repeatedly used in a hierarchical structure, except for programs that can only be a top-level POU, inside an application.

The use of Functions and Function Blocks encourages well-structured design which in return will speed up program development. This is especially true in plants containing a lot of similar control functions which means that Functions and Function Blocks can be reused many times.

Reuse is accomplished by first establishing a Function Block Type and then creating one or many Function Block Instances of that type.

The Function Block Instance only contains data that are unique to the particular set of Function Block Type. This means that although a Function Block Type may use a very complicated algorithm the memory required to store the data for each Function Block Instance may be very modest. An application program based on several Function Blocks therefore requires less memory than programs containing duplicates of code.

Functions and Function Blocks are executed list driven and cyclic. You define execution by observing rules like “top / left” to “down / right” such that the required cycle and the enclosed actions are maintained.

Control Module Programming

Control Modules extend the IEC 61131-3 standard to support requirements of object-oriented programming and of larger applications. Besides application program code they can contain graphics to visualize and operate the implemented function during test and commissioning. They are executed data flow-driven, the compiler automatically determines the execution order by analyzing the data flow between the modules. [Figure 31](#) shows a software model for both the standard and the extensions.

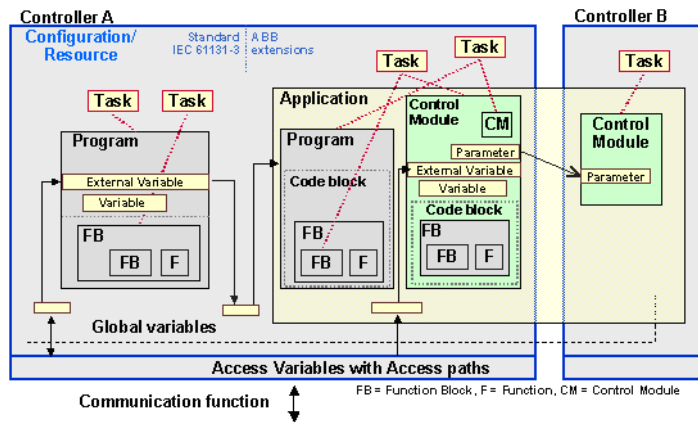


Figure 31. ABB Software Model

To use Control Modules is recommended for:

- Larger applications, because of increased performance.
- Process objects, as code sorting facilitates application structuring.
- When a flexible solution is needed: Object orientation, reuse.

Diagrams

In Control Builder, diagrams are created under an application, and diagram types (which can be reused as instances in a diagram) are created under a library or under the same application as the diagram.

The graphical code block in diagrams and diagram types allow mixing of functions, function blocks, control modules, and other diagrams, and graphically connect them to achieve a particular logic. A diagram can then be connected to a controller task and the logic can be executed in the controller. The logic can also be extended across different applications by using communication variables declared in the diagram editor.

In addition to the graphical code block, the diagram also supports code blocks of Structured Text and Sequential Function Charts, which are invoked from graphical code block or sorted separately.

Another advantage of using diagram for the design with SIL applications is that the graphical code block in a diagram of higher SIL can accept a signal of lower SIL, if required, using communication variables. The graphical code block also displays such a lower SIL signal with a distinct color.

The main differences among Function Blocks, Control Modules, and Diagrams are listed in [Table 26](#).

Table 26. Differences among function blocks, control modules, and diagrams

Property	Control modules	Function Blocks	Diagrams or Diagram Types
Container POU inside the application	Diagrams or Single Control Modules	Programs, Diagrams or Single Control Modules	Diagrams
Graphical connections between objects	Yes	Yes	Yes, including graphical connection to parameter/variable objects.
Code sorting	Yes.	No.	Yes. All code blocks, in control modules inside the diagram type, are sorted together with the code blocks outside the diagram type in the container Diagram.
Execution	Control modules are executed only once per scan.	Function blocks are executed from code. Therefore, a function block is executed once or several times per scan, or it is not executed at all.	The execution order shown in the diagram is followed. Codeblocks from control modules are sorted into the diagrams execution order according to control module sorting rules (writing into a variable is sorted before a read).

For details, refer to *System 800xA Control AC 800M Configuration (3BSE035980*)*.

Functional Planning with Function Designer

Function Designer allows you to configure control logic in Function Diagrams using Control Modules and Function Blocks and other Function Components (for example field devices or engineering signals).

A Function Diagram typically shows a process control loop with all its function components and their signal connections (Figure 32).

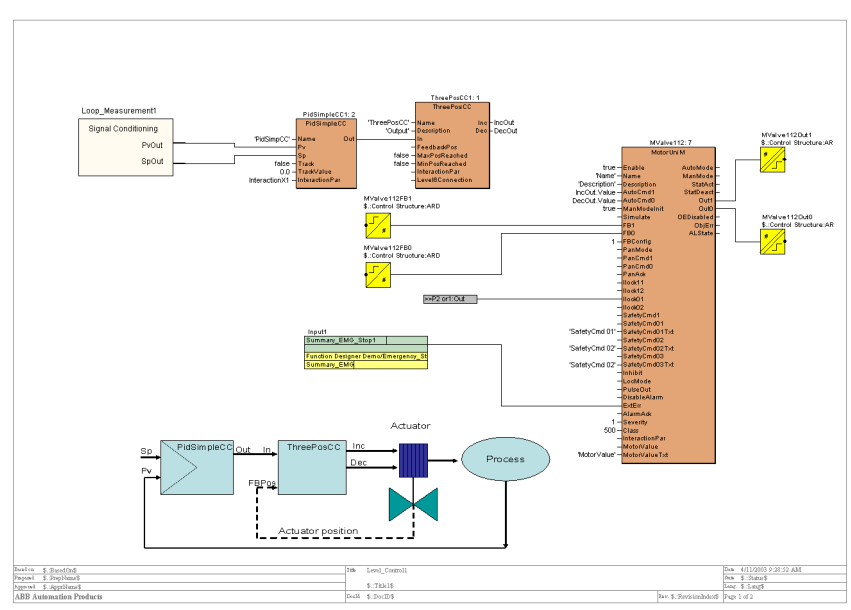


Figure 32. Example Function Diagram

With Function Designer you are able to perform graphical plant function oriented engineering with Aspect Objects. This is different from the traditional controller or task oriented engineering. The graphical user interface makes complex control strategies easy to engineer and even easier to maintain and document.

The Function Designer provides you with a Component view and a Diagram view on a Function Aspect.

Function Components. In Component view you design a graphical symbol for a Function Aspect representing an Object Type or Composite Object Type from the functional perspective. Instances of such a Function Component object type you can insert in a Function Diagram and connect the graphically based on the representing graphical symbol.



For object types of AC 800M / Control Builder M Professional standard libraries you need not design graphical symbols yourself: After loading Function Designer system extensions into your system these object types are equipped with Function aspects and default graphical symbols. The same applies for HART device object types supplied by HART Device Library and Device Library Wizard.



For object types you have designed for AC 800M / Control Builder M Professional you only need to add a Function aspect to the object type to make it work as a Function Component with a fitting default graphical symbol.



HART and PROFIBUS devices are supported as function components by default. Additionally also referencing the IO variable representing the device in the control application using a diagram reference can be used.



By default FOUNDATION Fieldbus devices are not supported as function components.

There are two possibilities to support documentation and navigation for FOUNDATION Fieldbus devices in Function Diagrams of Function Designer:

- * Referencing the IO variable representing the device in the control application using a diagram reference.

- * Adding an own Function aspect with a Function Component to the device type.

Refer to the *System 800xA Engineering Engineering Studio Function Designer (3BDS011224*)*.

The object type can have aspects of one or several different aspect systems. You can use basic graphic components as for example line, box, text and circle and ports referencing to a configuration entity in the represented Aspect Object to configure your specific graphical representation for an object type. You can configure

Function Components for any aspects of aspect system that you need to represent in Function Diagrams.

Control Module aspects and Function Block aspects as supported by Control Builder M Professional are the basic aspects for Function Components regarding control logic. You use the corresponding editors of Control Builder M Professional to configure these aspects in the object types / composite object types. You can use the supported IEC 61131 languages Structured Text (ST), Instruction List (IL), Sequential Function Chart (SFC), Function Block Diagram (FBD) or Ladder Diagram (LD) to write the Code Blocks for the Control Module and Function Block types.

Function Diagrams. In Diagram view you are using the graphical symbols of Aspect Objects needed to implement a certain control strategy when you arrange them on a diagram and interconnect ports of them. The ports of the Aspect Objects can be connected with an easy drag and drop operation. You can nest Function Diagrams according to the functional hierarchy. Interconnections between Aspect Objects on different pages of a diagram as well as those between different diagrams are shown as connection symbols. Connections can hide complex IEC61131 data types behind one connection line. You can follow the connection using point and click operations. Point and click navigation across Function Diagrams following the functional hierarchy of nested diagrams top down is also possible. These navigation features give you a comprehensive Function oriented view on plant wide configuration.

In a Function Diagram you also can use the set of functions from the System Library and the set of Function Blocks from the Basic Library of Control Builder M Professional. By default these Function Components are not visible as object instances in Functional Structure and Control Structure, they are only visible in the Function Diagram and in the generated code of it.

To display any reachable Aspect Object property you can insert corresponding property references into Function Components and Function Diagrams. This follows the same mechanism as known from Document Manager, Parameter Manager, Bulk Data Manager and Control Builder M Professional Control Properties aspect.

During work Diagram States are assigned to Function Diagrams that are displayed in the Function Designer window and in System Status Viewer. The basic Diagram States are modified (red), generated (yellow), and downloaded (green).

Sequences. Function Designer supports a Sequential Programming Language for the graphical design of sequences with steps, transitions and actions. A sequence can be inserted in a Function Diagram or in a Function Diagram type. Two different sequence components are supported:

- Sequence (Standard IEC 61131-3).
- Sequence2D (Two-directional Sequence, based on control library SeqStartLib).

The sequences are configuration data generated into Control Builder M Sequential Function Charts (SFCs, refer to [Sequential Function Chart](#) on page 235) and can be viewed / operated with the SFC Viewer.

Sequences designed with Function Designer enable you to work in Function Diagrams for sequence overview, steps, transitions and actions. Function Diagram related signal tracing and navigation functions and documentation functions can be used for sequences.

Connection Mapping. Connection mapping of Function Designer provides a function to connect components of variables of different structured data type. You use this function to connect two ports of different structured data type. You can also explicitly invoke the Connection Mapping command from the context menu of a link or of a variable component. In both cases you can select which substructures shall be interconnected. Necessary variables and links get automatically created.

Switching Between Aspect Object and Symbol Object. To give you the possibility to determine which Function Components shall be handled as aspect objects in System 800xA platform, Function Designer provides a function that allows Functions, Procedures, Variables, Function Blocks, Control Modules and Diagram Parameters to be switched from aspect objects to symbol objects, and vice versa.



Starting with System 800xA Version 5.0, Diagram References and the connected Diagram Variables are created as symbol objects by default. Do not convert them to aspect objects.

Symbol objects are stored and visible on a function diagram only, but not in Plant Explorer Functional Structure nor in Control Structure.

Inside a function diagram, select one or more Function Blocks, Control Modules, and use context menu item Aspect Object to switch between aspect objects and symbol objects.

Function Designer modifies the aspect properties Name, Description of the Name aspect and Control Builder Name aspect, to set the name or Control Builder name of symbol objects.

Name Synchronization. To support efficient object naming and to allow names beginning with digits (not allowed in IEC 61131-3), for example names according KKS naming rules, Function Designer synchronizes the different naming properties of Function Diagrams, Control Modules, Function Blocks, Variables, Signals in the Function Structure. For naming properties ruled by Control Builder M Professional it generates corresponding IEC 61131-3 compliant names.

Rename During Copy. The following workflow is often used with Function Designer:

1. Prepare a Function Diagram in Function Structure.
2. Copy this diagram with a Bulk Data Management worksheet in Functional Structure. The copied diagrams get new names.

Function Designer supports automatic renaming of the components within the copied diagram. It uses an algorithm to figure out, how the related components should be renamed. In addition configurable naming rules are supported. One of two naming concepts can be used:

- Naming according to a name pattern.
Two different kinds of pattern are supported: Simple and Standard.
- Automatic naming.

The actual used concept is determined by user configurable settings of Function Designer or per Function Diagram. It is possible to switch between both concepts or to switch off automatic renaming during copy/paste completely.

Data Flow Order. The data flow order influences configuration data generation, for example generation of additional move blocks for variables and additional code blocks for Control Modules.

The data flow order gets additionally displayed for

- Control Builder M signals.
- HART devices.
- Variables.
- Diagram references.

It is possible to either show or hide the data flow order for all components, but not for individual components, for example Control Modules.

A dialog allows to change the data flow order for all components. On instantiation, a default data flow order gets calculated on base of the graphical position.

Allocation and Code Generation. Allocation of the Function Components of a Function Diagram to an Application is based on a so called Allocatable Group which by default is assigned to a Function Diagram. An Allocatable Group can be allocated to/inserted below an Application in Control Structure.

The concept of allocating Allocatable Groups allows you to build your Function Diagrams in Functional Structure without taking care of the concrete control system structuring in early stages of work. You can separate functional design and configuration work from control system related structuring and configuration work. Allocatable Groups also allow you to define the execution order of the AC 800M related Function Components. An Allocatable Group is mapped to a Diagram in Control Builder M Professional.

If you have allocated/inserted a Function Diagram into Control Structure, you can generate the appropriate code in Function Designer for the control system devices (such as AC 800M controller and I/O devices) including variable and connection generation.

Bulk configuration data generation is supported in the Bulk Data Manager by context menu items and by a multiple select on the Generated column of the Allocatable Group aspect. The Generate Data option is available under the Aspect Commands context menu option.

Code of the Single Control Modules generated by Function Designer for consistency reasons cannot be changed by other Control Builder Professional tools. Changes done by Control Builder M Professional or other tools to other items (Diagram variables, object names, ...) generated by Function Designer are detected and correctly reflected in Function Designer.

I/O Signal Allocation. The I/O Signal Allocation function allows project engineers to enter I/O signal parameter data at an early stage without the necessity to create the specific I/O boards at that time. The allocation of signals to boards and the writing of signal parameters to the channels of the boards can be done in a later step.

The system extension AC 800M Signal Extension/object type group CBM_Signals provides you with signal object types (CBM_AIS, CBM_AOS, CBM_DIS, CBM_DOS, CBM_PulseS, CBM_DINTIS, CBM_DINTOS, CBM_DWIS, CBM_DWOS, and CBM_HwStatus) supporting I/O Signal Allocation. They allow to plan and configure the required I/O signal informations for functions in Functional Structure.



Do not mix up these signal objects with the Function Blocks Types and Control Module Types in the SignalLib of Control Builder M Professional. They allow to operate and supervise signals, and view and modify their parameters at runtime.

You can build Signal Groups from the CBM_Signal object types, refer to [I/O Signal Groups](#) on page 241. Regarding instantiation and allocation HART devices are treated similar as CBM_Signals.

With the I/O Allocation tool included into Engineering Workplace you can allocate the Signal instances and the HART device instances in bulk to channels of I/O board hardware units in Control Structure and, in a second step, write the allocation data to these units at the latest possible point in time. Writing allocation data includes creation of corresponding variables and connections. Also refer to I/O Signal Allocation in the *System 800xA Configuration (3BDS011222*)*.

Function Designer configuration data generation for an Allocatable Group, which includes I/O signals and HART devices already allocated to I/O boards, includes writing allocation data.

Inter Application Communication (IAC) in Function Designer. Inter Application Communication (IAC) can be configured using Communication Variable establish communication between different applications of an AC 800M controller or several

AC 800M controllers. Also, the Function Diagrams can be allocated to same or different applications of the same or different control projects.



For more details on Cross Communication between Applications, refer to *System 800xA Engineering Engineering Studio Function Designer (3BDS011224*)*.

Prerequisites

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

- All Function Diagrams containing Communication Variables must be allocated to required applications.
- Generate configuration data for these diagrams.
- Applications executing these Function Diagrams must be assigned to controllers and tasks.



For details on configuring and generating MMS Cross Communication for diagrams created prior to Engineering Studio 800xA 5.1, refer to *System 800xA Engineering, Engineering Studio Function Designer (3BDS011224*)*.

Download and Test. You download the control logic of allocated Function Diagrams using Control Builder M Professional Tool menu functions.

Loaded Function Diagrams can show Live property values and Configuration values. Besides the values of signals / page references shown by default any port can be configured (added to display list) to be shown. Items from the display list can be forced using a Watch Window provided access rights allow this. Additionally Component and Diagram view are enabled to host bitmaps and controls (e.g. ActiveX and Graphic Elements made with Graphics Builder). Further, for a Function Component inserted into a Function Diagram, you can open and operate all aspects for the component, for example the faceplate.

Applying these capabilities you can use Function Diagrams to test your control logic including signals and field devices.

List Based Configuration

Often there are many loops in a plant that equal each other except for the names, some parameter values and the assignment to Applications and Hardware Units. To configure these loops efficiently you can preconfigure the loop and then use Bulk Data Manager to instantiate or copy the needed instances. You can do it in at least three different ways:

- Configure a composite object type for the loop with the ready made Function Diagram in the Function aspect of the top object.
 - Introduce names according to naming rule support in Bulk Data Manager (same as naming rule support in Function Designer).
 - Configure the Aspect Properties of the Function Aspect to hold all properties of sub-ordinate objects needed for parametrization.
 - Introduce the needed Allocatable Groups and assign the Function Components to them.
 - Instantiate and parametrize the needed number of instances via a Bulk Data Manager worksheet. You could also do allocation to Control Structure in this or another Bulk Data Manager worksheet.
- Configure a composite object instance for the loop in Functional Structure with the ready made Function Diagram in the Function aspect of the top object.
 - Configure it as outlined above for the composite object type.
 - Copy and parametrize it as the needed number of typical solution instances using a Bulk Data Manager worksheet, if needed including allocation to Control Structure.
- Build up the composite object instances for the loops by instantiating and parametrizing the Function Component Object Types from the standard libraries, add a Function aspect to the loop object and finalize the Functions and their Function Diagrams manually instance by instance. This proceeding you could for example use if there are significant differences between the loops to take care.

Documentation Printing

To provide documentation to the customer on paper or as.pdf-files the configured Function Diagrams can be printed in bulk. A documentation package can contain:

- Configurable contents diagram, into which the contents list is generated automatically.
- Diagrams and child diagrams.
- Instances of diagram types.
- Generated documents containing additional port information.

You can print subtrees of Functional Structure or the whole Function Structure.

Control Module Graphics

You can configure the interaction window of a Control Module type with graphical representations including interaction elements for the control algorithm implemented. The interaction window then can be opened in Test Mode and Online Mode of Control Builder M Professional and can help you in test, diagnostic and commissioning of the control algorithms. For configuration of the contents of an interaction window the Control Module Diagram Editor CMD of Control Builder M Professional includes a graphical editor. The interaction window of Control Module types of the Standard libraries typically contain faceplate-like representations and interaction elements. Refer to the *System 800xA Control, AC 800M Configuration (3BSE035980*)* and Control Builder M Professional Help for further informations.

Sequential Function Chart

Sequential Function Charts (SFCs) can be configured with Control Builder M Professional and viewed on a workplace (especially on the Operator Workplace) via the SFC Viewer aspects. They also can be operated using corresponding faceplates. You can integrate and connect SFCs configured with Control Builder M into Function Diagrams by configuring a Function Component for the Control Module or Function Block implementing the SFC or by using Off-Diagram References.

Sequences designed in Function Designer (refer to [Functional Planning with Function Designer](#) on page 229) are generated into SFCs and do not need special integration effort.

Load-Evaluate-Go

Load-Evaluate-Go gives the possibility to make on-site changes to a running application, and evaluate these changes online without interfering with the running application, before either deploying or rejecting the change.

Load-Evaluate-Go is an option of download in Control Builder M Professional.

The AC 800M controller has run-time support for download and execution of two versions of an application concurrently. The version loaded to be evaluated does not control actual outputs, but a Dynamic Difference Report tool makes it possible for the user to compare behavior with the version loaded to control the process. After deployment of the evaluated version the previous version is still available to revert to. Environments provide the means for presenting the relevant versions of aspects. Four environments are used to support the Load-Evaluate-Go functionality:

- **Engineering Environment** - Holds the modified application version during engineering.
- **Production Environment** - The environment that runs the plant. It is always connected to the active application version currently controlling the process in the controller.
- **Evaluate Environment** - Holds the modified application version during evaluation. It is always connected to the new version in the controller.
- **Previous Environment** - When the evaluated application version is switched to control the process the content of the production environment is transferred to the previous running environment to enable the possibility to make a fast rollback.

The granularity for loading via Load-Evaluate-Go into the controller is one complete control application, including the local types in the object type structure. Controller Configuration (i.e., hardware configuration) only exists in one version in the controller.

Only one modified application is allowed to be loaded in a Load-Evaluate-Go session at a time. If several applications have been modified and need to be downloaded, the user can select which application to evaluate.

Basic LEG Workflow

The basic workflow elements of a Load-Evaluate-Go session are (the exact order in which to follow the steps depends on the situation):

- **Start a New LEG Session.** In the Engineering Environment, you start a new LEG session, and then deploy the modified application to the Evaluate Environment.
- **Download the Modified Application (LOAD).** In the Evaluate Environment, you download the modified application to the controller. When finished, the Evaluation Report is displayed.
- **Evaluate the New Application Version (EVALUATE).** You use the displayed Evaluation Report to compare the new and old application versions, and evaluate detected differences.
- **Abort the Evaluation.** If necessary, you can abort the evaluation. The new application version will be removed from the controller.
- **Switch Process Control (GO).** After the evaluation, you switch process control from the old application version to the new application version.
- **Verify Stable Process.** When the new application version controls the process, you verify that the process remains stable, and that everything is as expected.
- **Revert to the Old Application Version.** If necessary, you can switch process control back to the old application version. (Then, you abort the evaluation).
- **Accept the New Application Version.** When you have verified that the process remains stable, you accept the new application version. The old application version will be removed from the controller.
- **End the LEG Session.** When you either have accepted the new application version or aborted the evaluation, the LEG session ends automatically. The Load Evaluate Go is now finished.

For further information on LEG, refer to *System 800xA Engineering Engineering and Production Environments (3BSE045030*)*.

Further Readings

Basic configuration steps using Function Designer are found in the *System 800xA Configuration (3BDS011222*)*. Tutorial information is found in the *System 800xA Engineering Engineering Studio Function Designer Getting Started (3BDS100968*)*. A comprehensive description of Function Designer functionality is given in the *System 800xA Engineering Engineering Studio Function Designer (3BDS011224*)*. Especially refer to subsections on Function Designer AC 800M Integration.

I/O Allocation is described in the *System 800xA Engineering Engineering Studio (3BDS011223*)*.

An introduction to the IEC 61131 languages is found in the *System 800xA Control AC 800M Planning (3BSE043732*)*. A description of how to work with Control Builder M, programming Control Module types and Function Block types is found in the *System 800xA Control AC 800M Getting Started (3BSE041880*)* and *System 800xA Control, AC 800M Configuration (3BSE035980*)* and in Control Builder M Professional Help.

Control Hardware Units

Hardware Unit object types are provided by AC 800M Connect in versioned hardware libraries.

You can create Hardware Units such as AC 800M Controller and CPU, communication units of the controller (Ethernet, Com, PPP, ModuleBus) and I/O boards beneath ModuleBus using Plant Explorer and you can configure them in the Hardware Editor of Control Builder M Professional as single units one by one ([Figure 33](#)).

If your engineering task is to create and configure a lot of these units, especially I/O boards, you use Bulk Data Manager. From a Bulk Data Management worksheet you can create the I/O boards and parametrize (configure) their properties from a list in one shot as shown in the example [Figure 34](#).

I/O Signal Allocation

When you do Signal engineering in Functional Structure using the I/O Signal objects of the AC 800M Signal Extension / object type group CBM_Signals you

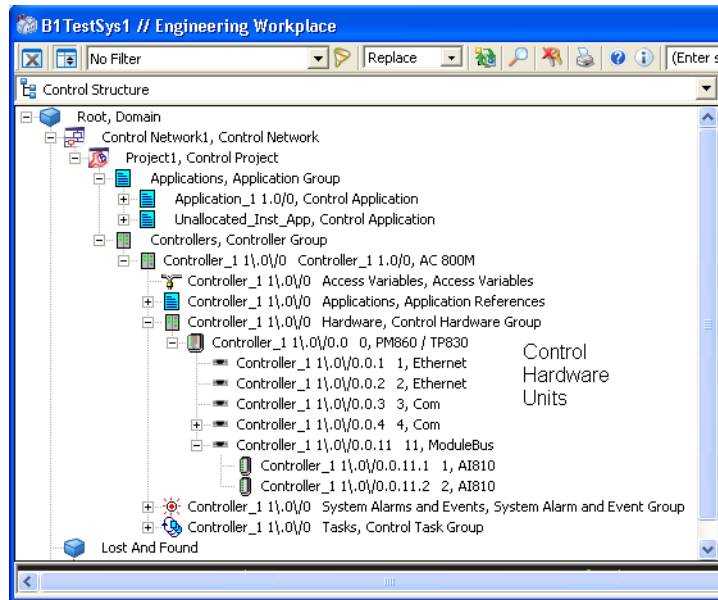


Figure 33. Control Hardware Units in Plant Explorer

Command	Object Identification	Source	Control Structure	Parent Control	Control Desig Name
2	[DIRECT][Control Structure]Root.Control Network.CSO.Controllers.AC800M_Con1.Hardware.0.11.1	AB10	[DIRECT][Control Structr	Control Desig	1
3	[DIRECT][Control Structure]Root.Control Network.CSO.Controllers.AC800M_Con1.Hardware.0.11.2	AB10	[DIRECT][Control Structr	Control Desig	2
4	[DIRECT][Control Structure]Root.Control Network.CSO.Controllers.AC800M_Con1.Hardware.0.11.3	AB10	[DIRECT][Control Structr	Control Desig	3
5	[DIRECT][Control Structure]Root.Control Network.CSO.Controllers.AC800M_Con1.Hardware.0.11.4	AB10	[DIRECT][Control Structr	Control Desig	4
6	[DIRECT][Control Structure]Root.Control Network.CSO.Controllers.AC800M_Con1.Hardware.0.11.5	AB10	[DIRECT][Control Structr	Control Desig	5
7	[DIRECT][Control Structure]Root.Control Network.CSO.Controllers.AC800M_Con1.Hardware.0.11.6	AB10	[DIRECT][Control Structr	Control Desig	6
8	[DIRECT][Control Structure]Root.Control Network.CSO.Controllers.AC800M_Con1.Hardware.0.11.7	DB10	[DIRECT][Control Structr	Control Desig	13
9	[DIRECT][Control Structure]Root.Control Network.CSO.Controllers.AC800M_Con1.Hardware.0.11.8	DB10	[DIRECT][Control Structr	Control Desig	14
10	[DIRECT][Control Structure]Root.Control Network.CSO.Controllers.AC800M_Con1.Hardware.0.11.9	DB10	[DIRECT][Control Structr	Control Desig	15
11	[DIRECT][Control Structure]Root.Control Network.CSO.Controllers.AC800M_Con1.Hardware.0.11.10	DB10	[DIRECT][Control Structr	Control Desig	16
12	[DIRECT][Control Structure]Root.Control Network.CSO.Controllers.AC800M_Con1.Hardware.0.11.11	DB10	[DIRECT][Control Structr	Control Desig	17
13	[DIRECT][Control Structure]Root.Control Network.CSO.Controllers.AC800M_Con1.Hardware.0.11.12	DB10	[DIRECT][Control Structr	Control Desig	18
14	[DIRECT][Control Structure]Root.Control Network.CSO.Controllers.AC800M_Con2.Hardware.0.11.1	AB10	[DIRECT][Control Structr	Control Desig	1
15	[DIRECT][Control Structure]Root.Control Network.CSO.Controllers.AC800M_Con2.Hardware.0.11.2	AB10	[DIRECT][Control Structr	Control Desig	2
16	[DIRECT][Control Structure]Root.Control Network.CSO.Controllers.AC800M_Con2.Hardware.0.11.3	AB10	[DIRECT][Control Structr	Control Desig	3

Figure 34. Import I/O Boards

finally have to allocate the Signal objects to the channels of the I/O boards, configure the channel properties, and create the I/O variable. The I/O Allocation function and the function Write I/O Allocation to Control Builder M relieves you

from doing these tasks manually item by item. Both are available in the object context menu of Engineering Workplace.

The smart I/O Allocation function gives you a tabular view on the I/O boards and their channels and on the Signals. The Signals can be allocated to the channels using multiple selection and drag and drop, as outlined by [Figure 35](#).

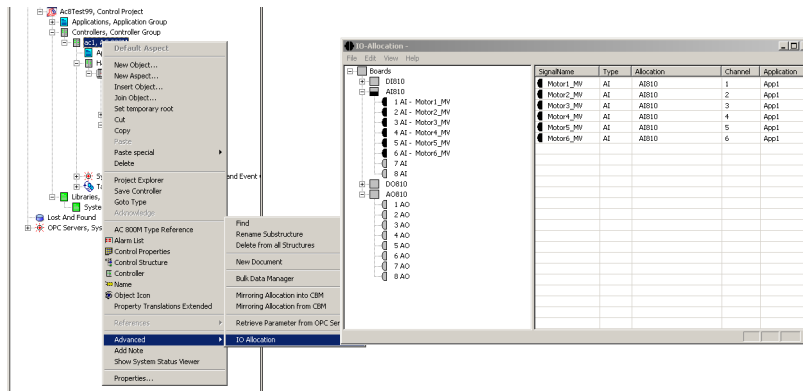


Figure 35. Advanced Context Menu and I/O Signal Allocation



The Advanced option in the context menu is available only in the Engineering Workplace and not in the Operator Workplace.

Executing the menu item Write I/O Allocation to Control Builder M configures the channel properties of the I/O boards in Control Structure beneath and including the selected object from the Signal Parameter aspect properties of the allocated Signal objects including I/O variable creation. You can switch this function to auto-update the hardware units during I/O Allocation.

I/O board hardware units affected by I/O Allocation are locked to prevent changes in Control Builder M Professional to keep consistency. The lock can be removed in Control Builder M Professional.

Existing allocations done in Control Builder M Professional can be read back with some limitations by executing the menu item Read I/O Allocation from Control Builder M.

I/O Allocations also supports allocation of HART field devices and variable creation for PROFIBUS field devices.

I/O Signal Groups

I/O Signal Allocation also supports signal grouping. A Signal Group is a specific signal object type and represents a couple of base I/O signal objects types (e.g. CBM_AIS, CBM_DOS). Their I/O variables are combined into a in variable of structured data type for Control Builder M.

Signal group object types are composite object types and shall be built in a library. They can be configured to be connected to corresponding variables on application level, on single control module level and, for object types, on control module level.

Further information on I/O Allocation is found in the *System 800xA Engineering Engineering Studio (3BDS011223*)*.

Topology Status Viewer

Diagrams that show the network topology of the system, the status of the devices and additional important information are a valuable help when it comes to problem tracing and maintenance.

With Topology Designer you can configure graphical network topology diagrams that include the hardware devices of the whole plant network, the client/server network and of the control Hardware Units of the Control Network in including live hardware status information of devices that expose this information.

An operator or a maintenance employee that inspects a completely configured graphical network topology diagram including status information on an Operator Workplace uses the Diagram view of a Topology aspect as Topology Status Viewer.

Loading the Topology Designer system extension equips the Hardware Units for AC 800M on CPU and I/O board level with Topology aspects which hold graphical symbols representing these hardware units. These symbols are available in the Component view. Further standard skeletons for Topology Diagrams (in Diagram view) on Controller and on Network level are available. You also can add Topology aspects to objects and object types as you require and configure the Component view specifically.

If you open the corresponding skeleton on the Topology aspect of the Controller object then you automatically get a Topology Diagram showing the Controller and its I/O (Figure 36).

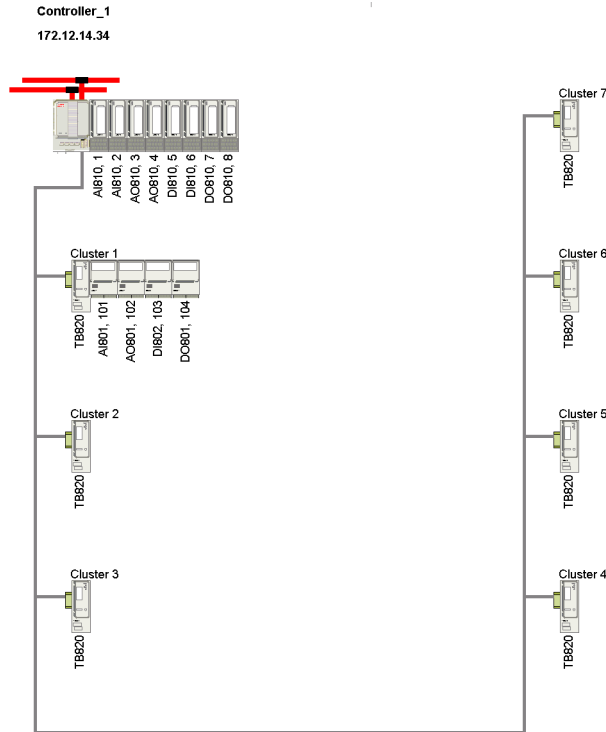


Figure 36. Topology Diagram on Controller Level

If you create a Topology aspect and open the corresponding skeleton on the Network object then you get a Topology Diagram populated with the Controllers of the Network. If you create a Topology aspect and open the corresponding skeleton on a higher level object the diagram contains, provided the objects of interest below are prepared with Topology components, additionally these symbols. You easily can finalize the diagram by arranging and connecting the controller symbols and other symbols.

From the Controller symbols you can call up the Controller Topology diagrams (Figure 37).

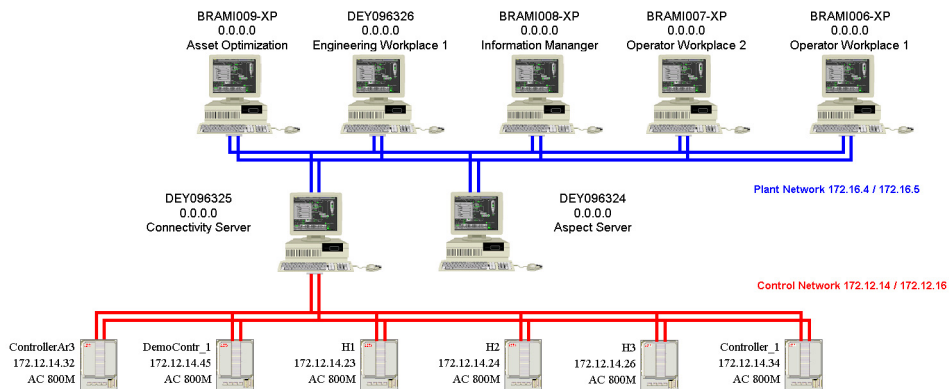


Figure 37. Topology Diagram on Plant Network Level

When the corresponding OPC server is operating and the controllers are running you see the live hardware status information of the Hardware Units displayed in the Topology Diagrams.

More information on Topology Designer is found in the *System 800xA Engineering Engineering Studio Topology Designer (3BDS011225*)*.

Process Graphics

With Process Graphics functionality in System 800xA you can

- Build graphic elements.
- Configure graphic displays.
- View graphics.
- Use faceplates for information and control.

You configure graphic aspects with the Graphics Builder engineering tool.

Graphics Builder

Graphics Builder is a tool that helps to configure graphic aspects such as graphic displays, graphic elements, and faceplate elements. The main parts of Graphics Builder are Edit Panel, Menu Bar, Tool bar, Toolbox window, Expression Editor, Properties window, Status Bar, and Element Explorer.

The following features are specific for Graphics Builder:

- **Properties Window** is used to configure the properties of graphic items and input items.
- **Expression Variables** are used to define and edit the expression variables.
- **Expression Editor** is used as an advanced alternative to configure property values. It contains a richer set of data selectors than available in the Properties Window. It also helps the user to add expressions containing references to aspect object properties, resources, expression variables, or input properties.
- **Toolbox** Window to display the graphic building blocks defined in the system.
- **Element Explorer** to select and insert graphic elements in the currently edited graphic aspect.
- **Input Properties** to define and edit user defined properties.
- **Solution Library** to define graphic entities as reusable solutions.
- **User Enumerations** to create user-defined data types.
- **Data References and Resource References** to view or change the data-entity references or resource references in the graphic aspect.

These windows are accessed through the menu bar, toolbar, context menus, and shortcut keys. [Figure 38](#) shows the main user interface characteristics of Graphics Builder. How to work with Graphics Builder is described in the *System 800xA Engineering Process Graphics (3BSE049230*)*.

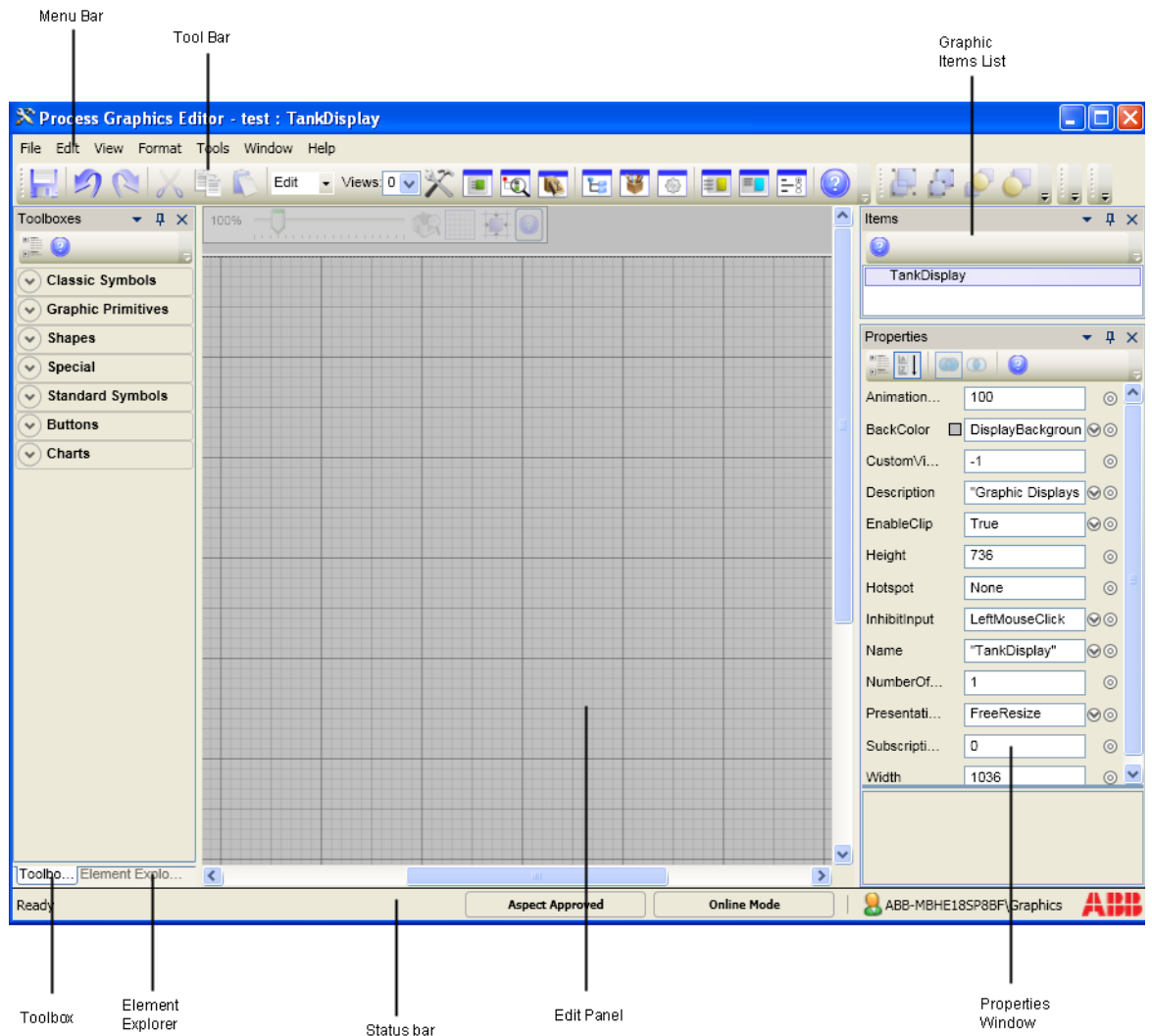


Figure 38. Graphics Builder

Graphic Aspects

The Process Graphics 2 aspect system provides four different aspect types.

- Graphic Display PG2, which is the graphic aspect used directly by process operators.
- Graphic Element PG2, which is an object aware building block.
- Generic Element PG2, which is a generic building block.
- Solution Library PG2, which is a graphic aspect that supports copying or pasting solutions for graphic displays in a Graphics Builder.

Aspects created from each aspect type are configured using the Graphics Builder. Graphic Displays are invoked directly by the process operator for monitoring and controlling the graphics during the run time.

Graphic Elements and Generic Elements are building blocks that are used while configuring graphic displays, other graphic elements, or faceplates.

Each aspect type is categorized into different aspect categories that define the usage of a graphic aspect

Graphic Aspects in Structures. Graphic aspects may be created on objects found in various structures. The structure used reflects the purpose for the graphic aspect.

Structures used and the corresponding purpose of the aspect is described in [Table 27](#).

Table 27. Relevance of Structures for Graphic Aspects

Structure	Purpose of Aspect
Object Type Structure	A graphic aspect placed on an object type object is inherited by all instances created from the object type. Built in aspect categories suitable for object type objects are Graphic Element, Faceplate Element and Object Display.
Graphics Structure	Aspects placed in Graphics Structure shall be generic. Refer to Graphic Aspects on page 247 for a description of generic elements. Graphics Structure is used to create toolboxes in the Graphics Builder. An object placed in Graphics Structure corresponds to a toolbox tab. Built in aspect categories suitable for Graphics Structure are Property Element, Subelement and Symbol Element.
Any other structure (typically Functional Structure)	Graphic aspects placed in other structures are typically aspects of the Graphic Display aspect category. Graphic aspects placed for example in the functional structure are sometimes called instance graphics.

Object Aware and Generic Graphic Aspects. A major characterization of graphic aspects is to divide them into one of the following subsets:

- Generic Graphic Aspects**
Generic graphic aspects are characterized by being applicable regardless of object type and even of the controller family connected to. Generic graphic aspects are therefore quite versatile creatures. Yet, they require more configuration, as every property that requires dynamic data must be connected individually.

Generic graphic aspects are entered, using Graphics Builder, by selecting them from a toolbox.

- **Object Aware Graphic Aspects**

In object aware graphic aspects there is in contrast to generic graphic aspects, support for object references. The main feature is that there is always an invocation object available in each instance of object aware aspects. The invocation object makes it possible to use relative references to aspect objects, aspects and aspect object properties within the graphic aspect.

For more information about graphic aspects dividing into different subsets, refer to [Graphic Aspects](#) on page 250.

The invocation object is set up differently for graphic displays and for instances of graphic elements:

- Graphic Element aspects have a property named ObjectPath. This allows object aware graphic element instances to be parameterized with a reference to an object. This parameterization takes place automatically when using the Element Browser.
- For graphic displays the invocation object is set up in runtime by the runtime system.

Built-in Graphic Aspect Categories. For each of the Graphic Display and Graphic Element aspect types, there are a number of built-in (predefined) aspect categories, see the two tables, [Table 28](#) and [Table 29](#) below. It is also possible for users of graphics to create custom aspect categories with templates.

By selecting a certain aspect category when you create a graphic aspect, you select a number of predefined quantities like:

- The default size of the graphic aspect.
- The background color of the graphic aspect.
- A set of properties available.

[Table 28](#) describes built-in aspect categories used to configure object aware graphic aspects.

ABB provides libraries of object aware graphic aspects as part of the connectivity for various controllers, while others may be built by you, or purchased from other sources in the form of graphic libraries.

Table 28. Built-in Aspect Categories Used for Object Aware Graphic Aspects

Aspect Category	Description
Graphic Display	The aspect category is intended for “instance graphics” i.e. graphic displays that are built and placed in a concrete structure, typically Functional Structure.
Object Display	The aspect category is intended to be used on object types to be general presentations for all objects of the type
Faceplate Element	The category of graphic aspects used by faceplates to perform monitoring and control of the process
Graphic Element	Aspects of the Graphic Element aspect category typically represents objects of a certain object type. For example a valve.

Table 29 describes built-in aspect categories to configure generic aspect aspects.

Table 29. Built-in Aspect Categories Used for Generic Graphic Aspects

Aspect Category	Description
Subelement	Aspects of the Subelement aspect category are generic aspects intended for presentation purposes, i.e. elements with properties to which data can be provided using expressions.
Property Element	Aspects of this category are intended to be used when configuring Faceplate elements. The usage is not limited to faceplate elements though.
Symbol Elements	The Symbol Element aspect category is prepared to create elements of “icon” type, shapes. The symbol element, by default, contains an Image Control to which a picture (icon, windows metafile, gif-file etc.) can be assigned.

ABB provides libraries of generic elements as standard building blocks.

Subsets of Elements. There are a number of terms that are used to define different subsets of elements used by and produced by Graphics. In some cases these terms are equivalent with names of aspect categories. This section tries to clarify these terms.

Factors that control if an element belongs to a certain subset are the following:

- Whether the element is object aware not.
- Implementation, i.e. whether implemented as a graphics aspect using the Graphics builder or if implemented in any other way.
- Usage - Whether the element is used directly by the process operator in the run time situation or if it is a building block.

[Table 30](#) defines a number of these subsets. The columns describes values of controlling factors. An “X” means that the factor is not relevant for the subset.

Table 30. Subsets of Graphics Elements

Subset	Object Aware	Implemented using Graphics Builder	Usage
Element	X	X	X
Graphic aspect	X	Yes	X
Graphic Display	Yes	Yes	By operator
Graphic Element	Yes	Yes	Building block
Generic graphic aspect	No	Yes	Building block
Generic Element	No	X	Building block
Object aware graphic aspect	Yes	Yes (So far always)	X

Generic graphic aspects constitutes, as can be seen from [Table 30](#), the subset of the graphic aspects that are implemented using the Graphic Builder. Other generic elements are Graphic Primitives, Extended Graphic Primitives, standard Visual Basic components and third party ActiveX controls.

Building Blocks Used in Graphic Aspects. Elements used in the graphic aspects are:

- Generic elements.
- Object aware graphic elements.

Generic elements can be used to implement the static part of the configured graphic aspect. Generic elements can also be used to present dynamic information by connecting expressions to their properties.

Object aware elements represent the other way to implement presentation of dynamic information in a graphic aspect. Object aware elements are usually a more convenient to use than generic elements as the only requirement is that a reference to an object is set up, in contrast to defining the data subscription using expressions. Generic elements are, on the other hand much more flexible.

Graphic aspects are described in detail in the *System 800xA Engineering Process Graphics (3BSE049230*)*.

Using Object Types with Graphic Aspects

The AC 800M Connect graphical object type library for example contains object types with object aware graphic elements.

After instantiating a PID controller object type in Functional Structure below a tank object you can use the corresponding graphic element in Graphics Builder by browsing to the element and placing it in the tank's graphic display.

After saving and building the display you have made the graphical elements dynamic representation available and you can call-up the faceplates of the PID controller by mouse clicks as shown in [Figure 39](#).

Display Services

Additionally you can use Information Management Display Services to build and distribute graphical process monitoring capabilities in your plant.

A Display Server as a collection of service and data providers for graphic display call-up, management, and data access handles an interface to several Display Clients. A data provider for the 800xA system exists. The Display Client provides a

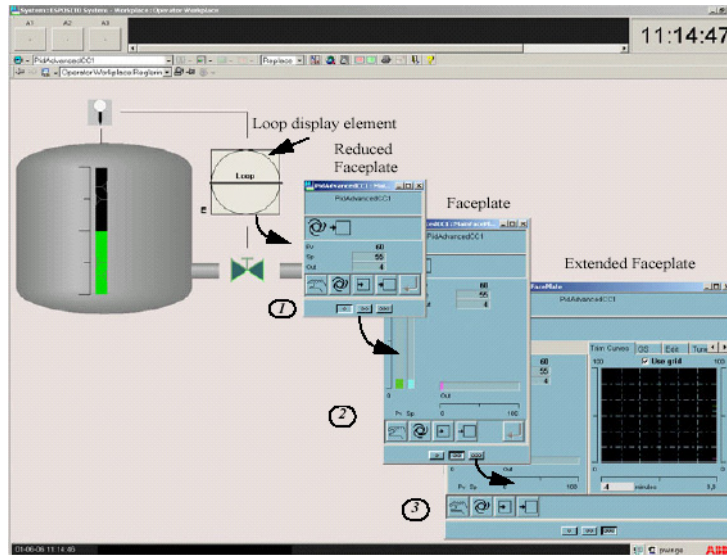


Figure 39. Tank Graphic Display: Regulator Valve with Connected PID Controller

graphical user interface for display building and viewing and for administration of Display Services.

Refer to *System 800xA Information Management Display Services (3BUF001093*)*.

Alarm and Event

Alarm and event management is supported in System 800xA.

While an event just informs the user that something has happened, an alarm is an event that alerts the user of an abnormal state and needs to be acknowledged.

Alarms are indications given to the operator by an audible sound, some form of visual indication, and/or with a message of some kind. The alarm indicates a problem and the purpose is to direct the operator's attention towards plant conditions, so he/she can correct potentially dangerous situations in time.

There are two types of alarms and events.

- Process Alarms and Process Events are generated from your process, such as failure in a valve or pump or an overpressure in a tank.
- System Alarms and System Events are generated from the 800xA System, for example network problems, file system errors, server errors or hardware errors.

System 800xA handles alarm and event information generated in controllers or other hardware units or applications and provides presentation, alarm acknowledgement, logging, archiving, transfer to SMS and e-mail, and management functions.

Figure 40 shows the conceptual alarm and event flow from a Controller to the Operator Workplace, to SMS and E-Mail devices and to Information Management.

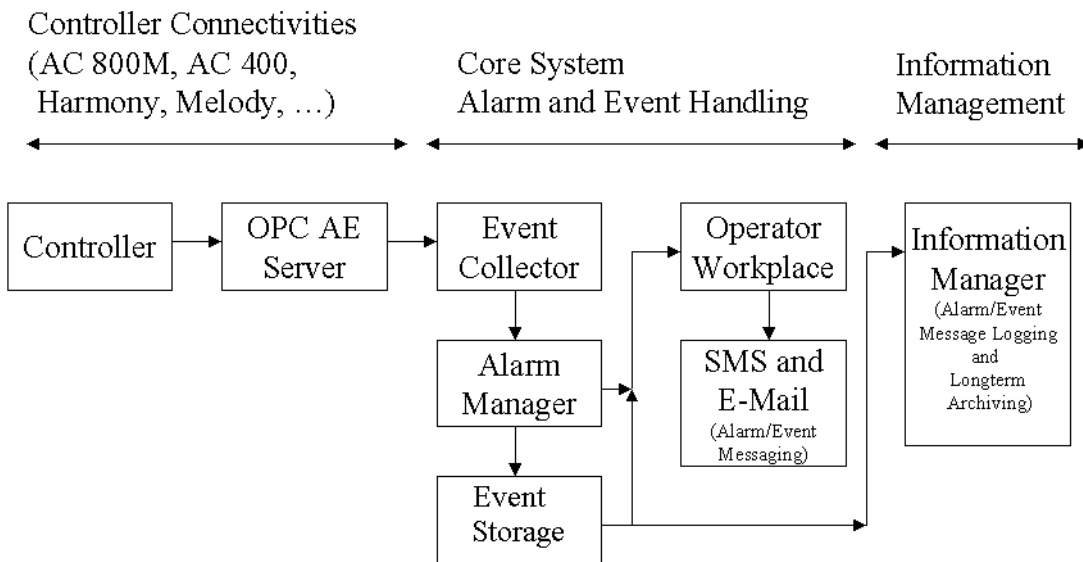


Figure 40. Conceptual Alarm and Event Flow

The following schematic figures show more details on alarm and event handling on Controller and on Server and Workplace level (Figure 41).

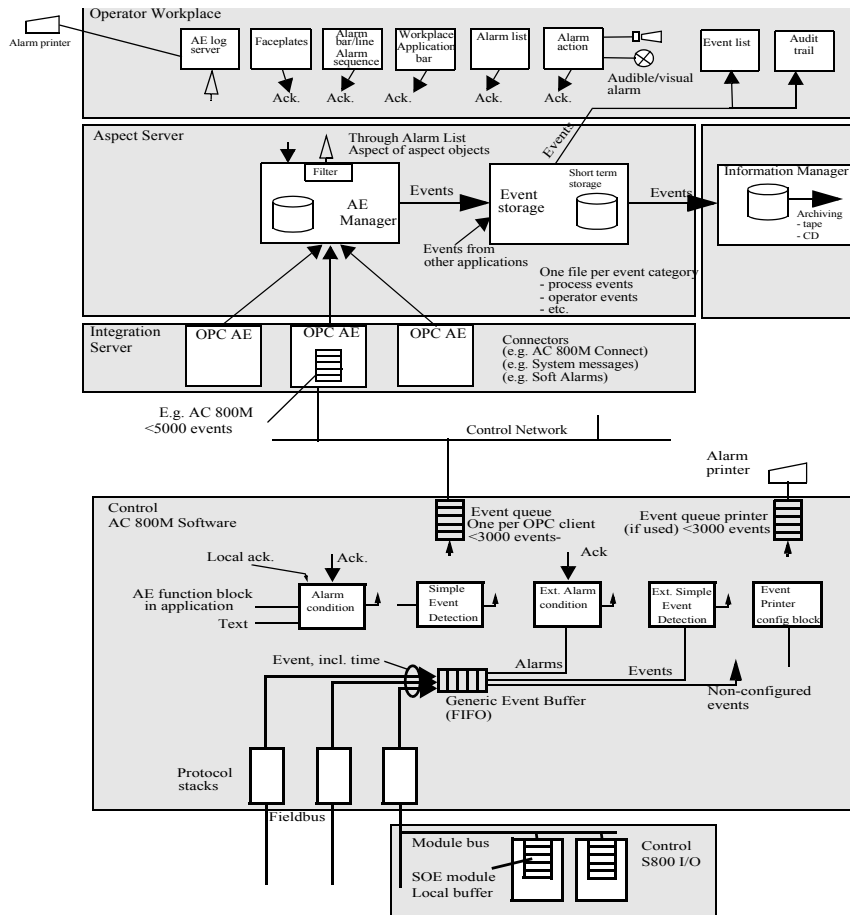


Figure 41. Alarm and Event Handling on Controller Level (AC 800M), Server and Workplace Level

Alarm Philosophy

An alarm is one of several sources that help the operator to diagnose a system. It helps the operator to:

- Correct potentially dangerous situations in time.

- Recognize and act to avoid hazardous situations.
- Better understand complex conditions.

Another function the alarm system has, it to provide a log, which can be viewed via an event list or alarm printout etc. The alarm log can be used for optimizing plant operation, for analyzing incidents, and for improving the performance of the system.

The alarm log can show:

- Alarms generated during a specific time-span.
- Alarms per unit or system with priority distribution.

It is recommended to define an alarm philosophy before setting up an alarm environment. It typically describes alarm generation and structuring principles, the role of the operator, acceptable alarm rates, alarm priorities and definitions, etc.

A helpful guide to do this is *Alarms Systems - A guide to design, management and procurement*, publication no. 191, ISBN 085931 076 0, published by *The Engineering Equipment and Materials Users Association* in 1999 and available at www.eemua.co.uk.

This document has the objective "to help engineers to develop alarm systems which are more usable and which result in safer and more cost-effective operation of industrial systems".

A main topic in this guide is the need for addressing the usability of the alarm environment. A common problem is alarm rate overload. This makes an alarm system very hard to use and is a safety risk. A few rules of thumb are listed below. Remember that these are approximations and that specific systems may differ.

Alarm systems should be designed to meet user needs and operate within the user's capabilities. This means that the information presented by the alarm system should:

- Be relevant to the user's role at the time.
- Indicate clearly what response is required.
- Be presented at a rate that the user can deal with.
- Be easy to understand.

The table below displays approximate figures for target maximum rates of occurrence of alarms of different priorities.

Table 31. Presentation Rate per Priority

Priority	Target Maximum Occurrence Rate
safety related/critical	very infrequently
high	less than 5 per shift
medium	less than 2 per hour
low	less than 10 per hour

Table 32. Acceptable Long Term Alarm Rates

Long Term Average Alarm Rate in Steady Operation	Acceptability
more than 1 per minute	Very likely to be unacceptable
one per 2 minutes	Likely to be over-demanding (industry average in HSE survey)
one per 5 minutes	Manageable
less than one per 10 minutes	Very likely to be acceptable

The Hiding functionality in 800xA can be helpful in reducing the amount of irrelevant alarms even after the alarm system was designed. It can easily be adjusted during system operation to accommodate new operational situations that generate unforeseen irrelevant alarms.

Disabling/inhibiting and hiding functions as shown by [Figure 42](#) are available that allow to configure provisions against alarm floods which would overstrain the operator.

Alarm/Event Overview

Alarm and Event functionality provides:

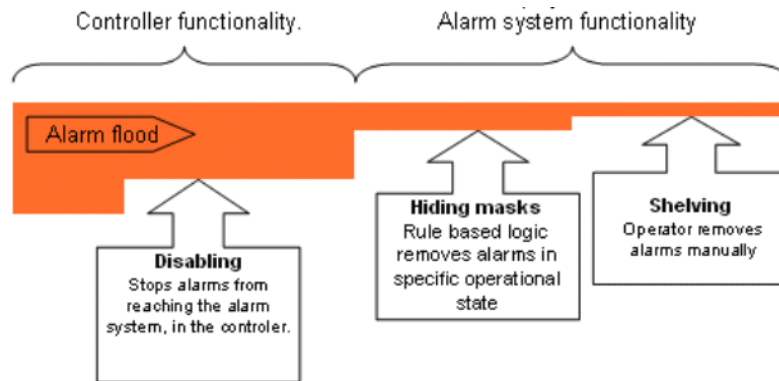


Figure 42. Reducing Alarm Flood

- Alarm lists or event lists are presented in a table type format.
- An alarm logger is supported for printer output.
- Alarm Bands provides a number of unacknowledged alarms in a summary display for selected alarm lists.
- The Sequence Bar is a status display, where the defined number of alarms is displayed horizontally.
- The Alarm and Event function can be seen as holding a time stamped queue of all alarm and event messages.
- Alarm and event list configurations can be shared between lists.
- Irrelevant Alarms can be removed from alarm lists by configuring Hiding Masks. Single alarms can be hidden as well as signals from an object and its descendants.

Seven services as shown in [Figure 43](#) provide the functionality of Alarm and Event.:

- The Event Collector Service is used to affiliate one or more connectivity Alarm and Event OPC Servers to the 800xA system.
- The Event Storage Service is used for storage and retrieval of events.

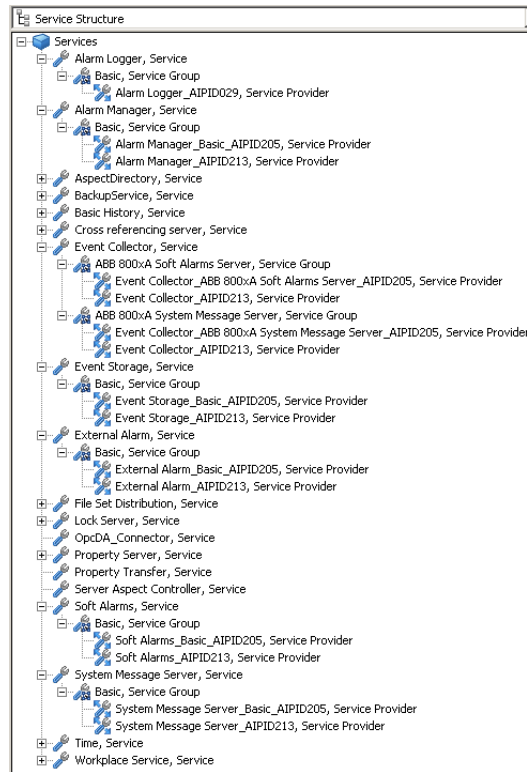


Figure 43. Alarm /Event Related Services in Service Structure

- The Alarm Manager Service provides the alarm state machine
- The Alarm Logger Service is a service which makes it possible to continuously log alarms and events and print them.
- The External Alarm Service makes it possible to enable external alarms.
- The Soft Alarms Service is used for the Alarm Expressions and for Programmer API.
- The System Message Server Service is used for system events and messages.

As an operator, you can view alarms according to the selected filter and presentation settings or you can easily make changes. For example, a list can be sorted by double

clicking on a head line, columns can be resized or they can be moved using drag and drop. Filtering and presentation functions let you view alarms in dedicated ways as well as monitor and maintain the plant in a more effective way. For example, a list can show all alarms or only those for the associated object and its descendants. A details page is also associated with each item in the alarms list.

Alarm acknowledgment or silencing of local audible sounds is fast and easy. You can also acknowledge an alarm when it occurs on an object within graphics.

Up to 1000 priorities can be defined with acknowledged and unacknowledged colors. An audible sound can be set to any file (in .wav format) and played once or until silencing. Date formats are selectable and time stamps can be generated (as from the controller). Customized set-ups can be used for other alarm lists.



In a configuration with redundant alarm servers, a server that is starting up will be updated from controllers that support this function.

You can access the alarm and event functionality as part of the configuration (Application Bar, Status Bar, display shortcuts) or through aspects selectable from a workplace. Alarm information can be presented using:

- Alarm List - view current alarm status for all objects or for a particular set of object and descendants (optional).
- Event List - view events.
- Alarm Acknowledge - you can acknowledge alarms from lists or from graphics displays or using the context menu Acknowledge verb.
- Alarm Band - view the number of Unacknowledged Alarms and use the button type link to the associated list view (commonly placed in the Application Bar).
- The Sequence Bar is a status display, where the defined number of alarms is displayed horizontally.
- Message List - view selected categories of messages in a list type format.
- Logging - Continuous printout of alarms and or events to a dedicated line printer.
- Acknowledge security - set up security to allow and deny groups or users to acknowledge alarms.
- External Alarm - activate and deactivate an external alarm notification.

Alarm Server and Storage. The Alarm Server holds the Alarm Database, storing the current status of alarms for the objects accessed by the server. In other words, each process object added to your system automatically gets alarm information stored.

Event Server and Storage. The Event Storage Server holds the Event Database storing all process events that occur as well as all system messages including audit trail messages. By default the system message server holds approximately 6000 messages in a “circular” file but the size can be changed.

Alarm Expressions. The Alarm Expression is a useful tool to get information in an Alarm and Event list based on property values. You can define an expression related to a property value, which gives you a predefined message line in the list. Configuration is done via an Alarm Expression aspect in an Aspect Object associated to the property.

If you have for example an OPC server without event handling, you easily can configure an alarm related to a special property value using the of the Alarm Expression aspect.

External Alarm. The External Alarm function provides aspect systems and components to make it possible to activate and deactivate an external alarm notification. The user of the external alarm selects an alarm list and digital inputs and outputs in the configuration view. The alarm list filter is used to subscribe for alarms and events from the alarm and event server. Setting the outputs and reading the inputs are done via Data Subscription. When active unacknowledged alarms, fulfilling the filtering criteria, are received by the external alarm function the Audible Alarm output is set. The operator can silence (resetting the Audible Alarm output) the external alarm through the silence view. Silence can be done for all configured alarms or for one explicit external alarm.

Alarm/Event User Interface Configuration. You can reconfigure System 800xA Alarm/Event handling by changing the default presentation layouts or create new layouts in the Library Structure. The default layouts are:

- Common Alarm Line (sets the filter for alarm type messages and the columns, that is appropriate, for three lines of alarms).

- Common Alarm List (sets the filter for alarm type messages and the columns, that is appropriate, for the 500 lines of alarms).
- Common Audit List (sets the filter and columns suitable for audit events generated from Security Definition aspects).
- Common Event List (sets the filter for event type messages and the columns, that is appropriate, for 500 lines of events).
- Common Logger Configuration (is used to configure the Logger function for printout of Alarm List).
- Common Operator Message Line (sets the filter for operator type messages and the columns, that is appropriate, for one line of events).
- Common Operator Message List (sets the filter for operator type messages and the columns, that is appropriate, for 500 lines of events).
- Common Process Alarm Line (sets the filter for process alarm type messages and the columns, that is appropriate, for three lines of alarms).
- Common Process Alarm List (Hidden Only) (sets the filter for hidden process alarm type messages and the columns, that is appropriate, for the 500 lines of alarms).
- Common Process Alarm List (sets the filter for process alarm type messages and the columns, that is appropriate, for the 500 lines of alarms).
- Common Process Event Line (sets the filter for process event type messages and the columns, that is appropriate, for one line of events).
- Common Process Event List (sets the filter for process event type messages and the columns, that is appropriate, for 500 lines of events).
- Common System Alarm Line (sets the filter for system alarm type messages and the columns, that is appropriate, for three lines of alarms).
- Common System Alarm List (sets the filter for system alarm type messages and the columns, that is appropriate, for the 500 lines of alarms).
- Common System Event Line (sets the filter for system event type messages and the columns, that is appropriate, for one line of events).
- Common System Event List (sets the filter for system event type messages and the columns, that is appropriate, for 500 lines of events).

Among the things you can configure for each layout are:

- Filtering.
- Included columns, their size and name.
- Setting of presentation details, such as alarm color, date and time format.
- Time range.
- Sorting of the list.

Alarm/Event Service Configuration. Some of the six system services (refer to [Figure 43](#)) for Alarm and Event found in the Service Structure need to be or can be configured. Refer to the *System 800xA Configuration (3BDS011222*)*.

The services Alarm Logger, System Message, External Alarm and Soft Alarms do not need configuration.

For the Alarm Manager service you can configure the Service Group Definition aspect using the Special Configuration tab regarding Alarm Handling, Event Logging and Alarm Storage items.

For the Event Collector service you have to configure the Service Group Definition aspect using the Special Configuration tab regarding OPC A&E Server and Collection Mapping (Collection Definition to use) items.

Alarm/Event Overview - Controller

If you use the AC 800M connectivity of the 800xA system, then the alarms and events presented at the Operator workplace are generated from the AC 800M controllers. The principle that the alarms and events are generated from the controller applies for other tightly integrated controller series too.

Alarms and events can be generated in the AC 800M controller in three ways:

- By using objects based on library object types containing alarm and event functions, for example objects derived from the object types of ProcessObjExtLib or SignalLib.
- By using objects especially made for alarm and event handling (based on object types in the AlarmEventLib)
- By hardware units throughout the system (system alarms and events).

The standard alarm functions delivered with the AC 800M controller product are packed in a library called AlarmEventLib.

The library comprises 10 alarm/event Function Block Types and two Control Module Types provided to be used for implementation of alarm and event handling in control applications when not using objects that already include it. For available types refer to [Figure 44](#).

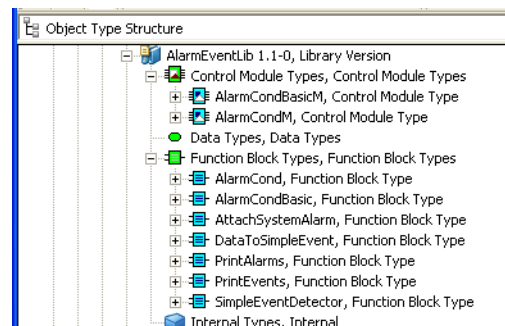


Figure 44. Object Types of AlarmEventLib

Alarm and event handling on controller level based on objects of the AlarmEventLib library are described in detail in the *System 800xA Control, AC 800M Configuration (3BSE035980*)*.

The built-in alarm and event handling in Signal-, Process- and Control loop - object types of other standard libraries is based on the same principles as implemented by AlarmEventLib and is described in the *System 800xA Control AC 800M Binary and Analog (3BSE035981*)*.

Inhibit and Disable Alarms and Events. Sometimes there is a need for temporarily suspending alarm and event generation, for example to reduce an alarm flood. This can be done for all objects with built-in alarm handling:

- **Disable** – the alarm condition is disabled, no alarms and events are generated, nothing is logged, and no control action is taken (that is, the system does not act upon the alarm condition).

- Inhibit – the control action itself is inhibited (that is, the system does not act upon this alarm or event), while alarms and events are still logged and presented to the operator in the operator interface.

Alarms and events can be disabled from the faceplate and from alarm list, as well as from the application, via interaction parameters.

Figure 45 illustrates the difference between inhibiting and disabling an alarm.

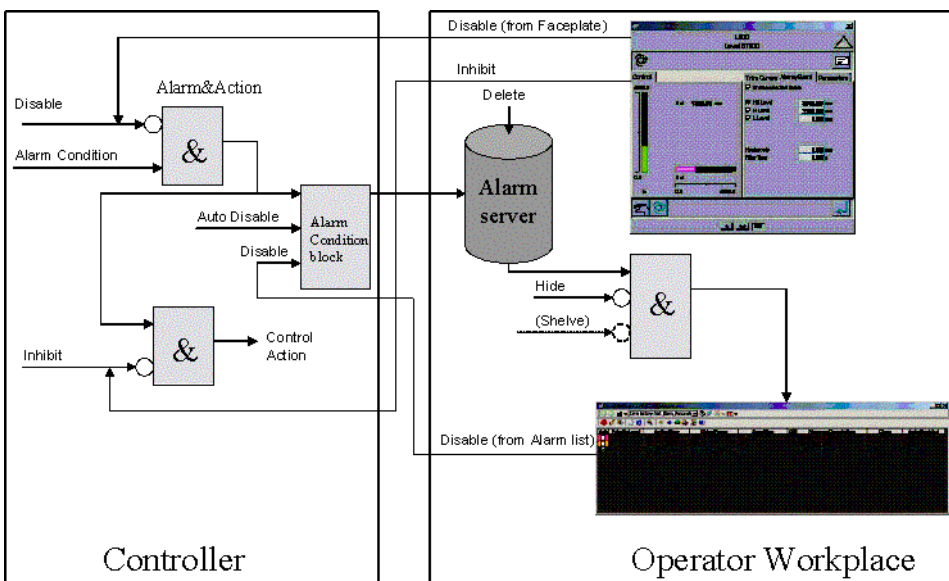


Figure 45. Disabling/Inhibiting Alarms

Normally, the control action will be a boolean signal that causes a certain reaction, for example, a signal that stops a motor. However, a control action could also cause a more complex series of actions. Inhibit is only available in certain the object types. In a SIL application, alarms cannot be enabled or disabled via MMS. However, alarms can be disabled or enabled from the IEC-61131-3 code.

Set Up Alarm and Event Handling OPC Server. Before alarms and events can be received, it is necessary to connect the OPC Server for AC 800M to the System 800xA Alarm and Event Server in the Connectivity Server node. You do this by

configuring the Event Collector Service Group in the Service Structure of Plant Explorer. Note that the Event Collector and the OPC Server must run in the same node.

Alarm and Event Buffer Configuration

The alarms from the process are stored in buffers in the controller(s) and then distributed to buffers in the OPC Server for further distribution to 800xA system (refer to [Figure 46](#)).

The figure illustrates a buffer configuration for a control system, where OPC clients have subscriptions on alarms and events from different OPC Servers. When a buffer is full, a System Simple Event is sent upward to 800xA system. All buffers are created in accordance with CPU and OPC Server settings.

For more details on System 800xA Alarm and Event configuration, refer to the *System 800xA Operations Operator Workplace Configuration (3BSE030322*)* and subsection on configuration of Alarms and Events in the *System 800xA Configuration (3BDS011222*)*.

Alarm/Event Overview - Message Logging

All alarm and event messages for the 800xA system, including Process, Operator, and Audit Trail messages, are collected and stored by the System 800xA Event Storage service.

This provides a short-term storage facility with the capacity to store up to 50,000 messages. The messages can be organized into filtered lists for viewing. This functionality is described in the *System 800xA Operations Operator Workplace Configuration (3BSE030322*)*.

If your system has Information Management History Services installed, the messages stored by System 800xA Event Storage service may be forwarded to a History Services message log for extended online storage. This message log can store up to 12 million messages. In addition, with History Services you can consolidate messages from multiple servers onto a dedicated consolidation node, and save the messages on an archive media for permanent offline storage ([Figure 47](#)).

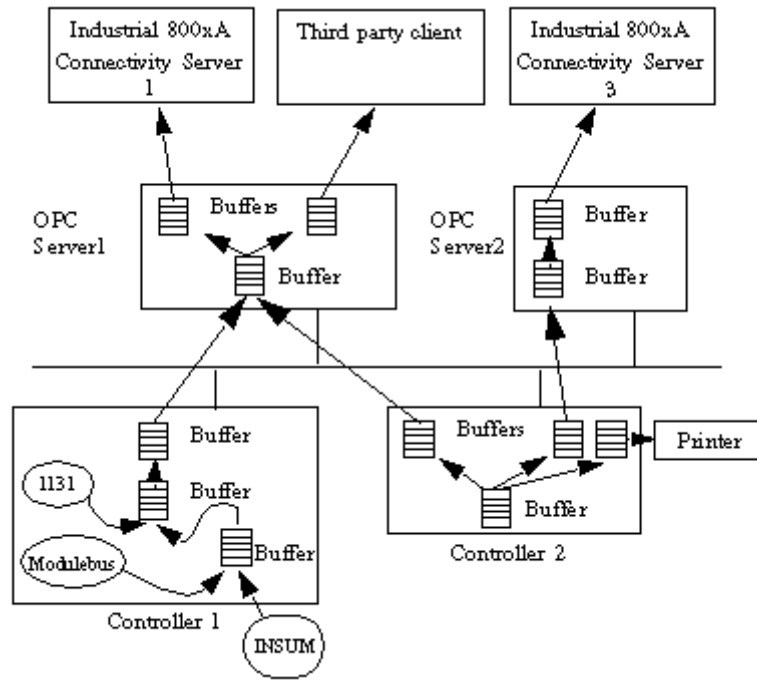


Figure 46. Example of a Buffer Configuration.
 (When a buffer is full, a System Simple Event is sent upward to Industrial^{IT} System 800xA)

History Services Message Log Overview

Message logs store events and system messages that are generated by control and operator applications. Three types of message logs are available, depending on where the events occur and what functionality is needed:

- Alarms and events which are buffered in the System 800xA Event Storage service may be forwarded to an **OPC_MESSAGE** log type. This type of storage is modeled after the data definitions in the OPC/Alarm and Event specification. This includes Audit Trail messages. The Audit Trail function tracks operator changes such as activating/deactivating historical logs, changes

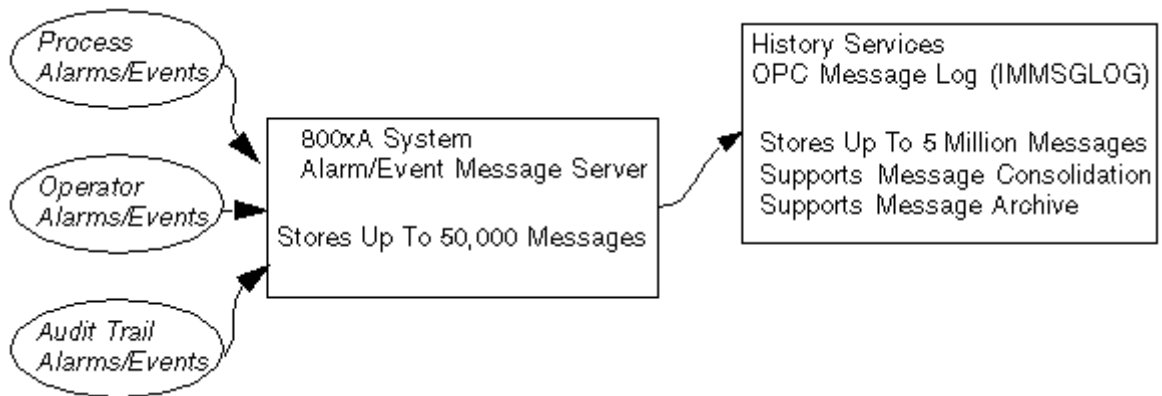


Figure 47. Alarm/Event Message Logging Overview

to production data, log configuration changes, and so on. The log stores the identification of the person that made the change, time that the change was made, previous and new value for the data being changed, as well as other information.

- The **PDLMSGLOG** log type is a special implementation of the **OPC_MESSAGE** log type for storing batch events related to the Batch Management application.
- The **DCS_MESSAGE** log type is provided to support consolidation of **Advant OCS messages** from earlier Enterprise Historian nodes (version 3.2/1 or earlier).

Accessing Message Log Data

The messages can be read via interactive dialogs in DataDirect and Desktop Trends. DataDirect also lets you specify re-executable functions for implementing reports. You can also access the data directly using SQL queries. For instructions and guidelines on using these tools, refer to the applicable section in the *System 800xA Information Management Configuration (3BUF001092*)*.

Guidelines for Configuring Alarm/Event Logging

The configuration and set-up requirements for message logs are listed and briefly described below. Detailed instructions are provided in the referenced documents.

- Integrate message-generating applications with 800xA system's System Message Server.
- Configure the 800xA system's System Message Server.
- Configure message logs.
- Configure consolidation for message logs.
- Configure archival and offline storage for message logs.
- Tune the interface for System 800xA and Information Management Message Services
- Configure Oracle access for message logs.

Integrating Message-Generating Applications. Certain message-generating applications in the 800xA system, including History Services, Softpoint Services, and AC 800M, write their messages to the 800xA system's System Message Server via application-specific OPC alarm servers. These application-specific OPC alarm servers must be added to the Alarm and Event Service Group for each connectivity server where the applications run. These Service Groups reside under the Alarm and Event Services category in Service Structure.

This configuration should have already been done as part of the Information Management post-installation. Guidelines for verifying this configuration (or performing the configuration if not already done) are provided in the *System 800xA Information Management Configuration (3BUF001092*)*.

Configuring Message Logs. The operating parameters for a message log are specified in a Message Log object. A dedicated object is required for each message log that you need to configure. Typically, you only require one message log for each message log type you are using (OPC, PDL, or DCS). These Message Log objects must be added under a specific node in Node Administration Structure. Instructions are provided in the *System 800xA Information Management Configuration (3BUF001092*)*.

Message Log Consolidation. Alarm/event messages from multiple server nodes may be consolidated onto a dedicated consolidation node. This functionality is configured using the Scheduling Services and IM Consolidation action plug-in. Instructions are provided in the *System 800xA Information Management Configuration (3BUF001092*)*.

Offline Storage. All events and messages stored in a message log can be copied to an offline storage media. How to configure this functionality is described in the *System 800xA Information Management Configuration (3BUF001092*)*.

Configuring Oracle Access for Message Logs. Message log data is stored in an Oracle database. Client applications such as DataDirect and Display Services access Oracle-based History data, including message logs, via an ADO data provider named DBA. The data provider must be configured and must reside on the same node where the message logs reside. This configuration is performed as a post-installation procedure. You can verify the ADO data provider configuration via the ADSS Config tool in the Windows Control Panel. For details regarding configuration requirements, refer to the Information Management section in the *System 800xA Post Installation (3BUA000156*)*.

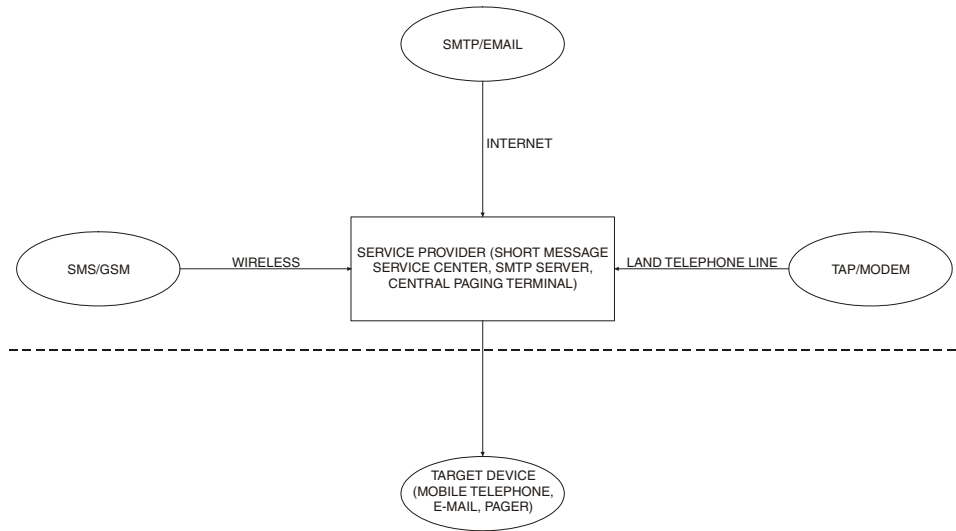
SMS and e-mail Messaging

SMS and e-mail Messaging is a system extension to System 800xA. It provides a method for sending messages based on alarm and event information to user devices such as mobile telephones and e-mail accounts. It is possible to control sending messages by configuring a message schedule for each user.

Figure 48 shows and Table 33 lists the three methods SMS and e-mail Messaging employs to notify users of alarm and event information. The table also lists the devices that are compatible with each notification method, and which devices, using the SMS/GSM notification method, allow the user to confirm receipt of the message back to the 800xA system.

The notification methods work as follows:

- SMS/GSM - uses Short Message Service (SMS), a service for sending short text messages to compatible devices, to send messages based on alarm and event information to the Global System for Mobile Communication (GSM) service provider's Short Messaging Service Center (SMSC) over a wireless



T05164A

Figure 48. Notification Methods

Table 33. Notification Methods

Device ¹	Notification Method		
	SMTP/E-mail	TAP/Modem	SMS/GSM
Numeric Pager	—	—	—
Alphanumeric Pager	Notify	Notify	Notify
2-Way Pager - Fixed Reply	Notify	Notify	Notify
2-Way Pager - User Entered Reply	Notify	Notify	Notify/Confirm Receipt
2-Way GSM Pager	Notify	Notify	Notify/Confirm Receipt
Text Messaging Enabled Telephone	Notify	Notify	Notify/Confirm Receipt
Wireless Equipped PDA	Notify	Notify	Notify/Confirm Receipt

Table 33. Notification Methods (Continued)

Device ¹	Notification Method		
	SMTP/E-mail	TAP/Modem	SMS/GSM
E-mail	Notify		Notify/Confirm Receipt

NOTE:

1. This table lists the capabilities of SMS and e-mail Messaging. The selected hardware and/or service provider may impose other limiting factors.

network. The SMSC sends the message to compatible devices of users configured to receive them. This method allows users of the compatible devices to confirm receipt of the message.

- SMTP/e-mail - uses Simple Mail Transfer Protocol (SMTP), a protocol for sending e-mail messages between SMTP servers, to send messages based on alarm and event information to an SMTP server over the Internet. The SMTP server sends the message to e-mail accounts, or to compatible devices via e-mail accounts, of users configured to receive them.
- TAP/Modem - uses Telocator Alphanumeric Protocol (TAP), a protocol for submitting requests to a pager service, to send messages based on alarm and event information to the pager service provider's Central Paging Terminal over a land telephone line. The Central Paging Terminal sends the message to compatible devices of users configured to receive them.

A block diagram of SMS and e-mail Messaging is shown in [Figure 49](#).

The figure shows the data flow from the time the Alarm and Event List is notified of an event. SMS and e-mail Messaging consists of several components. When an alarm is generated, it is sent to the Alarm and Event List. An SMS and e-mail Messaging component, the Alarm and Event Message Source, monitors the Alarm and Event List associated with it. When it senses an event it formats a message (based on configured values) and sends the message to the Message Handler. The Message Handler sends the message to destinations (individual users or user groups) subscribed via the Message Subscriber.

The Message Schedule determines which users receive messages based on their schedule. If it determines a user is scheduled to receive a message via SMS/GSM, SMTP/e-mail, or TAP/Modem, the message is sent to that user.

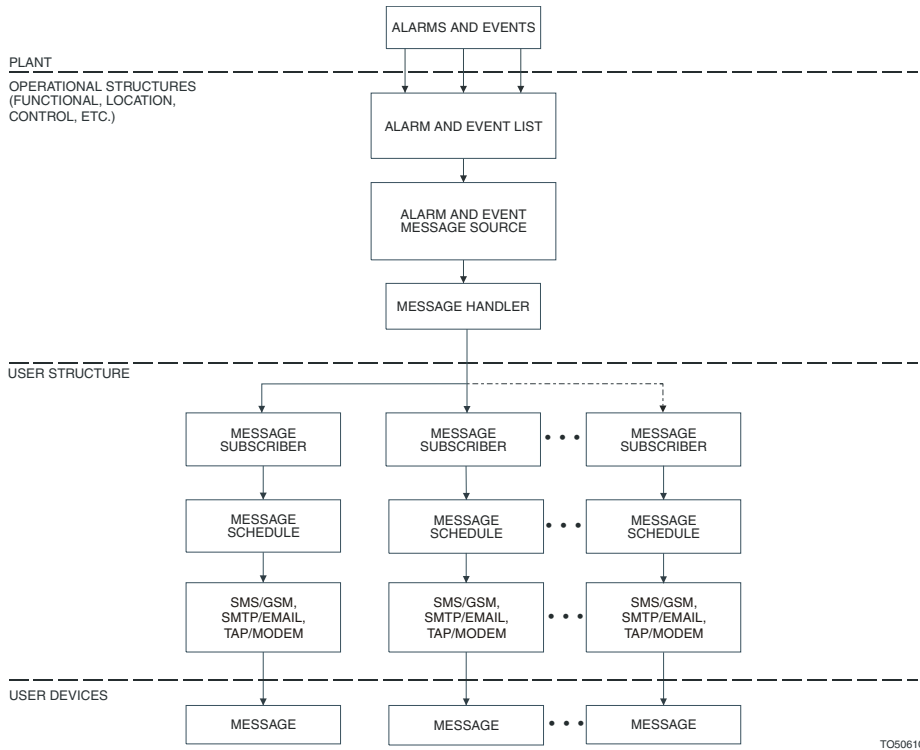


Figure 49. Block Diagram SMS and e-mail Messaging

Import/Export

During engineering of a project you often need to transfer your engineering work or parts of it:

- To another engineering system (for example from an off-site engineering system to an on-site engineering system, refer to [Distributed and Multi-user Engineering](#) on page 57).
- To a test system.
- To a production system.

You may also need to save limited project specific application data before introducing changes or upgrades.

These engineering tasks are supported by the Import/Export tool (Figure 50).

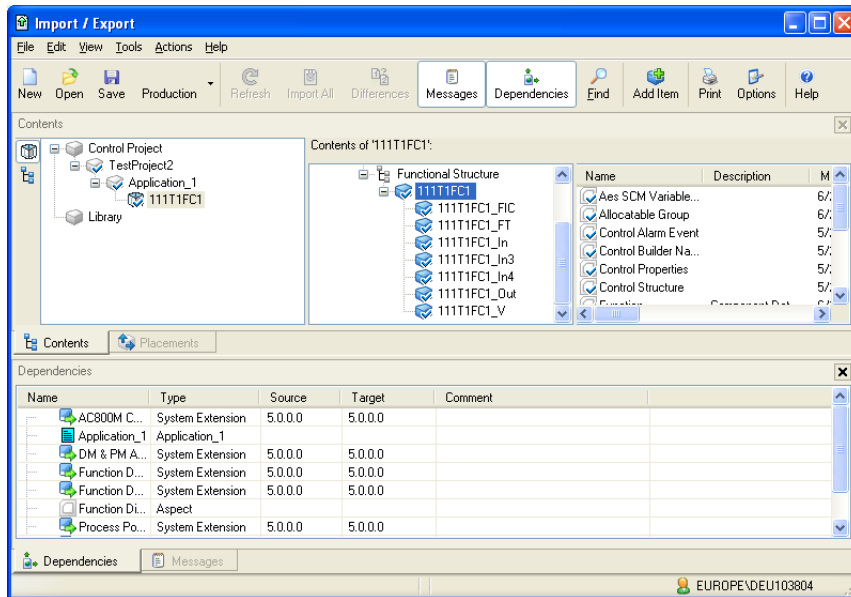


Figure 50. Import / Export Tool

The Import/Export tool allows you to import data to and export data from any 800xA system.

Exporting objects, aspects, entities and entity extensions is done by drag and drop from the Plant Explorer to the Contents tab of the Import/Export tool or by pushing the **Add Items** button, selecting the items to export from a Select Item structure browser.

The data is saved in archive files, structured files with file extension .afw.

Importing objects, aspects, entities and entity extensions is done using a menu item **Import All** for all the contents of the archive file or **Import** for a selected part of it. Before doing import it is also possible to view the contents of an archive file, detect and show differences between two archives and between an archive and the system.

In the Contents tab and in the Select Item dialog the tool provides two basic views: Overview (Entity View) and Structure View. The Overview view shows the dropped in contents as hierarchy of entities, - entities displayed as containers -, the Structure View shows this contents as structures and objects similar as Plant Explorer.

The Placements tab shows the placements of the contents items in structures and can be used before import to define new placements, for example if a parent object is not available.

In the Dependencies tab the dependencies of a selected item are displayed during export. You can decide to include the dependencies into the exported archive if required.

In the Messages tab messages are displayed during export and import.

The Import / Export tool should be used to transfer smaller applications in the form of entities, entity extensions, object types and/or object instances, and aspects between systems. Dependencies of these items are shown and can be exported and then imported together with these items.

The full functionality and use of the Import / Export tool is described in the *System 800xA Maintenance (3BSE046784*)*.

For system backup and restore actions, use the Backup / Restore tool described in the *System 800xA Maintenance (3BSE046784*)*.

Entities

The Import / Export function available in Plant Explorer is entity and entity extension aware.

Entity aware export means that all objects and aspects contained in an entity are brought in automatically when you select the entity object or one of the contained objects or aspects and are exported to the same archive file.

If a selected entity has dependencies, these are shown in a list but are not automatically exported. If you want to include the objects/aspects the entity depends on, into the export archive, you have to add them explicitly from this list.

Entity aware import means that the imported entity replaces an already existing corresponding entity in the system completely, especially objects and aspects no longer contained in the entity are deleted in the system.

Entities are normally exported and imported as units, but it is also possible to export parts of entities for special purposes.

Dependencies



The Import/Export function shows, by default, the dependencies. The dependencies are not a part of the archive file. If that is required than they must be included manually.

Dependencies for an item selected in the Overview or Structure View are shown in the Dependencies tab. A selected dependency or all dependencies can be added to the contents to be exported.

This allows you to export and import complete applications, for example a control application and user defined libraries and insertions of the objects into other structures than Control Structure, using one file.

The exported contents might be dependent on a specific version of System 800xA, a System Extension or a standard library (for example AC 800M Connect libraries) brought into the system by a system extension. For these dependencies it is necessary to resolve them by installing and loading them - before continuing the import.

A notification of missing dependencies is shown in the Import / Export user interface (if a system is running) as well as when trying to import.

Further special dependencies not shown can exist and have to be resolved by providing them before import, for example:

- Windows Application and Web Page aspects can reference files. You have to take care of referenced files yourself: Store a copy of these files with the archive file set and describe where to store and how to install them in the import system so that the corresponding imported aspects can find them.
- ActiveX Wrapper aspects reference the GUID of an installed and registered ActiveX control. You have to take care of transferring the ActiveX control, installing and registering it in the import system yourself.
- MS Excel or MS Word files of Document Manager and of File Viewer aspects can require macros. You also have to take care of them yourself.

- Property references (as supported by Document Manager, Parameter Manager, Function Designer, and Topology Designer) specify dependencies which are not exported. You have to export/import the corresponding objects additionally if not already covered.

Resolving Errors

While importing, there might be situations where you have to make a decision for the import to be able to continue.

One common situation is that the parent of the object that's being imported, does not exist in the system. You need to resolve this by defining which object should become parent object in the Placements tab or by providing the parent object with the same name.

Another example is that you import an aspect that already exists. You need to resolve this by choosing whether to replace that aspect or skip it and continue with the rest of the import. It is possible to let the tool make the same decision every time a similar incident occurs.

Upgrading Files

Import/Export can read old files, but not create. This allows you to import afw-files of previous system versions.

Filter Exports

There are two filtering options:

Either you use the find tool to export the items you want. (You use the find tool as a filter.)

Or you drag the items you want to export into the Import/Export tool and delete the items you do not want to export in the Import/Export tool before you press **Save**.

Differences

With the Differences function you can check the contents regarding objects, aspects, entities and entity extensions of an archive against the contents of another archive or the contents of the system and environment to decide which has the contents you

want to work on. Or, if you like to import only the new parts you can detect these in the archive and then import only these parts.

Detailed Difference Report

Difference report provides the differences between Function Diagrams in the system and Diagrams present in the afw file to be imported. If there are any modifications between the Function Diagrams, the difference report provides a report for the Function aspect indicating the additions, deletions, or modifications between the Function Diagrams. For more details, refer to the *System 800xA Maintenance (3BSE046784*)* manual.

Audit Trail

The Import / Export tool supports Audit Trail. All major operations performed are published in the Audit Trail log. Information about when the operation occurred and which user performed it is also visible here.

Recommendations for Archive Files

Archives are afw-files that you can store somewhere in a directory on a disk of your computer or on a disk in the network. To increase thread mitigation the archive files shall be stored on protected folders. A descriptive file name, an informative description in the file properties and a documentation how to handle the file during import shall always be available and updated. As Import/Export will not perform well regarding speed and memory consumption if an afw file is too big, a large export should be divided into several files, e.g. one file per library and one file per Control Builder project.

Configuration Using Spread Sheets

Engineering data describing an automated plant under design and configuration often is collected in Microsoft Excel spreadsheets. These informations define the design of the plant.

It is most favorable to use these bulk engineering data directly to create and parameterize all the needed objects forming the plant model in Plant Explorer as there can be needed hundreds or thousands of objects. To create and parameterize

these objects manually in Plant Explorer object by object would consume too much time and is error-prone.

This basic requirement and many more requirements regarding the efficient handling of bulk engineering data are supported by the Bulk Data Management functionality included in System 800xA.

Bulk Data Management bases on Microsoft® Excel, a set of Excel Add-Ins that integrate Microsoft Excel into a 800xA system and set of predefined templates and applications. Bulk Data Manager connects the full productivity benefits of Microsoft Excel worksheets to System 800xA configuration activities. Connections to relational databases can be implemented using the corresponding external data source functionality of Excel. The possibility to reimport lists of objects and aspects and to track changes in Bulk Data Manager supports an iterative analysis and design process. When different persons or group of persons (customers, external contractors) are involved in the engineering process Bulk Data Manager supports you to organize a controlled design data list exchange.

The basic functionality of Bulk Data Management allows you to configure a worksheet for reading out and writing back the properties of aspects of objects.

You configure the worksheet by dropping in the aspects from Plant Explorer and selecting the aspect properties of your interest. The data contents of this worksheet you can acquire by dropping in the objects of interest from Plant Explorer. Each row holds the data for one object. The properties of the corresponding aspects of the dropped in objects are shown in the configured columns.

This worksheet also allows you to create new objects from object types or from already existing object instances and to create the placement of the objects in structures. Refer to [Figure 51](#).

To duplicate a substructure of objects you just copy the area of rows corresponding to the objects of your interest in the worksheet using well-known Excel functionality, you edit the names of the new objects and the data of other properties and save the changed spreadsheet contents to the system by executing a Save All Objects command.

Once you have configured a worksheet you can use it for offline editing or for importing and mapping engineering data from other sources (refer to [List Based Data Exchange](#) on page 289) without having attached it to the system. After

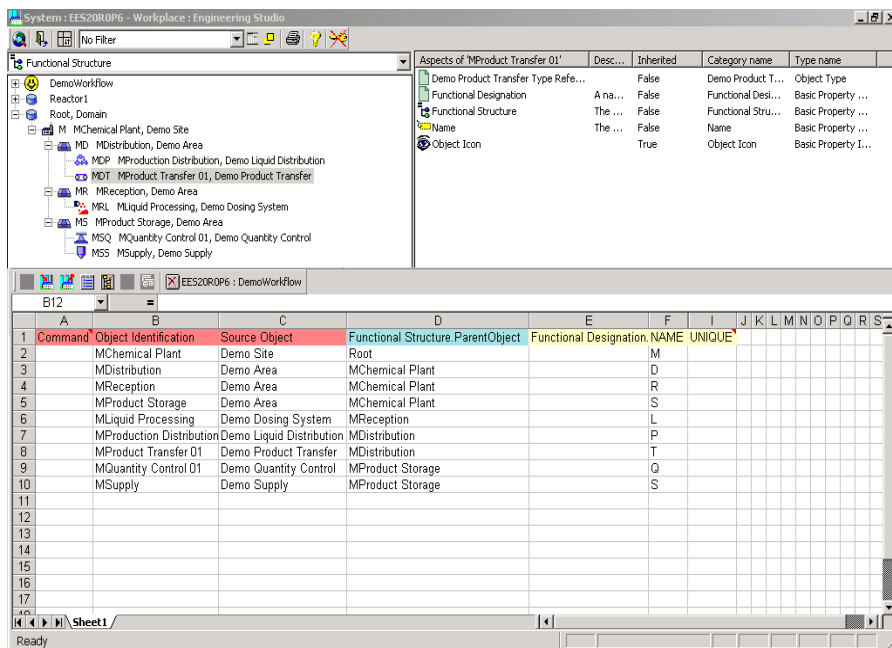


Figure 51. Building Basic Functional Structure with Bulk Data Manager

finishing offline work you can reattach the worksheet to the system and work online again.

Typical examples for worksheets that are edited offline (for example by a contractor) and then used online to create and/or parameterize objects are Signal Object Lists, Process Object Lists, I/O-Board Lists.

You can choose to include a Bulk Data Management worksheet into an object as a Document aspect. Then the worksheet is automatically attached to the system when you open the worksheet.

Data Area

In a worksheet you can configure either one Default Data Area (this we assumed in the short description above) or several Auto-Update Data Areas. Each data area has a configuration headline that defines which properties of which aspect category are

to be handled in the data area. You can configure the headline by dragging and dropping in aspects of the categories of interest from Plant Explorer to the worksheet. During drop of an aspect you select also the properties of interest, the properties that then define the columns in the configuration headline.

Data of a Default Data Area is loaded by drag and drop of objects of the attached system from Plant Explorer. Data of a Auto-Update Data Area is loaded automatically from the attached system when opening the worksheet or if you execute an Update command on the whole worksheet or the selected data area.

An Auto-Update Area is especially useful if you want to monitor changes or do changes in the system repeatedly.

Object Identification

Bulk Data Management can identify objects and their locations in structures by names, paths, Absolute Reference Designations or object ids (GUIDs). You have to choose a corresponding option to define the identification method fitting for your case. For saving data the objects must be identified unambiguously.

Filter

For every data area you can configure a filter criterion that is executed during the loading of data. You can for example filter for all objects in Functional Structure having the object type MotorUniM and a certain value of a property of the Control Properties aspect.

Track Changes

The Track Changes function allows you 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 worksheets. This function helps you to check for changes and introduce changes to your system in a controlled way if you exchange your worksheet with other engineers, for example with a contractor.

Property References

Additionally to configuring Data Areas to read / write object (aspect) properties Bulk Data Management supports insertion of property references into cells of your worksheet. A Property Reference is a dynamic link to a property of an aspect of any object. It allows to access the data provided by the property it is pointing to. Property references are a means to configure a view on different properties of different aspects (of different aspect systems) for data access.

In general a Property Reference is specified by a character string with a defined syntax describing the path to locate the property. As this string can be complex a dialog to all engineering tools supporting property references is provided that allows to specify the path by navigating through the structures, objects, aspects and aspect properties.

In Bulk Data Management this dialog is provided by context menu on a cell: Property Reference >Insert...>Create Property Reference.

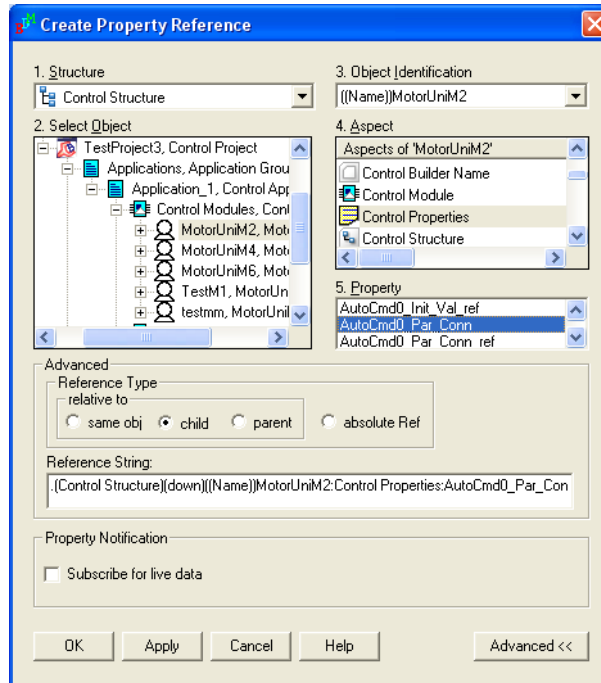


Figure 52. Create Property Reference

A Property Reference is automatically updated when the workbook is opened, this means the property value is read into the workbook cell carrying the reference. When a workbook cell carrying a Property Reference is changed, the modification is written back to the source aspect property where the reference points to.

Two kinds of Property References exist: Absolute References and Relative References.

Absolute References are independent of the context or the object to which the workbook was attached as Document Aspect. They point to the same object when the object the Workbook is related to is copied.

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. Using Relative References has the advantage of making a template independent of an object.

Property References can be configured 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 you, for example, to create animated bar graphs in your worksheet.

Structured Properties

Structured properties have a substructure of structured or simple sub-properties. Examples are given by the configuration data exposed by the Log Configuration aspect, by control aspects of Control Builder M Professional or by user defined structured properties of Parameter Manager aspects.

Bulk Data Management supports reading and writing structured property data exposed via System 800xA bulk data interfaces. The latter bulk data interfaces are supported in the examples mentioned above. A structured property is mapped to a grid of cells in an Excel worksheet as shown in Figure 53. This partly resolves the structure but gives the possibility to perform bulk data operations.

Property	Aspect	DataType	LogAspect	LogName	EngineeringUnits	Display
P1	General Pr	long	OPC Single Log	OPC 5s 5h	lit	
P2	General Pr	long	OPC Hierarchical Log	OPC 6s 6h Log_5m_1w_TAvg Log_1h_1y_TAvg	kg	Interp
P3	General Pr	long	OPC Hierarchical Log	OPC 6s 6h Log_5m_1w_TAvg	lit	

Figure 53. Structured Properties Mapped to Cells

After configuring the configuration headline of a worksheet by dropping in an aspect exposing a structured property and selecting the components of the structured property to include into the headline you can read in the data in a next step by

dropping in the corresponding objects and then change the data values and save them back to the object.

Formatted Templates

Combining Auto-Update Data Areas, Filters and Property references you can configure sophisticated formatted templates. Formatted templates are Microsoft Excel based spreadsheet applications that allow you 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 the system. Examples of templates are budget planning, resource calculations, price calculations, or even advanced applications like TagSheets, SignalLists, LoopDiagrams, dimensioning of motors or valves. Bulk Data Management provides some formatted template examples for you:

- LoopDiagram.xls, a template for documenting Control Loops
- TagSheet.xls, a template for documenting Tags
- and others

Most of formatted template examples are available as Document Management templates.

Preconfigured Bulk Data Applications

Combining the Bulk Data Management functions with standard Excel functions and VBA script programs that access standard Aspect Object Programming Interfaces, and/or specific tool interfaces (for example Open Interface of Control Builder M Professional) you can build your own powerful Bulk Data Applications.

Some preconfigured Bulk Data Applications based on this implementation principle are included in Bulk Data Management:

- BDM_for_Function_Diagrams.xls, an application to support Function Designer with bulk operations on Function Diagrams: Copy, Allocate and Generate Configuration Data.



The Bulk Data Manager does not support the copying or moving of Function Block Diagrams between projects.

- DocumentBatchPrint.xls, an application to support Document Manager with batch print of documents represented by Document aspects.
- Label_Report.xls, an application to print out labels for the I/O modules defined in a Control Builder M control project.
- CBM_CrossReferences.xls, an application to produce a cross reference lists of your control applications.
- CBM_IO_Variables.xls, an application to create, delete and edit Variables for your control applications.
- CBM_Applications_old.xls, an application to create, delete and edit Control Modules and Function Blocks of control applications (Still available for compatibility reasons).
- CBM_FunctionBlockDiagram.xls, an application to create, delete and edit (change) Function Blocks for Function Block Diagram, including positioning on Code Block and page of Code Block.
- ObjectTypeDefinition.xls, application to create object types.

Another set of preconfigured Bulk Data Applications is available that bases on the the System 800xA Bulk Data Interfaces for structured properties supported by some aspect systems (Log Configuration, Control Builder M Professional, Parameter Manager) and by Bulk Data Management:

- LogConfig.xls: Supports configuration of Log Configuration aspects.
- TrendConfig.xls: Supports Trend configuration.

The following applications support Control Builder M Professional:

- CBM_Applications.xls: This template is divided into the following sections supporting various operations on Application objects:
 - Global Variables/Variables: Global variables/variables can be created, deleted or properties like name, type or initial value can be modified.
 - Structured Variables Initial Values: Initial values of variables which are of structured data type can be set or modified.
- CBM_Programs.xls: This template is divided into the following sections supporting various operations on Program objects:

- Task Connection: Set, modify or delete the connection of the program to a task.
 - Variables: Create, modify or delete program variables including their properties like Name, Data Type, etc.
 - Function Blocks: Create, modify or delete Function Block declarations within the program.
 - Code Blocks: Add or delete code blocks of type Structured Text. Moreover the code of these code blocks can be edited in the related cell. This allows you for example to build Function Block calls in an MS Excel sheet and copy it into the cell ST_Code.
 - Structured Variables Initial Values: Initial values of structured variables can be set or modified.
- CBM_FunctionBlockTypes.xls: This template is divided into the following sections supporting various operations on Function Block Type objects:
 - Description: Add or change the description of the Function Block Type.
 - Parameters: Add, modify or delete parameter declarations of the Function Block Type including. all properties like Name, Data Type, etc.
 - Variables: Add, modify or delete local variable declarations of the Function Block Type including. all properties like Name, Data Type, etc.
 - External Variables: Add, modify or delete external variable declarations of the Function Block Type including. all properties like Name, Data Type, etc.
 - Function Blocks: Add, modify or delete Function Block declarations within the Function Block Type including. all properties like Name, Type, etc.
 - Control Modules: Add, modify or delete Control Module declarations within the Function Block Type including. all properties like Name, Type, etc.
 - Code Blocks: Add or delete code blocks of type Structured Text. Moreover the code of these code blocks can be edited in the related cell.

- **CBM_ControlModuleTypes.xls:** This template is divided into the following sections supporting various operations on a Control Module Type objects:
 - **CM Parameters:** Add, modify or delete CM Parameters including. their properties like name, type, etc.
 - **Variables:** Add, modify or delete Variables including. their properties like name, type, etc.
 - **External Variables:** Add, modify or delete External Variables including. their properties like name, type, etc.
- **CBM_FunctionBlocks.xls:** This template is divided into the following sections supporting various operations on a Function Block objects:
 - **Task Connection:** Enter or modify the connection of the Function Block to a Task.
 - **Initial Values:** Initial values of variables can be set or modified.
- **CBM_ControlModules.xls:** This template is divided into the following sections supporting various operations on a Control Module object as stated below
 - **Task Connection:** Enter or modify the connection of the Control Module to a Task.
 - **CM Connections:** Set or modify the actual parameter value for a CM connection.
 - **Initial Values:** Initial values variables can be set or modified.
- **CBM_SingleControlModule.xls:** This template is divided into the following sections supporting various operations on Single Control Module objects:
 - **Description:** Set or modify the description of the Single Control Module.
 - **CM Parameters:** Add, modify or delete CM Parameters including. their properties like name, type or initial value.
 - **Variables:** Add, modify or delete Variables including. their properties like name, type or initial value.
 - **External Variables:** Add, modify or delete External Variables including their properties like name, type or initial value.

- Function Blocks: Add, modify or delete Function Block declarations within the Single Control Module including all properties like Name, Type, etc.
 - Control Modules: Add, modify or delete Control Module declarations within the Single Control Module including all properties like Name, Type, etc.
 - Code Blocks: Add or delete code blocks of type Structured Text. Moreover the code of these code blocks can be edited in the related cell.
 - CM Connections: Set or modify the actual parameter value for a CM connection of the Single Control Module.
 - Structured Variables Initial Values: Initial values of structured variables can be set or modified.
- CBM_HWUnits.xls: This template is divided into the following sections supporting various operations on Hardware Unit objects:
 - Parameter Settings: The value of HW-unit specific properties can be set or modified. Remark: New properties can neither be added nor deleted. This is ignored during save.
 - HW Channels: Properties like Min, Max, Unit, (Connection-) Variable, etc. can be set or modified. Remark: New properties can neither be added nor deleted. This is ignored during save.

These templates can be modified. To change the structure of the sheet you have to reconfigure the headline via a dialog that allows to include the components of the structured properties into the configuration headline. There are some rows and columns hidden, which must be un-hidden first before the headline can be configured.

The preconfigured Bulk Data Applications and corresponding documentation files are available in the folder Engineering Templates on the desktop of your Engineering Workplace.

Hints on when and how to use Bulk Data Management are included in corresponding subsections of this manual. Bulk Data Management is further described in the section Bulk Data Manager of the *System 800xA Engineering Engineering Studio (3BDS011223*)*.

List Based Data Exchange

Engineering data describing an automated plant under design and configuration are often collected by third party engineering tools in other spreadsheets than Microsoft Excel or relational databases. The Excel functionality of Bulk Data Management can be used to import this engineering data lists to an 800xA system and to export data so that the third party engineering tools can import them.

Export Data

Data stored in the system 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). These formats can be imported by corresponding other engineering tools.

Import Data

Data can be imported into Microsoft Excel from a variety of data sources (Menu item **File > Open or Data > Get External Data**).

The imported data must be mapped to the configuration 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 item **Bulk Data Manager > Save**).

Extended Configuration Data

During engineering and operation of an automated plant in many cases storage and access to additional informations of different kinds is required. Examples of such informations are parameters for solutions, product and physical device informations or process or laboratory informations originating from foreign systems or products.

System 800xA gives you various possibilities to integrate these informations into the plant model of your system for easy access via Plant Explorer and from Aspect Systems.

Parameter Management

With Parameter Management you can flexibly create your own aspect categories to hold engineering and configuration information and complementary object or aspect information for operation. Parameter aspects provide these data for data sharing via property references, AESPropertyTransfer aspects or programmatic use and for access using Bulk Data Management.



Note that Bulk Data Management stores data in Excel files (.xls) only, while Parameter Management uses a relational database for storage of data.

Parameter aspect property data and the parameter aspect category definitions are stored in a Microsoft SQL Server 2000 Desktop Engine data base. If you need to extend existing aspect categories these extensions are propagated to the existing instances.

You can define parameter aspect categories of several kinds:

- Table-like (containing non-structured properties).
- Structured (containing structured properties).
- Extendable (allows to introduce instance-specific properties).
- Customized (mixture of structured and extendable)

For the properties you can configure default values, pick lists, validation rules, units and number of digits. Expressions and property references can be used to provide the property value.

Data types supported are: Bit, datetime, decimal, float, integer, nchar, nvarchar, smallint, uniqueidentifier.

Besides support for aspect category definition Parameter Management gives you support for entering, modifying and storing dedicated data into properties of aspect instances of these categories. You can view and enter data of one Parameter aspect in the aspect pan. To view and edit parameter aspects and their properties in bulk, Parameter Management interfaces via the Open Properties context menu item to a preconfigured Bulk Data Management Excel worksheet. Here you can configure and apply filters to filter out corresponding sets of Parameter aspects. In this way it is also possible to use Bulk Data Management to configure and produce reports of Parameter aspect data.

Of course you can configure your specific Bulk Data Management worksheets to import and enter data to Parameter aspects.

Changes to Parameter aspect properties can be audit trailed with old and new value.

Parameter aspect properties data are available for all System 800xA clients such as Graphics, Trends and OPC clients. This means that you can present them in Graphics, in Trends and in Reports.

Via property 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.

To transfer the property values of a Parameter aspect to the properties of another aspect in an object automatically, you can configure a corresponding AESPropertyTransfer aspect.

For example, you can configure a Control Module object type that gets its instance specific initial values and variable connections from the contents of a corresponding Parameter aspect. The preconfigured aspect category examples ArticleData, DriveSpecification and TagData allow you to explore working with Parameter aspects and corresponding document templates DeviceList, DriveSpecification and TagSheet. They show how you can include physical device information into the system.

Parameter Management has been combined with Document Management to one system extension due to the fact that both tools share the same Microsoft SQL Server 2000 Desktop Engine data base, flexible category definition functionality and common user interface parts.

Basic configuration steps for Parameter Management are found in the *System 800xA Configuration (3BDS011222*)*. Parameter Management is further described in the *System 800xA Engineering Engineering Studio (3BDS011223*)*.

General Properties

Aspects of the General Properties aspect category can also be used to store engineering and configuration data, but the focus is on complementary object or aspect informations for operation. The main differences to the Parameter Management aspect categories are:

- No category support, only templates. If you need to extend existing aspect categories these extensions are not propagated to the existing instances.
- The data is not stored in a database but in the aspect directory.
- Property configuration and property value entry support is more limited.
- Support of structured data does not reach the functional level of Parameter Management aspects and their Bulk Data Management based user interface.

The General Properties aspect is useful for storage of data of any kind such as integer, real, boolean and string formats. Current main usage of General Properties is to store Industrial IT product certification data.

General Properties are available for all System 800xA clients such as Graphics, Trends and OPC clients. This means that you can present them in Graphics, in Trends and in Reports.

General Properties can be configured using Bulk Data Management Excel worksheets.

Softpoint Services

SoftPoint Services lets you configure and use internal process variables not connected to an external physical process signal. Once configured, the softpoints may be accessed by other information management applications as if they were actual process points. For example, softpoint values may be stored in property logs in History Services. Reporting packages such as Crystal Reports may access softpoints for presentation in reports. Also desktop tools such as Desktop Trends and DataDirect can read from and write to softpoints.

Softpoint alarms can be configured and are directly integrated with system alarms and events. Engineering unit definitions for softpoints include minimum/maximum limits and a unit descriptor.

The Softpoint Services software may run on any number of servers within an Aspect System. Each Softpoint Server can have up to 2500 softpoint objects. Each softpoint object can have up to 100 signals; however, the total number of signals cannot exceed 25,000.

Data types supported are: Boolean, integer (32bit), single precision floating point (32 bit) and string. Also, double precision floating point (64 bit) is supported as an extended data type.

For details on Softpoint services, refer to the *System 800xA Information Management Configuration (3BUF001092*)*.

Calculations

Calculations Services lets you configure and schedule calculations for aspect objects. Typically calculations are applied to real-time database objects, including both softpoints and actual process points. Calculations may also be applied to object types. This lets you reuse the calculation each time a new object is instantiated from the object type.

Calculations can be triggered by changes to the inputs, or be scheduled to execute cyclically or at a given date and time. A calculation aspect may be applied to any aspect object such as a unit, vessel, pump, or softpoint. Inputs can be any aspect object property, and outputs can be any writable point in the system. Input/output definitions can be made relative to the object for which the calculation is defined. Data quality and alarm generation are supported. Calculation logic is written in VBScript.

For details on Calculations services, refer to the *System 800xA Information Management Configuration (3BUF001092*)*.

Documents

During engineering and operation of an automated plant many documents are required to be available and accessible. Examples of such documents are functional descriptions, specifications, data sheets, tag sheets and memos. System 800xA gives you various possibilities to integrate these documents into the plant model of your system for easy access via Plant Explorer.

Document Management

With Document Management you can configure your Microsoft Word and Excel documents and your AutoDesk AutoCad drawings to be dynamic documents, containing up to date information of Aspect Object properties. Other file based

documents (for example .PDF files of Acrobat Reader) and Web pages or html documents are supported as non-dynamic documents.

The documents can have meta data (administrative data like title, author, status, creation date, etc.) which are stored in a relational database based on Microsoft SQL Server 2000 Desktop Engine. This functionality supports you in managing large amounts of documents in an automation plant.

Document Management provides the Aspect Categories Document and AutoCad_Drawing. It further provides a collection of document file and folder template examples in the Library Structure. The top object of this collection is named Document Manager Templates.

Document file templates are applicable to create new documents, document folder templates allow to include a set of document files into one aspect with one file to be displayed by default. You can extend the given collection of templates by creating your own templates.

You can introduce a document file for an aspect of category Document or AutoCad_Drawing by creating it new from a template file, copying it from an existing document file or by referencing an existing document file.

You can also configure a Document Aspect to display a referenced Web page using a URL. Using the Document folder template feature you can configure a Document aspect to display the contents of a Web folder, for example a Microsoft Powerpoint slide set saved as .htm/.html file.

Dynamic Documents

To make a document dynamic you have to insert property references pointing to the properties of interest. The property references are handled bi-directional: If you open or refresh the document the actual values of these properties are read and shown in the document. If you change the values in the document (provided you have the permission to do so and the document is not read only) the new values are written to the properties.

For Microsoft Excel documents (including the Bulk Data Management functions) additionally you can configure property references that subscribe for live property data (for example process values) and auto-update data areas. The latter functionality provides tabular areas that read out directly properties covered by the area's configuration headline and filters if you open the worksheet or refresh the

worksheet. If you change the values in the data area and save them they are written to the corresponding properties.

Document Versions

Document Management handles a checked in version of a document and a checked out version. If a document is checked in you can only view it. If you check out the document you can change it, and no other user can change it. Other users can view the checked in version or the checked out version.

Import of Document Files

You can easily import Document files initially stored on a directory in the file system to their corresponding aspects using Bulk Data Management. Copies of the documents are checked in during import. You can use the same import worksheet to create the objects and the Document aspects.

For a reimport of the document files you have to delete the Document aspects first. You can do this easily in bulk with Document Management. After executing Open Properties on a Document Aspect you can execute the Show All or Show Subtree command to get a list of the Document aspects. In this list you can do multiple selection and apply the Delete Aspect command on the selection.

Basic configuration steps for Document Management are found in the *System 800xA Configuration (3BDS011222*)*. Document Management is further described in the *System 800xA Engineering Engineering Studio (3BDS011223*)*.

Information Access

Additionally System 800xA provides different aspect categories typically referenced as Information Access aspects. You can use the Information Access aspects to integrate applications and documents.

File Viewer

One possibility to integrate document files into your system is the File Viewer aspect. You can use the File Viewer aspect to integrate non-dynamic document files without database-administrated document properties into your system. Documents inside a File Viewer aspect can be signed and are read-only when being viewed. It is

possible to have a Work document and a Public document for the same aspect. Only the Public document can be accessed by users of the operator category.

The Work document can be the source document of the Public document, or a document that you want to store in 800xA. Work and Public documents can be stored in different formats. For example, the Work document can be a Microsoft Word document, and the Public document the corresponding Adobe Acrobat pdf-document.

Another function is the ability to create and use templates. A collection of File Viewer aspect categories with preconfigured templates needed for example for product documentation is provided by the system.

See how to use the File Viewer in the *System 800xA Configuration (3BDS011222*)*.

Bookmark

In addition to the File Viewer aspect category and its template categories a Bookmark aspect category is available. The Bookmark aspect is a viewer of file viewer and web page aspects. It has the ability to go to a specific page if the document viewed. See how to use the Bookmark Aspect in the *System 800xA Configuration (3BDS011222*)*.

File Viewer and Bookmark aspects can be created using Bulk Data Management but the aspect properties cannot be configured using Bulk Data Management. Therefore the files related to File Viewer and Bookmark aspects cannot be imported using lists.

Web Page

You can use the Web Page aspect to access any Web page or html document. It utilizes the WebBrowser ActiveX component from Microsoft. You configure the Web Page aspect towards a web page or documents located on a disk, which in turn may contain URLs linked to other pages on the Web. When you click on such an URL, the referenced Web page will be displayed. This is the normal way of web browsing, and the only thing you need to do is to configure the Web Page aspect.

The Web Page aspect can be created and configured using Bulk Data Management. See how to use the Web Page Aspect in the *System 800xA Configuration (3BDS011222*)*.

ActiveX Wrapper

The ActiveX Wrapper aspect provides you with a method to use an ActiveX Control as an aspect. When you have configured the ActiveX Control, you simply select that aspect to execute it. See how to use the ActiveX Wrapper Aspect in the *System 800xA Configuration (3BDS011222*)*.

ActiveX Wrapper aspects can be created using Bulk Data Management but the properties cannot be configured using Bulk Data Management.

Windows Application

The Windows Application aspect makes it possible for you to launch any executable installed on your Windows system. There are a number of useful Windows applications, like office applications, administrative tools for your plant, etc. that may be useful to access as aspects.

Windows application aspects can be created using Bulk Data Management but the properties cannot be configured using Bulk Data Management. See how to use the Windows Application Aspect in the *System 800xA Configuration (3BDS011222*)*.

Help Aspect

The Help aspect is used to provide online-help for an object type or an object instance. You can create files containing online help information, using various third party tools. You can also refer to information in the comprehensive set of online help files.

Advantages for Object Type

If you create online help aspects and place them on the object type, that help can function as help for that specific object type. If you choose to use inheritance for the help, you can even change the help later on and get an automatic update of the object instances help.

Advantages for Object Instance

You can create individual instructions and help for the respective object. For complex application objects created in a plant this can be very useful. Tuning

instructions, maintenance instructions or help for emergency situations can be part of the help.

See how to use the Help Aspect in the *System 800xA Configuration (3BDS011222*)*.

AUTOCAD Drawings

The CAD Drawing aspect allows to integrate AUTOCAD file into the system. See how to add and configure a CAD Drawing Aspect in the *System 800xA Configuration (3BDS011222*)*.

Section 16 Preparing for Operation in an FDA Regulated Industry

Requirements

Overview

The US Food and Drug Administration (FDA) issued 21 CFR Part 11 in response to the pharmaceutical industry's request to utilize paperless record systems under the current good manufacturing practice (cGMP) regulations in parts 210 and 211 (21 CFR parts 210 and 211). Part 11 went into effect on August 20, 1997. The regulation does not require a manufacturer to maintain records electronically. However it does provide the criteria under which the FDA will consider electronic records to be equivalent to paper records.

Many manufacturers have continued to maintain and compile records used to satisfy the cGMP regulations in a manual, paper-based format. However, these manufacturers do, in fact, have systems that produce electronic records. ABB, for example, has hundreds of control systems installed in such manufacturing environments that are using paper batch records. These control systems collect and store production data on operator consoles or in a historian environment. If the manufacturer is using these stored records for reference in a batch production record (e.g., trend printout, alarm/event list, etc.) or in a product investigation, the manufacturer has an electronic record. These systems do not need to comply with the regulation sections associated with electronic signatures, but they must comply with all the other sections of the regulation. Section 11.3 of the regulation defines electronic records as "any combination of text, graphics, data, audio, pictorial, or other information representation in digital form that is created, modified, maintained, archived, retrieved, or distributed by a computer system."

The support of compliance to 21 CFR Part 11 is an absolute, non-negotiable requirement for automation products sold into manufacturing environments subject

to FDA regulation. This is primarily a concern for manufacturers in the life science industry, but can also include food, beverage, and cosmetics manufacturers as well. Also some chemical and other manufactures who supply materials to the life science industry are required to comply with the regulation.

The requirements for 800xA systems to enable compliance have been categorized in the following table. Several requirements identified in 21 CFR Part 11 require the system owner to comply by having appropriate Standard Operating Procedures (SOPs) in place. Not all of the required SOPs are included with System 800xA product, offering, however ABB engineering services for validation can provide assistance in creating the appropriate documentation on a project basis. The primary sections from Part 11 are listed below.

Subpart B - Electronic Records

- Sec 11.10 - Controls for closed systems.
- Sec 11.30 - Controls for open systems.
- Sec 11.50 - Signature manifestation.
- Sec 11.70 - Signature/record linking.

Subpart C - Electronic Signatures

- Sec 11.100 - General requirements.
- Sec 11.200 - Electronic signature components and controls.
- Sec 11.300 - Controls for identification codes/passwords.

Table 34. References

Feature Category	Section References from 21 CFR Part 11 Regulation
Authorization	Aspect Operation (function) access checks, re-authentication at time of action SubPart B, Sec 11.10: (g)
Access Control	System and Function Access (i.e. logon, set point change) SubPart B, Sec 11.10: (d)

Table 34. References (Continued)

Feature Category	Section References from 21 CFR Part 11 Regulation
Electronic Signature	<p>User identify verification and tracking during operation on a record</p> <p>SubPart B, Sec 11.50: (a)</p> <p>SubPart B, Sec 11.70</p> <p>SubPart C, Sec 11.100: (a)</p> <p>SubPart C, Sec 11.200: (a), (l), (i), (ii), (3)</p> <p>SubPart C, Sec 11.300: (a), (b), (d)</p>
Versioning	<p>“Ability to discern invalid or altered records...”</p> <p>SubPart B, Sec 11.10: (a), (e)</p>
Audit trail	<p>“Use of secure, computer-generated, time-stamped audit trails to independently record the date and time of operator entries and actions...”</p> <p>SubPart B, Sec 11.10: (a), (e)</p> <p>SubPart B, Sec 11.50: (a), (b)</p>
Archive	<p>“The ability to generate accurate and complete copies of record in both human readable and electronic form suitable for inspection, review and copying by the agency.”</p> <p>“Protection of records to enable their accurate and ready retrieval throughout the records retention period.”</p> <p>SubPart B, Sec 11.10: (b), (c)</p> <p>SubPart B, Sec 11.50: (b)</p>

Table 34. References (Continued)

Feature Category	Section References from 21 CFR Part 11 Regulation
System Checks	“Use of operational system checks to enforce permitted sequencing of steps and events, as appropriate.” “Use of device (e.g. terminal) checks to determine, as appropriate, the validity of the source of data input or operational instruction.” SubPart B, Sec 11.10: (f), (h)

Production Records

Production records are required to prove the manufactured production lots were made with in-specification material and product specifications were measured and verified to be within expected tolerances.

National Language

The majority, if not all, countries require operations personnel to operate plant control systems in their native language. This is also a good practice to ensure critical information provided by the system is not interpreted incorrectly during urgent situations.

Training

Control system operations and engineering personnel require proper training for their role. Training should include standard system training as provided by ABB and specific operations training related to your process and your custom process control solution. ABB engineering services are available to assist with the generation of solution specific training materials and also to assist you in providing the proper training classes.

You will also need to keep appropriate records of training classes such that any point in time you can demonstrate that plant operations personnel are properly training for their role.

SOPs for 21 CFR Part 11 and cGMP Compliance

There are many aspects of compliance which require you to create and maintain standard operation procedures.

Considerations within System 800xA to Meet Requirements

Authorization

The first step in planning for your 800xA system is to determine who will be authorized to access the process control system. You should develop a procedure which establishes clear roles and responsibilities in order to ensure accountability during operation of your plant.

Re-authentication can be optionally used for critical operations such as writes to the control system, batch operations, and configuration changes in order to ensure that only authorized persons can take actions in the 800xA System. This option forces the user to resupply his/her credentials before the operations is executed. A double authentication may also be optionally used. In this case an additional person who has the respective secondary authentication authority has to give user name and password in order to approve the operation.

Access Control

Access to the 800xA System is controlled by the core system's security function and Windows security.

In order to support users having access to every workstation in the entire 800xA system with a single user name and password, you must implement Windows Domain server(s).

Define user roles for access to your automation system in order to have clear responsibility and accountability. Typical examples are: process operators, process supervisors, process engineers, system administrator. 800xA User Groups support the grouping of users based on the access they require or function they perform.

It is highly recommended that User Groups be implemented. User Groups provide an efficient method for user management (e.g. addition of new employee with one

entry, removal of employee from system access when they transfer to another job with one action, etc.)

Digital Signature

Digital signatures are supported as a Digital signature for all aspects of objects. A digital signature is generated and linked to an aspect. User verification via electronic method is performed by using Windows user id and password in combination with a selected reason for signature and an optional comment.

Consider which aspects will require digital signatures. Digital signatures are enforced by System 800xA in some cases (e.g. Report templates, Recipe approvals). For other cases, you may need to develop an SOP to ensure signatures are applied and checked at the appropriate times.

System 800xA supports customizable Reason codes which are required by all Digital Signature dialogs within the system. Consider which Reason codes are required to ensure easy tracking and analysis of the audit trail history. You should keep the number of reasons to the minimum number required in order to ensure that operations selection from the pick list is efficient and that you can correlate audited actions in logical grouping for your plant.

Versioning and Life Cycle Management

Some configuration data, such as Batch Recipe Procedures, are file based and present the need for versions which represent the state of the data at a user defined point in time.

Batch Recipe Procedures support major and minor version number increments. You should establish a procedure providing guidance to your operations staff regarding which type of changes you will need to consider major vs. minor for your plant.

Audit Trail

The Security and Access Control System allows audit of operator actions and security. The system supports logging of security violations, configuration changes, and operator actions to the process. The audit logs can be viewed in the alarm and event list. This makes it possible to see the effect of an operation. The audit log contains the following information:

- Date and time for the operation.
- Node from which the operation was performed.
- User name of the individual performing the operation.
- Type of operation.
- Object, property or aspect affected by the operation.
- Additional information from the involved aspect system.

The audit log is protected against modifications if the Aspect Servers are configured with access restrictions via Windows security. The security and access control system in the 800xA system allows audit of process control-specific activities as a complement to the audit logging available in the Windows system.

Archive

Archiving is essential to ensure retrieval during the records retention period.

System and Device Checks

System checks include sequence enforcement such as ensuring that PID loop must be in Manual mode prior to the user changing controller output. Some of these types of checks are covered by the 800xA System product directly while others must be configured as part of the user specific application engineering.

System health monitoring is another form of system checks. The Service structure provides an overview display including status of all 800xA system services including alarming capability for failed system components. User access to the 800xA system optionally ensures that the user has the access rights as defined for the specific workstation which the user is logged on and actively working.

Other connected devices such as weigh scales, PLCs, etc. provide data related to specific process equipment. The user specific application must ensure that the correct data source is configured and the user's application must ensure the health of the device prior to using data provided by the device. The system provides data quality for each object property in order to enable this type of check.

Production Records

Report templates created with electronic signatures. Generated production reports reviewed and electronically signed then archived. Plan for sufficient disk space in History server.

Generated reports awaiting electronic signature consume memory and computing resources in the Aspect server, plan procedure to minimize time from report generation to approval and archiving.

Training

Contact ABB University for consultation on developing a training plan based on the specific configuration of your 800xA system.

Plan an application specific training course for each unique role as defined above in the Access Control section above.

SOPs for 21 CFR Part 11 and cGMP Compliance

1. Configuration Management Procedure.
2. Password Change Procedure.
3. Change Authorization Procedure.
4. Problem Resolution and Tracking Procedure.
5. Application Specific Training Procedures.

Appendix A Aspect Objects and Structures

Aspect Objects in Engineering

The Aspect Objects handled in the engineering process for System 800xA represent physical objects (for example controllers, I/O boards) or virtual objects (for example functions, process objects), and are used as containers for engineering and runtime data.

Most of these objects can be instances of preconfigured, ready to use object types from libraries or object type groups provided with System 800xA. For example libraries and object type groups provided with AC 800M Connect. Further, configuring these objects is supported by preconfigured, ready to use aspect categories provided with the aspect systems of System 800xA.

The engineering tools support a concept of plant modelling by using Aspect Objects and structures. This concept is outlined below by describing example categories of objects from functional and control perspective. For more refer to [Basic Structures on page 315](#) and [Appendix B, Structured Plant Modeling](#).

Objects

Aspect Objects can, for example, be grouped into groups according to their function in the plant in the Functional Structure:

- **Plant Section Objects**
Consist of a group of units that each contains a set of process objects for each of which the control system has a set of signals. This kind of grouping is typically described by the structures in the system. Plant section objects typically contain graphic displays and, if control strategies are configured from the functional planning perspective, function diagrams.
- **Process Objects**
More objects representing the process such as Blender, Feeder, Mixer, Pump,

Reactor, Silo/Tank, Valve, Motor etc. Process Objects typically contain control aspects and basic graphic aspects to be used in graphic displays.

- **Signal Objects**

correspond to I/O signals in the control system. If signal engineering is done from functional planning perspective, Signal Objects provide engineering settings to hardware unit channels



Signal Objects are objects for signal engineering as provided by Engineering Workplace: System extension AC 800M Signal Extension, object type group CBM_Signals.

- **System Objects**

For example Network object, Node object, Workplace object, User object, Service Group object etc.

From the control system perspective Aspect Objects are grouped into the groups in Control Structure:

- **Control Networks**

provide the base for process data access.

- **Control Projects**

contain control applications and controllers, tasks and hardware units used to run these applications.

- **Control Applications**

group Process Objects.

- **Controllers**

group Control Applications according to their assignment to Tasks and required Hardware Units.

- **Tasks**

group Control Applications or their Process Objects from execution perspective.

- **Hardware Units**

such as I/O boards group Signal Objects (hardware signals required by the Process Objects and accessed via hardware unit channels).

Aspects

The properties (holding engineering or runtime data) of the Aspect Object are given by the properties of its aspects. An aspect is an assembly of data describing certain properties of an Aspect Object, such as:

- Functional properties.
- Physical construction properties.
- Location properties.
- Control properties.
- etc.

An Aspect Object has several aspects. Each aspect is handled by tools in the aspect system. The aspect system provides preconfigured, ready to use aspect categories to create aspect instances in Aspect Objects.

[Figure 54](#) illustrates the definitions Aspect Object and aspect.

Views

Aspect data are presented by the assigned tools in views. An aspect may have several views.

Opening aspect views gives access to the aspect system with the relevant tool to inspect, create, modify and delete the data.

The data and their mapping may be presented, for example as:

- Lists.
- Tables.
- Diagrams (Circuit Diagram).
- Drawings (Arrangement Drawing).
- Graphics.
- Function Diagrams.

Views are predefined by the aspect systems in the System **800xA** software. Each aspect has one or several predefined views. Refer to [Figure 54](#).

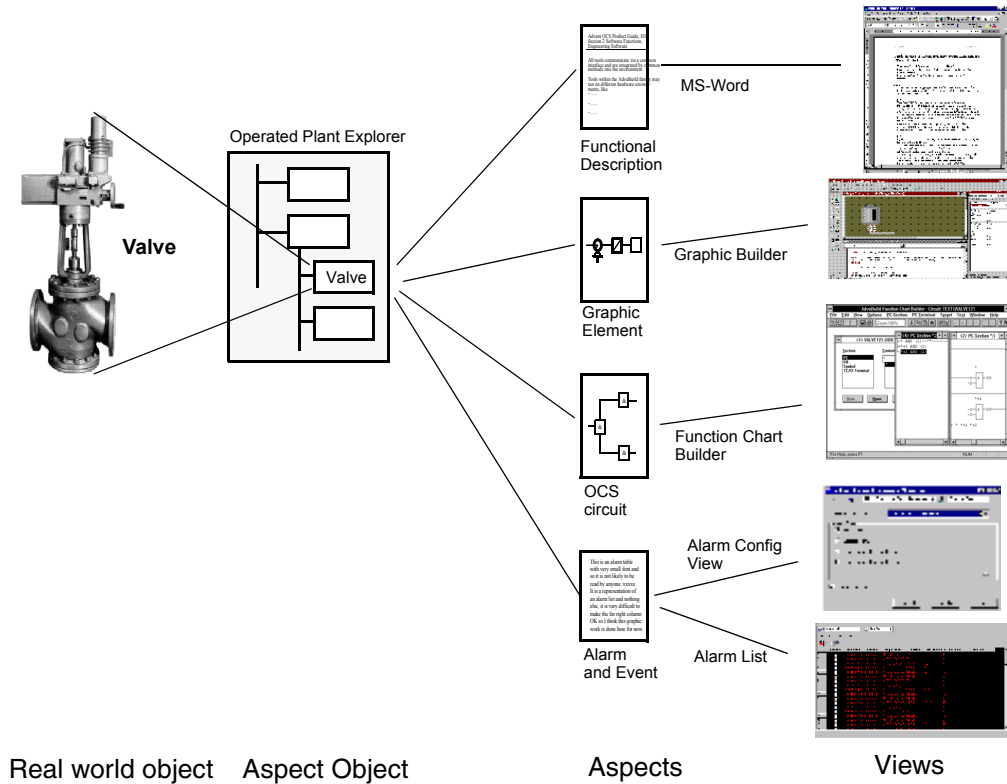


Figure 54. Illustration of an Aspect Object with its Aspects and Views

Object Types

System 800xA provides preconfigured, ready to use object types, for example the object types in the libraries of AC 800M Connect, and allows you to configure your own object types.

An Aspect Object type defines certain characteristics that are common to several Aspect Object instances, such as a basic set of common aspects. This makes it possible to create and efficiently reuse standardized solutions to frequently recurring problems. For example, rather than building an Aspect Object from scratch for

every valve in a plant, you can define a set of valve types, and then create all valve objects as instances of these types.

When an instance of an object type is created, the aspects that are defined in the object type are instantiated and associated with it. You can add aspects to a specific instance, or replace inherited aspects with instance specific aspects of the same type, but it is not possible to delete aspects that were inherited from the object type.

An object type has rules associated with it. These are either aspect rules that control what aspects can be associated with an instance of that type, or child rules that control what objects can be placed as children under an instance of that type, in a particular structure.

A simple object type describes one object; each time it is instantiated, precisely one object is created.

A composite object type describes a set of objects organized in a structure, with a parent object and one or several child objects. The children in a composite object type are called formal instances, because they inherit from object types defined elsewhere in the Object Type Structure, but they are not actual instances. Only when a composite object is instantiated are actual instances created for these child objects.

An Object Type is defined by an Aspect Object in the Object Type Structure. Object types can be collected in Object Type Groups in the Object Type Structure.

The existing object types initially provided in the Object Type Structure differs depending on the system extensions loaded. [Figure 55](#) shows the initial structuring of the 800xA system's Object Type Structure, when 800xA for AC 800M is installed and the AC 800M Connect system extension is loaded.

The object types as shown in [Figure 55](#) are organized as follows:

- **3-rd party OPC server support**
This group is used for generic OPC Support.
- **ABB System**
This group is for internal use within ABB.
- **Control System**
Here is where the network and controller connectivity objects are found, e.g. the AC 800M/C Connect object types in object type groups and control and hardware libraries.

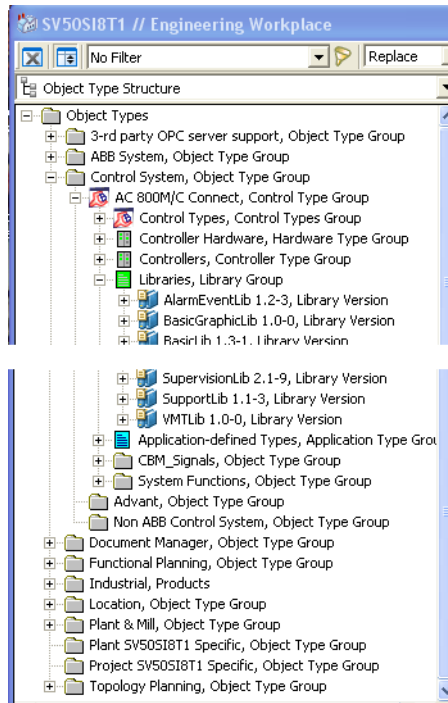


Figure 55. Example Initial Object Type Structure

- **Document Manager**
Object types for Document Manager.
- **Functional Planning**
Object Types to use with and for Function Designer.
- **Industrial**
Object Types for Industrial IT Certification.
- **Location**
The proposed place for Location object types.
- **Plant & Mill**
The proposed place for Plant&Mill object types.

- **Plant <system name> Specific**
The proposed place for the customer's business specific object types.
- **Project <system name> Specific**
The proposed place for your own, project specific object types.
- **Topology Planning**
Object Types to use with and for Topology Designer.
- Additionally other object type groups can exist, depending on the system extensions loaded.

Object Types can be packaged as versioned Object Type Libraries. Libraries are based in Library Structure and provide a container for its Object Types in Object Type Structure, refer to [Library Handling](#) on page 326. For more details about Object Types/Composite Object Types, refer to [Composite Object Types](#) on page 321.

For more information how to configure Libraries and Object Types/Composite Object Types, refer to the *System 800xA Configuration (3BDS011222*)*.

Creating a New Object Instance

You create an instance of an object type by using the New Object dialog (Figure 56).

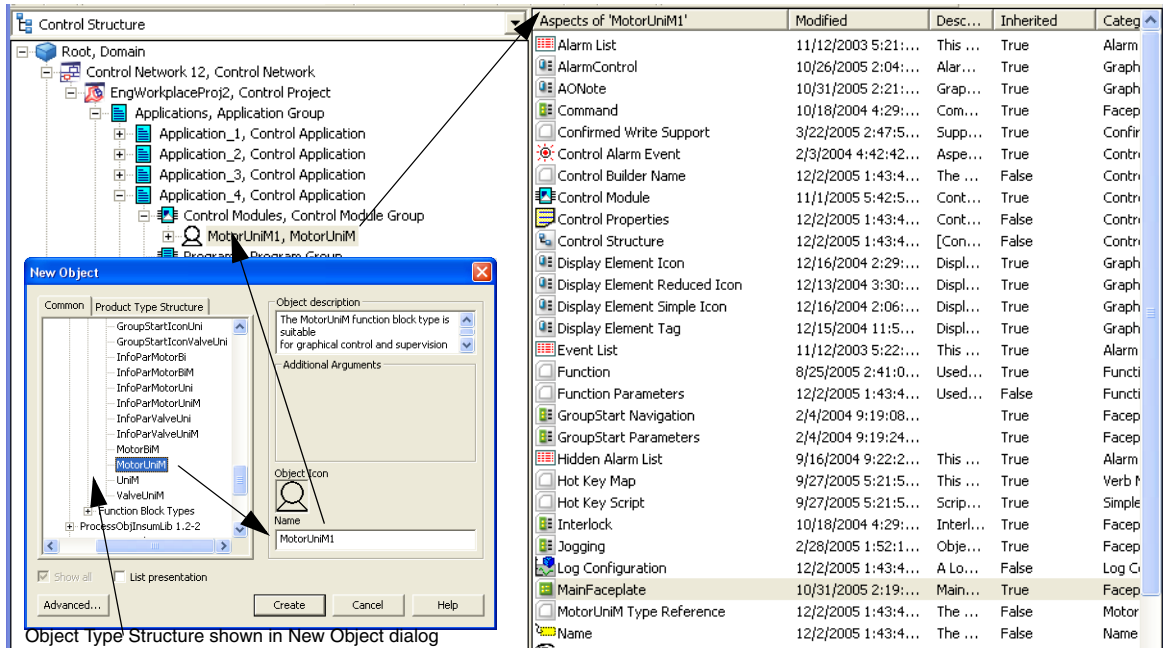


Figure 56. New Object Result for Object Type MotorUniM

Example in Figure 56: A process object named MotorUniM1 is created from the Object Type named MotorUniM provided by the AC 800M Connect library ProcessObjExtLib using the New Object dialog on the object Application_4.

Functionality automatically available through inheritance from this object type is (among others):

- A Control aspect.
- A Function aspect for Functional Planning.
- A couple of Graphic Element aspects.
- A MainFaceplate aspect.

- An Alarm List aspect and an Event List aspect - providing a list of alarms and list of events for this specific process object only.
- A Trend Display aspect - providing an online trending of the signal, or if there is a history log aspect for an object, providing trending of the history data together with an online trending.

Basic Structures

The Plant Explorer tool of System 800xA allows you to build up and maintain a model for the plant.

Depending on the project requirements you use the basic structures listed below or even more of the provided standard structures.

All existing predefined structures are described in [Object Structures](#) on page 342.

The relations between aspect objects are established by placing them in one of the following predefined basic structures for aspect objects:

- **Functional Structure**
contains objects with Graphic Displays and, in case the Functional Planning approach is chosen to configure the automation solution, objects with Function Diagrams and objects for Engineering Signals.
- **Control Structure**
contains control related objects and, in the Functional Planning case, objects with Function Diagrams and Signals, allocated to corresponding control objects.
- **Library Structure**
contains versioned Object Type Libraries and template objects and aspects
- **Object Type Structure**
contains Object Types of Object Type Groups and versioned Object Type Libraries.

Placing an object below another object in the structure in general expresses that the placed object belongs to this object. The two objects have a structural parent - child relation. In which sense the placed object belongs to the object above is determined by the aspect systems.

Placing aspect objects into these predefined structures supports the navigation among engineering data and, if configured accordingly, the designation (refer to [Names and Designations](#) on page 316) of the aspect objects in the plant.

You may also use the structures as an end-user of different categories (Operator, Maintenance Engineer,...) for fast navigation to, and location of, certain information.

One aspect object can be part of several structures at the same time, refer to [Figure 57](#).

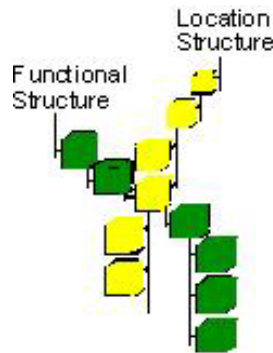


Figure 57. One Aspect Object in Two Structures (Functional and Location)

Names and Designations

A name typically identifies an object without regard to the hierarchy level in the structure the object resides. Whereas a designation typically identifies the object based on references to the hierarchy levels. The relative reference designation is identifying a single hierarchy level and the absolute reference designation provides an hierarchical name concatenating the reference designation of the levels above and of the level the object resides on.

Identification of Aspect Objects

Name and Designation aspects support identification of Aspect Objects from user's perspective. Internally the system identifies objects using GUIDs (Globally Unique Identifiers).

To each level in the basic structures (represented by Aspect Objects) you should give names and/or designations that allow you to identify the objects unambiguously according to the naming and designation conventions required in the engineering project.

The Plant Explorer supports for this purpose two different concepts:

1. Identification of Aspect Objects by name:
This is the default mechanism using the **Name** aspect and is the preferred concept, if the customer uses no reference designations.



You can, for example, follow the naming conventions described in the *System 800xA Control, AC 800M Configuration (3BSE035980*)*, subsections I/O Naming and Suggested I/O Signal Extensions. This proposal is based on ANSI/ISA S5.1 standard. It codes a designation into object name, therefore it is applicable if you only work in Control Structure or if one only designative name shall be used/visible for an object in all structures.

2. Additional identification of Aspect Objects by reference designations are kept in its own Designation aspects and optional display of the absolute reference designation. This is the preferred concept, if the customer uses reference designations according to IEC 61346. Refer to [Figure 58](#).

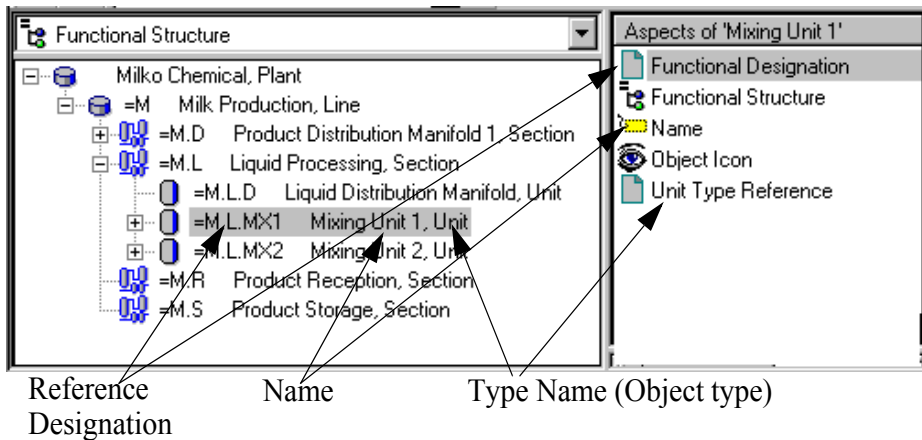


Figure 58. Identification of Aspect Objects

Dependant on the requirement that objects shall be identified primarily via name or primarily via reference designation the name has a primary role in identification or not. In the first case you shall choose a name that really describes the purpose of the object, in the second case you need not. But to use descriptive names is good advice even in the second case.

At least when building the upper levels of the basic structures one of those concepts should be used consequently. When adding details during later design stages, it might be necessary to use combinations of reference designations and names for identification.

Additional Names

Aspect Objects can have additional name aspects, which you can use to implement your own name spaces for the needs of certain users or which are used for the needs of certain tools.

Such aspects are Plant Name, Function ID, Document ID, Instance ID, Instrumentation ID, Location ID, Process ID, Product ID, Mechanical ID, Reference ID, Object Category, Control Builder Name, OPC Source Name.



You should only use the additional Name aspects if there is a serious requirement for additional names.



To make additional names visible in structures you have to customize your Plant Explorer settings in the Name Composer tab.



All name aspects and their Name property values can be searched for in Plant Explorer's Find Tool and in Bulk Data Manager.

Name Handling

Additional names can be displayed in Graphic Displays by designing the display accordingly. They are not displayed in Faceplates as provided by available standard libraries, for example for AC 800M.

For the Name aspect the Advanced object menu item Rename Substructure allows you to rename the objects in the substructure by substituting a certain substring of the names by a new one. The same scheme is used when instantiating a composite object type or copying a composite instance in Bulk Data Manager.

The Object Category aspect you should use when certain objects need categorization by a common name, for example the name Board for all objects that represent board hardware units.

OPC Source Name is a read-only name filled by the Uploader of certain controller connectivities, used to identify alarms.

Control Builder M Professional uses an additional Control Builder Name aspect to automatically map the name given in the Name aspect of a control object, which can contain an arbitrary name, to an IEC 61131-3 compatible object name.

After download you can synchronize Name and Description properties of the Name aspect from Name and Description parameters of the control aspect using the Name Uploader of Control Builder M.

For engineering efficiency, Function Designer synchronizes the Name aspect, Control Builder Name aspect, Name parameter and Description parameter for its control objects. Name parameter and Description parameter are properties of the control aspect running in the controller. Function Designer further allows you to use names starting with digits. This is required when using a KKS (Kraftwerk-Kennzeichensystem, a german power station identification system) like designation scheme in names. Function Designer also includes a rename substructure functionality as mentioned above for the Advanced object menu Rename Substructure and Bulk Data Manager. The advantage of having synchronized names is that objects always have the same name in Plant Explorer and Project Explorer with the exception of IEC 61131-3 name restrictions.



Do not use Control Builder M Name Uploader when you are working with Function Designer

Reference Designations

The general principles and ideas behind reference designations are described in international standard IEC 61346. Efficient identification of components when running and maintaining a large plant is the typical reason for the requirement of reference designations.

Absolute reference designations are built automatically within Plant Explorer by concatenation of relative designation parts according to structure hierarchy.

You can configure generation of absolute reference designations for the following structures:

- Functional Structure by use of **Functional Designation** aspect.
If absolute functional reference designation shall be shown in Plant Explorer's Functional Structure, Graphic Displays, Function Diagrams, and/or dynamic documents.
- Location Structure by use of **Location Designation** aspect.
If absolute location reference designation shall be shown in Plant Explorer's Location Structure and/or dynamic documents.
- Control Structure by use of **Control Designation** aspect.
If you use the Control Builder M Signal Extension system extension Control Designation aspects are provided in hardware unit object instances for AC 800M by default. Absolute control reference designations are displayed by I/O Allocation tool and Function Designer, relative control reference designations by Topology Designer. The control reference designation is determined by Control Builder M Professional's hardware unit subtree in Project Explorer.
- Documentation Structure by use of **Documentation Designation** aspect.
If absolute documentation reference designation shall be shown in Plant Explorer's Documentation Structure and/or dynamic documents.

Example for function reference designation:

- The relative part of function reference designation is stored per aspect object in a **Functional Designation** aspect within properties:
 - **Name:** relative designation string.
 - **Prefix:** prefix used for the relative designation.
- The absolute function reference designation is concatenated according relative designations and hierarchy:
Level 1: = M -> =M

Level 2: = D -> =M.D

Level 3: = SL1 -> =M.D.SL1

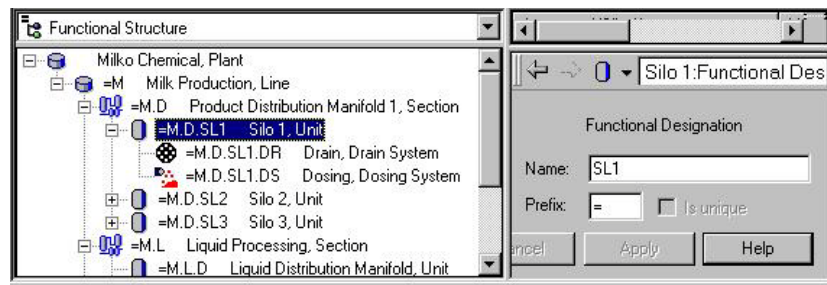


Figure 59. Building Absolute Function Reference Designation



You should only use additional designation aspects if there is a serious requirement for structured plant modelling with reference designations, refer to [Appendix B, Structured Plant Modeling](#).



Before you implement reference designations you have to design and describe the designation system, taking care of the standards and the detail customer requirements on separators, relative reference designation identifiers and structures to use. An example how a designation system can look like is shown in [Appendix C, Designation System](#).



To make Absolute Reference Designations visible in the structures you have to customize your Plant Explorer settings in the Name Composer tab, refer to the *System 800xA Engineering Engineering Studio (3BDS011223*)*, subsection on how to Display Absolute Reference Designations.



You have to use Bulk Data Manager to identify objects by Absolute Reference Designation. Plant Explorer's Find tool does not support identification by Absolute Reference Designation.

Composite Object Types

Composite Object Types are used to define typical solutions that decrease the engineering effort. This section outlines the principles how to build and use composite object types.

The reasons for building composite object types are:

- The engineering time is decreased due to field proven solutions. A composite object type that is tested and verified does not need the same amount of testing when it is reused.
- By using composite object types, you are able to break down your system into blocks. This makes it easier to build and maintain the structures.

A **Composite Object Type** describes a set of aspect objects organized in a structure, with a parent object and one or several child objects. The children in a composite object type are called **formal instances**, because they inherit from object types defined elsewhere in the Object Type Structure, but they are not actual instances. When a composite object is instantiated, actual instances are created for these child objects.

A composite object type also includes a description of how its child objects shall be inserted into different structures when the composite object is instantiated. The instantiation is described separately for each structure using a **formal instance list**. Below the formal instance list the wanted object hierarchy for a certain structure is specified. Refer to [Figure 60](#).



For technical reasons composite object types have to be put into a library, a library extensions (refer to [Library Handling](#) on page 326) or have to belong to a system extension.

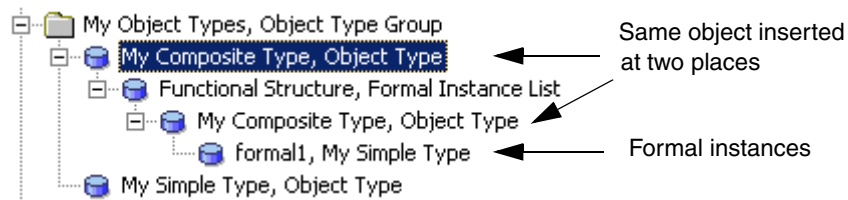


Figure 60. A Composite Object Type

When you instantiate a composite object type you get a composite object instance. The composite instance is one single unit, although it can include several child objects spread out in different structures.

Adding Additional aspects

To make the Composite Object type really useful you should do as much aspect configuration as possible on the object type. For example, you may want to add a Graphic Element aspect on the main type object that either presents data retrieved from its child objects or incorporates subordinate Graphic Elements that already exist on the child objects.

To be able to reference the child objects correctly in all composite instances, we use relative references. All formal instances have Relative Name aspects that are inherited to all corresponding actual child objects. The actual instances also have a local Name aspect that initially gets the same value, but the user can change it to an instance-specific name. Using the inherited and fixed relative names, a child object can still be identified using a relative name path consisting of a series of relative names.

For example Graphics Builder automatically inserts such relative references when you create a Graphics aspect on an object type. Its object list in the Expression Builder and Element Browser dialogs shows the available formal instances as shown in [Figure 61](#).

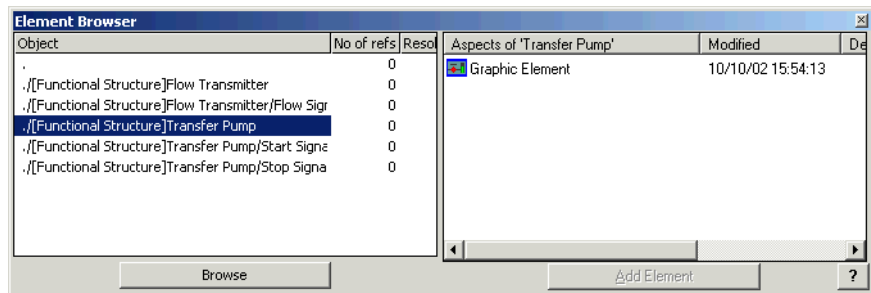


Figure 61. Element Browser

Other aspects often created on composite object types are the Faceplate aspect, the Alarm List aspect, and the Trend aspect.

Allowed Operations on Composite Instances

Instances of the composite type must follow the structure defined by the type, and instances must be created and deleted as a whole unit. For example, it is not possible to instantiate only a part of a composite type, or to delete a sub-object from an instance of a composite type.

Though it is allowed to add additional sub-objects to an actual instance, without defining corresponding formal instances in the type. The type only defines a minimum structure for the instance, not the complete structure. The minimum structure can be extended in each instance, but not reduced.

Details About Formal Instance List

What is the difference between the object types A and B in the example below?

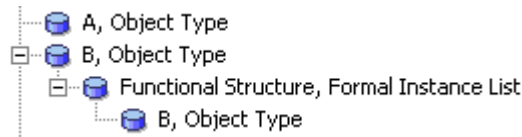


Figure 62. Formal Instance List

Type A is a simple non-composite object type. If an instance of A is created in a structure, for example Functional Structure, it will only be placed in that structure.

The type B on the other hand is a composite object type, although it has no child objects. If an instance of B is created in Functional Structure, nothing special would happen. On the other hand, if you create an instance of B in e.g. Location Structure, the instance will also be inserted in Functional Structure. This is because the composite object type requires that the instance always is present in that structure.

When objects are inserted into other structures due to formal instance lists, they are always inserted at the top of those structures. They can then be moved to the wanted place.

The composite object type is a general mechanism that can be used to create quite complex objects in many levels. One composite object type may use other composite object types, which in turn use other composite object types, and so on.

At each step, the formal instances can be extended with local aspects, and extra child objects may be added.



However, if you build and use composite object types based on object types of an connectivity package, for example for AC 800M, there may be restrictions. Refer to the documentation for your connectivity install package for more information.

For configuration steps and examples for composite object types, refer to the *System 800xA Configuration (3BDS011222*)*.

Typical Solutions

Defining and reusing typical solutions is a suitable method to decrease the engineering effort.

You can build and reuse a typical solution as a Composite Object Type as described in [Composite Object Types](#) on page 321.

Or you can build it outside the Object Type Structure for example in the Functional Structure as a composite object instance. The object instances in such a (typical) solution can be derived from object types and / or composite object types and / or from the generic object type.



When ever possible build your typical solutions as composite object types in the Object Type Structure and include them into a version handled library (refer to [Library Handling](#) on page 326). Using version handled object type libraries results in better application quality, less validation costs, and the possibility to reuse well proven object types/ composite object types in other projects.

You can easily reuse a typical solution built either as a composite object type or as a composite object instance using the spreadsheet based configuration functions of Bulk Data Manager to populate your plant model in the system. Composite object types you can instantiate and parameterize several times via a Bulk Data Management worksheet, composite object instances you can copy and parameterize several times via a Bulk Data Management worksheet.

Reuse Instructions

Often the task is given to manage, design and use typical solutions with variants. The system extension Reuse Assistant of the 800xA system supports you in performing this task.

The Reuse Assistant Architect Mode enables you to design Reuse Instructions in the Reuse Design Structure. For such an instruction you define substitution variables and a tree-like structure of questions with a set of possible answers and operations to be performed for each answer. After you have finished design you generate the Reuse Instruction into a corresponding Reuse Instruction aspect of an object type in the Object Type Structure.

After you have instantiated the object type you use the Reuse Assistant Build Mode on the ReuseAssistantBuilder aspect to apply the instruction stored in the Reuse Instruction aspect to the instance. In a wizard-like view you are 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, objects get created or modified accordingly. Typically a composite object instance results.

With one Reuse Instruction you can for example generate several variants of a highly standardized Control Builder M Professional Program or Single Control Module.

Values for substitution variables of Reuse Instructions can be set using Bulk Data Management worksheets.

For configuration steps, refer to Building Reuse Instructions in the *System 800xA Configuration (3BDS011222*)*. For more details, refer to the *System 800xA Engineering Engineering Studio (3BDS011223*)*.

Library Handling

System 800xA supports development, distribution and management of versioned libraries. Libraries are collections of object types and composite object types.



For technical reasons composite object types have to be put into a library or belong to a system extension.

Libraries in 800xA are used to support the library concept in Control Builder M, but they can also be used for general purposes.

AC 800M connectivity of System 800xA comes with a set of libraries which are the initial libraries for AC 800M application development, refer to [Figure 63](#).

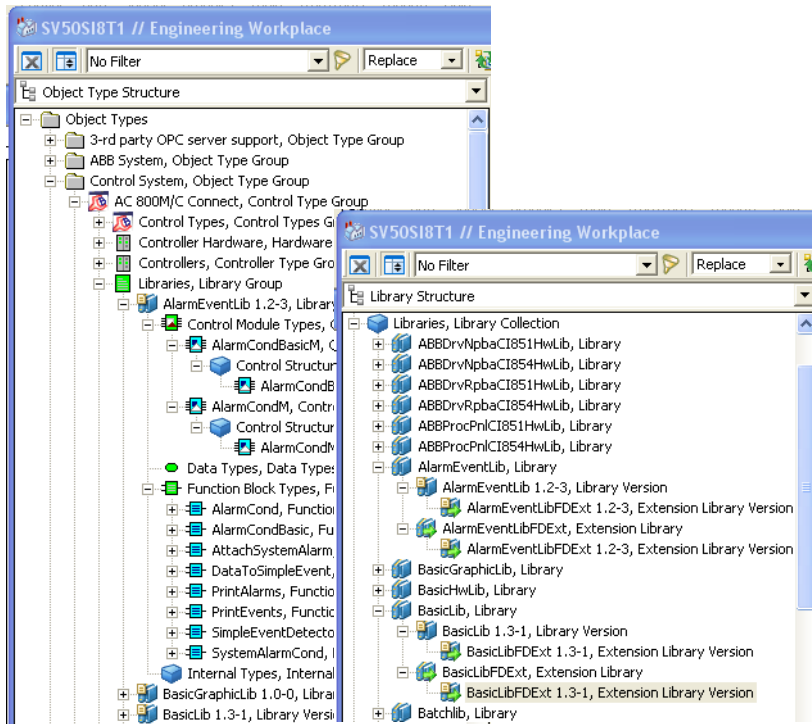


Figure 63. Libraries for AC 800M

In libraries you can accumulate your solution knowledge developing over time and make maximum profit from it.

Versioning of object type libraries gives the possibility to distribute, manage and track object type changes in a standardized, traceable way.

A Library Version object defines a version for a base library underneath a Library definition object in the Library Structure.

The Library Version object is also used as the root object for the object types of the base library in the Object Type Structure. This means that all object types of a Base

Library Version are created or moved underneath this object in the Object Type Structure. A new Base Library Version requires a new Library Version object.

The Library Member aspect represents object type's membership in the certain Base Library. This aspect is placed on every item, which belong to the Base Library. A Library Member keeps information about the current library and the history of the member inside of the library (e.g. creation date/time, who and when has added the new aspects to the library).

There are two types of libraries; Base Libraries, often just called libraries and Extension Libraries. A Base Library consists of new object types. These are created in the Object Type Structure. Each Base Library has a Library object with a Library Definition aspect in the Library Structure.

A Base Library can be extended with one or several Extension Libraries that contain additional aspects. An Extension Library can only contain aspects, refer to [Figure 64](#).

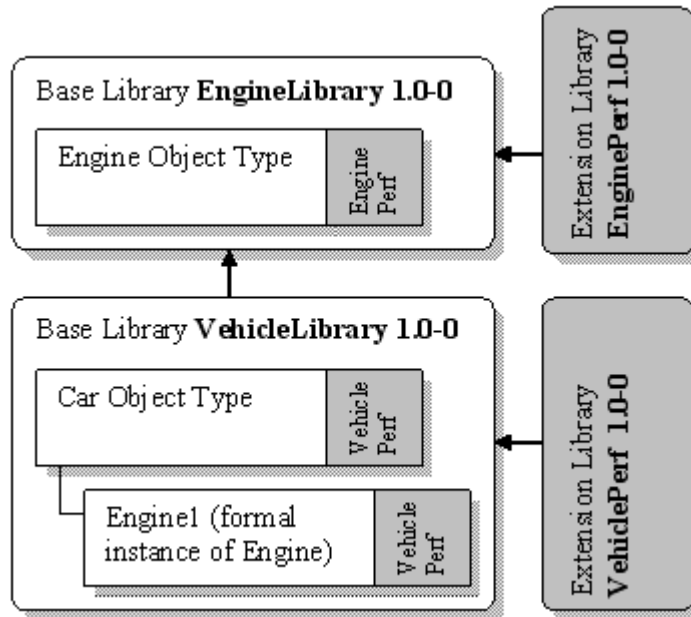


Figure 64. Example of Base Libraries and Extension Libraries

A Base Library is a sub-tree in the Object Type Structure, containing object type groups, object types and aspect groups. Objects from other structures cannot be placed in libraries.

Each base library has its own version number, e.g. 1.0-0. Several major versions of a base library can exist in parallel, to allow different applications to simultaneously use different versions of the library. Using a base library version here means to instantiate object types from that library version, or to have a reference to an aspect group from that library version.

Because major base library versions need to exist in parallel, the complete structure is copied and all object types get new IDs when a new version is created.

The drawback is that all user applications have to be reconnected to the new version. To avoid that, it is also possible to create minor versions and revisions that overwrite the previous data in the library version. Such minor versions of a base library cannot exist in parallel.

A base library version also has a state. It is typically **Released** when the library is used by end-users. If a library is released, then no aspect defined in that library can be changed or deleted. This can facilitate testing considerably, because we can have frozen and well-tested libraries available that do not have to be tested for every project.

However, it is allowed to add additional aspects to a library that are not considered as a part of the base library. This can be done also for released base libraries. The added aspects can either be placed in a version-controlled extension library, or just be unassigned and not belong to any library.

An Extension Library is always connected to a specific base library, and owns selected aspects placed on the objects in the base library. An extension library can only contain aspects. The objects (and their structure aspects) must always be placed in the base library.

An extension library has version and state, similar to a base library. However, two extension library versions can only be used in parallel if they are connected to different base library major versions. Other extension library versions can still exist however, if they are set to inactive state. An inactive extension library version still exists in the system, but the aspects are not visible on the objects of the base library.

As mentioned above, all essential data in a base library or an extension library are always placed in the Object Type Structure. But the libraries themselves have additional administrative information that is placed in the Library Structure.

Each Extension Library has an Extension Library object in the Library Structure.

An Extension Library Version object defines a version for an extension library underneath an Extended Library definition object in the Library Structure.

Unlike Base Libraries, Extension Libraries are only inserted into the Library Structure, not the Object Type Structure, but the extension aspects are placed on objects in the Object Type Structure that belong to the Base Library.

For more information about Library handling configuration, refer to the *System 800xA Configuration (3BDS011222*)*.

Library Management

A library introduces object types from which applications can be built. A library has its own version and is, when used by end-users, normally released. A released library cannot be changed and the system safe-keeps the integrity of a library above security settings and similar protection mechanisms.

Note that this means that if a library is released, then no aspect defined in that library can be changed.

Libraries are distribution units that support version handling and controlled upgrades, i.e. changing an application from using one version of a library to another.

Operations on a Library

The main operations that you can perform on a library as a library developer are:

- Create a library, and an initial library version.
- Add, delete and modify object types and other items that belong to the library.
- Close a library version.
- Open a closed library version again.
- Release a library version.
- Export a library version.

- Create a new library version.
- Remove libraries and library versions.

The main operations that you can perform on a library as an end-user are:

- Install a library version.
- Remove a library version.
- Create and use the instances based on the library object types or aspect groups.
- Upgrade/downgrade the chosen instances to the certain version of a library.

Libraries as Distributable Units

A library (base library or extension library) in any state can be exported and distributed as an *.afw file.

Each base library version and each extension library version should be placed in a separate *.afw file.

The parent in the Object Type Structure where the base library version is to be inserted, should be distributed separately, and not inserted into the *.afw file for the library.

Internal distribution of libraries within a development project may be done by importing *.afw files directly. When this *.afw file is loaded into a system then it is represented by an instance of the Library and Library Version type.

For external distribution, it is recommended to package the files as a system extension. Several *.afw files can be included in the same system extension.

Only files packaged as system extensions are allowed to overwrite a released library version. To be able to distribute an updated minor version or revision of a library to other users, it must be packaged as a system extension. The system extension framework will only load the updated *.afw files if a previous version of the system extension is already loaded, so the system extension can contain both a base library and its extension libraries even if only one of them has been updated.

To package a library as a system extension use the installation kit provided by Aspect Express, refer to the *System 800xA Engineering Aspect Express (3BSE032793*)*.

Library States

A Library Version has a state. This state is used to describe how a library version can be used. The states a library can be in are: Open, Closed or Released.

In the open state the contents of the Library Version may be changed. Object types can be added, modified, deleted or moved freely. When editing temporarily is completed, the library can be closed. Or if the library is ready for release it can be released.

When a Library Version has been released then it can no longer be changed. If any modifications are necessary then a new version of the library should be created. Note that a library cannot be released unless all libraries it depends on are released (Figure 65).

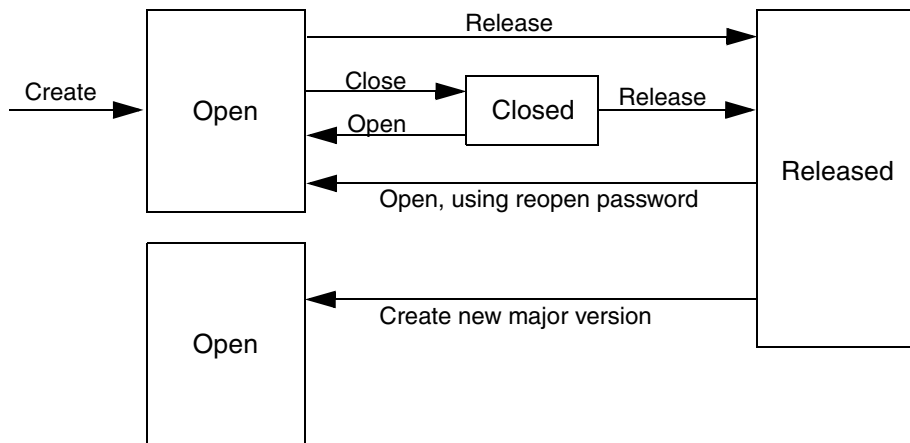


Figure 65. Library Version State Diagram During Development

Figure 65 illustrates the states through which a library will evolve. You normally release a library just before you send it out to the end-users of the library. If the library is only used internally, it is not meaningful to release the library unless you need to run two versions in parallel and therefore want to create a new major version. It is possible to reopen a released library to create a minor version of the library, if the library password is known.

Extension Library States

An extension library supports the same states as a base library; Open, Closed and Released. The version state of an extension library is independent from the state of its base library. It is possible to release one of them while the other one still is open.

AC 800M related extension libraries shall only be closed/released if the corresponding base library is closed/released.



During the Control Builder project upgrade, Function Designer related extension libraries for AC 800M base libraries must be open if the base library is open. An extension library only containing graphics can still be closed/released.

Unlike a base library an extension library can also be active or inactive. If the extension library is active, all its aspects are placed on the object type groups, object types or aspect groups that it extends.

If an extension library is inactive, the aspects of the library are no longer visible. Only the extension library objects in the Library Structure are visible.

Many different extension libraries can be active on a base library version, but at most, one version of each extension library can be active at a time for a given base library version. If a new extension library version is created, then any version on the same base library version is automatically inactivated (Figure 66).

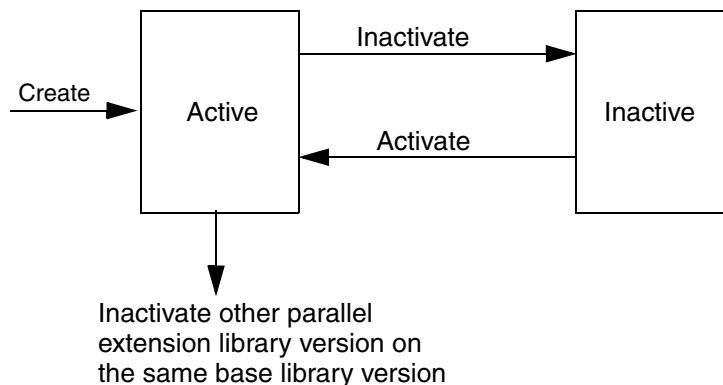


Figure 66. Activation/Inactivation of Extension Library

As for base libraries it is possible to reopen a released extension library to create a revision of it.

Entities

Entities make it easier for users to understand which parts will be affected by a change and define for system applications which parts must be handled as a unit.

An Entity is a collection of objects and aspects that a user treats as a unit with respect to various phases of engineering. An Entity is exactly aware of what it contains in terms of sub-objects and Aspects in one or more structures, such as the Control Structure or the Functional Structure.

Entity types are predefined by the base system and by system extensions:

- Library (this covers all Libraries, including the existing AC 800M Control Code libraries and the Hardware Libraries).
- Extension Library.
- Control Module Types / Function Block Types / Diagram Types.
- Control Project.
- Control Application.
- Controller.
- Function Diagram.



A Function Diagram defines one or more Allocatable Groups in the Functional Structure. It is actually these Allocatable Groups that form entities, not the Function Diagram as such. When a Function Diagram Type extends a Control Module Type in a library, it is the control module type (or an extension library) that forms an entity, not the Function Diagram Type.

- PROFIBUS/HART Fieldbus Topology.
- FOUNDATION Fieldbus HSE Subnet.

When a new version of an entity is imported to a system or deployed to an environment where it already exists, the existing entity will be replaced in its entirety, i.e. there is no merge. Hence, moving an entity may cause objects and aspects to be deleted, as opposed to when individual objects and aspects are moved.

An entity is typically defined as a sub-tree in a structure, such as the Control Structure or the Object Type Structure. The top level object has an aspect that defines the entity, and the entity includes all descendants. Entities can be nested, i.e. an entity can include entities. The hierarchy is strict, i.e. each object and entity can only be (a direct) part of one entity. For example a Function Diagram is a sub-entity of a Control Application.

Import/Export (refer to [Import/Export](#) on page 272), Deploy, Update and Replace tools are entity aware.

Deploy moves entity contents from Engineering to Production environment. The concept of environments are explained in [Environments](#) on page 339.

Update moves changes to aspects made in the Production environment into the Engineering environment. The Update tool has a similar user interface as shown for the Deploy tool in [Figure 67](#).

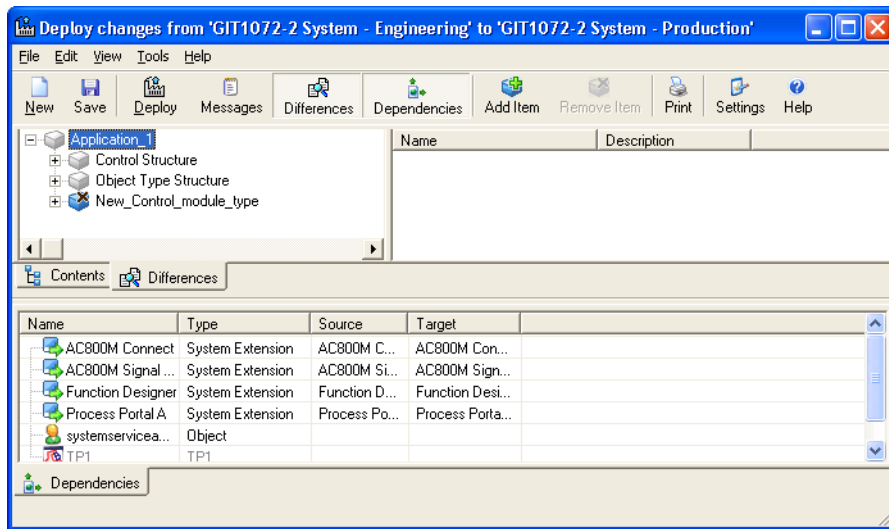


Figure 67. Deploy Tool

Replace overwrites the changes made in Engineering environment with the entire entity contents of the Production environment. The Replace tool has a similar user interface as shown for the Deploy tool in [Figure 67](#).



Update cannot be used with entities containing Control aspects. You have to use the Replace tool.

The entity concept is always supported, regardless of whether environment support has been selected or not.

An entity can be reserved by a user. This enforces exclusive modify access to the entity for a single user. The purpose is to prevent unintentional overwrite of configuration data from concurrent users. The user can explicitly reserve the entities that he/she intends to modify, and release the reservation when ready. The concept of reservation is explained in [Reservation](#) on page 338.

Browsing and managing an entity can be done from the Manage > View <entity> object context menu item (Figure 68).

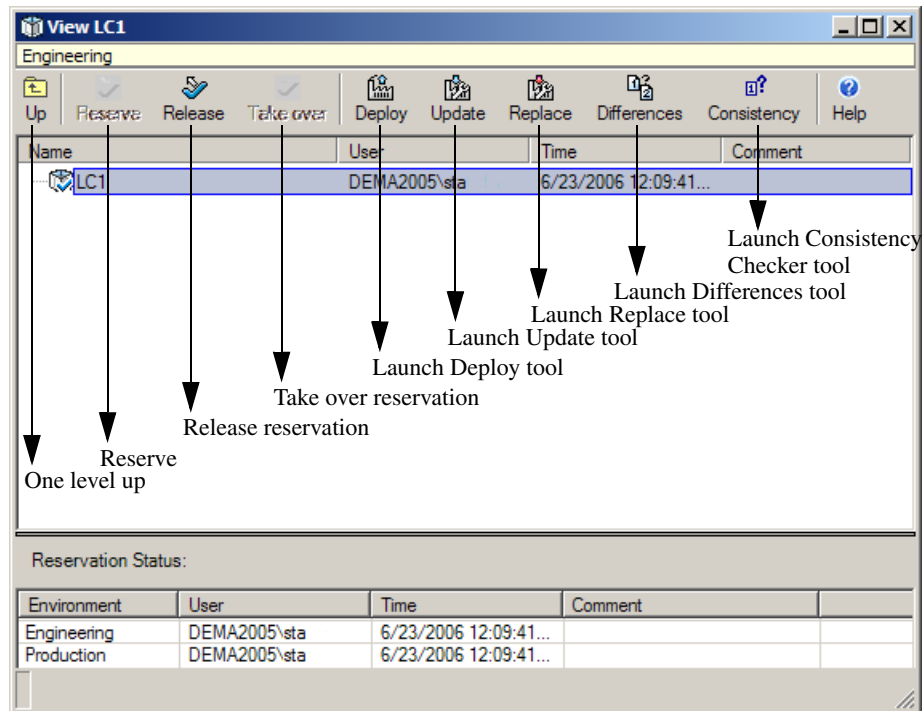


Figure 68. View <entity> Dialog

The opened View <entity> dialog shows the contents of the entity regarding structure, objects and aspects and allows to:

- Reserve and Release reservation of the entity.
- Deploy, Update or Replace the entity.
- Show Differences of the entity contents against any other environment.
- Check consistency.

To Reserve, Release, Deploy, Update or Replace an entity is also available as an object context menu item.

Reservation

A user can reserve items that he is working on, to prevent other concurrent users from accidentally overwriting his work. Reservation guarantees exclusive modify access for a single user to a reserved item.

Entities, entity extensions and single aspects outside entities are items that can be reserved by a user. Single aspects outside entities can only be reserved if Environment support is enabled.

If an aspect is part of an entity, the aspect can not be reserved alone. Instead the entire entity must be reserved.

When a user has reserved an item it will stay reserved until the user manually releases it. Reserving and releasing generate audit events.

Only one user can reserve an item, and while the item is reserved, no other user can modify it. However, another user that has the proper privileges can take over the reservation. This is useful to prevent that items are locked-up indefinitely if the user holding the reservation for any reason is not available.

It is not necessary to reserve an item in order to change it. However, before performing the change, the system checks that the item is not already reserved by another user. If it is, the system refuses to perform the change. Note that configuration tools may enforce reservation as a mandatory step in the workflow defined by the tool.

The system may automatically reserve items in certain situations, for example as a side effect of another reservation. Automatic reservations done by the system in this way are also released by the system.

Reserve/Release

The Reserve/Release operation relates to entities, entity extensions and to single aspects (as already mentioned above). Entities are handled as unbreakable units.

- **Reserve** means that all aspects within the entity are, or the single aspect is locked for reservation by other users.
- **Release** reservation un-locks all aspects within the entity or the single aspect so that they, it can be reserved by other users again.

- **Take Over Reservation** allows a user to break another user's reservation and make the aspect available for changes again. To take over a reservation the Take Over Reservation permission is required.

Reserve/Release reservation for an entity can be executed from the View <entity> dialog started from the entity object context menu item **Manage...** The scope of a reservation is the current environment. As an example, an entity can be reserved by a user working in the Engineering environment, and at the same time by another user in the Production environment. Reserve/Release can also be executed from the the corresponding Aspect context menu items **Reserve... / Release...**

Additionally Reserve/Release commands are available in the Control Builder M Professional and the Function Designer. These tools enforce reservation for entities to be worked on.



Reservations of single aspects that are not part of any entity cannot be kept during deploy. When deploy is made of such aspects, they will automatically be released by the deploy tool.

Environments

In 800xA versions prior to 5.0 there was only one environment, the Production environment, in which objects and aspects could exist. Engineering changes had immediate impact on the functions that were used to control and supervise the process.

Starting with System 800xA SV 5.0, it is possible to create an additional Engineering environment for a system. Creating the Engineering environment is done as a System Administration task in the Configuration Wizard using the Environment setup dialog, checking the Environment support check box. Creating environments is non-reversible. The only way to turn off environments is to delete the system and create a new one or make a restore of a backup. Environment support requires a corresponding license.

The Engineering and Production environments may hold different versions of one or more aspects. Each environment determines which version of an aspect that is

current in that environment (Figure 69). This makes it possible to make and test changes isolated from the on-going production in one system.

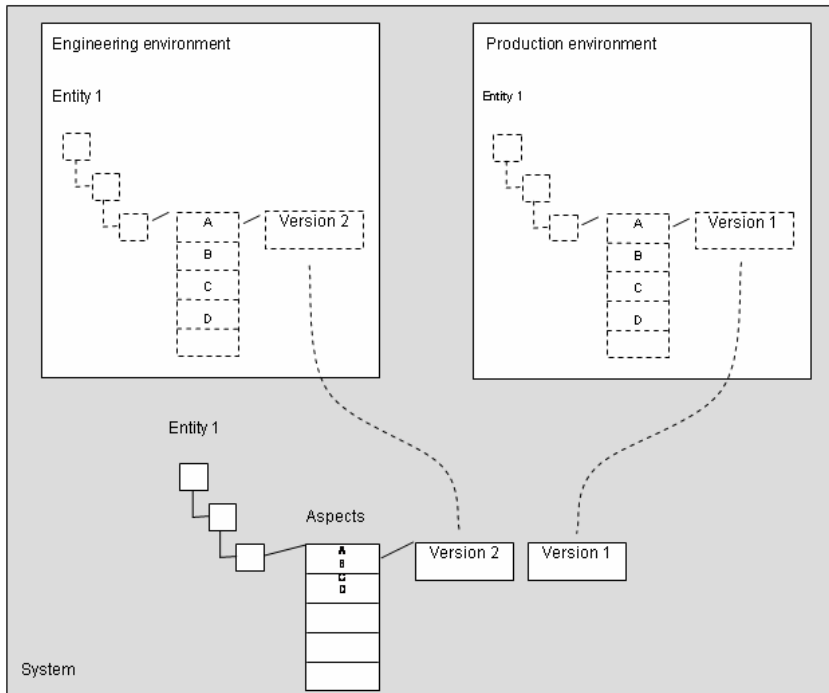


Figure 69. Conceptual View of Environments

A typical way of working is to do the engineering work in the Engineering environment and then deploy the changes to the Production environment. This way of working is called Configure-Deploy.

The normal starting point for an engineering session is that Production and Engineering are exactly identical, i.e., the same object and aspect versions are exposed in both Environments. This is automatically given after having switched on Environment support initially in the Environment setup dialog in the Configuration Wizard tool. During work this can be achieved by checking the Refresh engineering environment check box of the Environment setup dialog.

The normal workflow is that an engineer opens a workplace in the Engineering environment, and as changes are made new aspect versions are created and made visible within the Engineering environment only. Normally, engineers make changes of aspects and objects that are part of entities, e.g. libraries, but it is also possible to make changes to any version-enabled aspect. Note that deleting objects and aspects in the Engineering environment does not affect the Production environment until the changes are deployed.

It is possible to make changes in the Production environment in parallel with changes made in the Engineering environment. To ensure that such changes are taken into account, the Engineering environment may need to be updated before any deployment is done.

In addition to the Production and Engineering environments, the Load-Evaluate-Go (LEG) function uses two more environments:

- Evaluate environment - Holds versions of aspects that are being evaluated through the Load-Evaluate-Go procedure. During LEG the changes done in Engineering environment are deployed to the Evaluate environment.
- Previous environment - When a LEG session is started, the content of the Production environment is transferred to the Previous environment to enable a fast rollback in case the new version does not behave as intended.

Configuring the system to create these environments is done as a System Administration task in Configuration Wizard using the Environment setup dialog, checking the Environment support check box and the Load-Evaluate-Go check box. Both Environment support and Load-Evaluate-Go requires a separate corresponding license.

The Evaluate and Previous environments are created when a LEG session is initiated, and deleted when the session is terminated. Special rules apply to the use of these environments.

Environments are represented as objects in the Admin Structure.

Aspect Versions

When Environment support is enabled, a change of an aspect results in a new version of this aspect. This version is only visible in the environment in which the change has been done. For a user, handling of aspect versions is not visible directly.

Production-only Aspects

Depending on the aspect system some aspect categories may not support versions. An aspect that is not version handled exists in only one version. Such aspects can only be created, changed, or deleted in the Production environment. In the Engineering environment, production-only aspects are read-only.

Entities including such aspects cannot be created or deleted in the Engineering environment. It may even be impossible to change version handled aspects of such entities in the Engineering environment.

For further information on Environments, refer to the *System 800xA Engineering Engineering and Production Environments (3BSE045030*)*.

Object Structures

In this appendix you will find an overview of the basic object structures present in Plant Explorer. The system comes with a set of predefined structures. The set of structures may be extended by an arbitrary number of structures by anyone using the system. The structures described have in some cases been filled with more data than when first installed.



The structures described in this section are needed for correct system operation and **must not be removed or modified** in any way. Normally, these structures enforce their own rules, for example what can be inserted.

Each structure is described in more detail in the sections below. The system contains the following primary structures for a workplace:

- [Functional Structure](#) - gives the functional structure of the system.
- [Location Structure](#) - contains the location of objects.
- [Control Structure](#) - gives the control structure of the system.
- [Workplace Structure](#) - contains the workplace definitions of a system.
- [Graphics Structure](#) - which holds the alarm and event list configurations, the view classes, the tools, the workplace frames, the workplace panels, and the object type library objects.

These structures are primary structures, because it is assumed that most Operators and Application Engineers in the running plant will perform most of their work within these structures.

Some additional and important structures are:

- [Admin Structure](#) - contain the administrative structure of the system.
- [Object Type Structure](#) - contains the object types to be used in the system.
- [Aspect System Structure](#) - contains the aspects to be used in the system.
- [It is reflected in the Toolbox in Graphics Builder, and also in the corresponding wizard page.](#) - defines graphic primitives and subelements.

These structures contain the objects that are used to configure and control the system itself.

The administration structures are all structures that are precreated, and which may contain only objects that are placed in the system domain. These objects typically have restricted access, requiring System or Application Engineer user role, to allow modifications. There are also a set of additional administration structures. All these structures have different purposes and control much of the system behavior.

These additional administrative structures are:

- [Service Structure](#) - holds service definitions.
- [User Structure](#) - gives all user groups and where the users are placed.
- [Node Administration Structure](#) - holds all node groups (sometimes called networks) and nodes defined in the system.
- [Maintenance Structure](#) - defines backups.
- [System Structure](#) - holds the project object.

There are also some additional structures supporting you with additional information:

- [Product Type Structure](#) - can contain certified Industrial IT Products.
- [Product Structure](#) - contains instances of different object types.
- [Documentation Structure](#) - contains system documentation and, if built up during engineering accordingly, the plant documentation.

- **Obsolete Structure**- holds obsolete objects.

Aspect systems can introduce own structures. Examples are:

- **Reuse Design Structure** - defines Reuse Instructions (Reuse Assistant).
- **Equipment Structure** - defines Equipment Groups (Batch Management).
- **Procedure Structure** - defines Procedures (Batch Management).
- **Scheduling Structure** - defines Scheduled / Running Jobs (Information Management).
- **Asset Structure** - provides maintenance personal with the possibility to group/arrange plant assets.

Each structure is implemented as an aspect category and has therefore, for example its own authority control. Every structure with objects has at least one root object. A structure may have multiple root objects, and these are displayed as top-level objects in Plant Explorer.

It is possible that you define your own additional structures to map your special project requirements to the system, refer to the *System 800xA Engineering Engineering Studio (3BDS011223*)*.

Admin Structure

The root of Admin Structure ([Figure 70](#)) is the administrative object's root object. It currently contains two relevant objects, called Domains and Inventory. The Inventory object contains all aspect type objects of the aspect types that implement an admin object as its child. This means that all aspect systems, aspect types, affinities etc. are easily found in one place.

Aspect System Structure

Aspect System Structure ([Figure 71](#)) defines the aspect systems, aspect types and aspect categories available in a system. The structure shows the aspect systems, the corresponding aspect types and under the aspect types their categories. The structure is primarily used to populate the New Aspect dialog used by Plant Explorer and other tools. The structure is typically updated by loading aspect system definitions during the loading of a system extension.

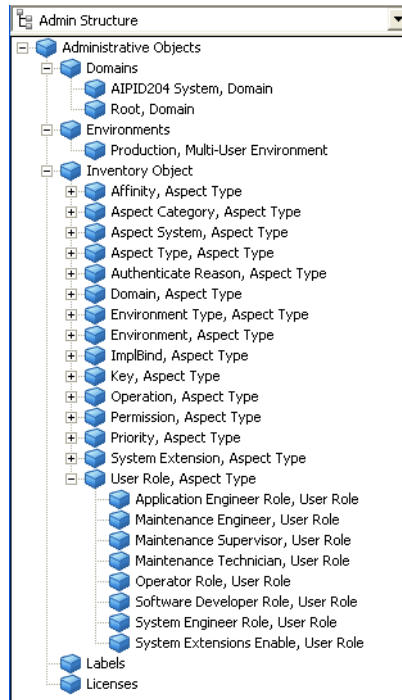


Figure 70. Admin Structure

Asset Structure

Asset Structure provides the possibility for maintenance personal to group and arrange plant assets in the way it is needed for efficient daily work. Control topology constraints can be overcome and even devices with different fieldbus protocols can be grouped together if they require similar maintenance procedures. Asset Structure comes by default with 5 major groups of Assets (Drives, Field Instruments, IT Assets, Motors and Process Equipment).

Control Structure

Control Structure is used to define where the different parts of the control application executes. It is initialized without any objects. After configuration is done, it contains the networks and nodes that are used to control the process, that is

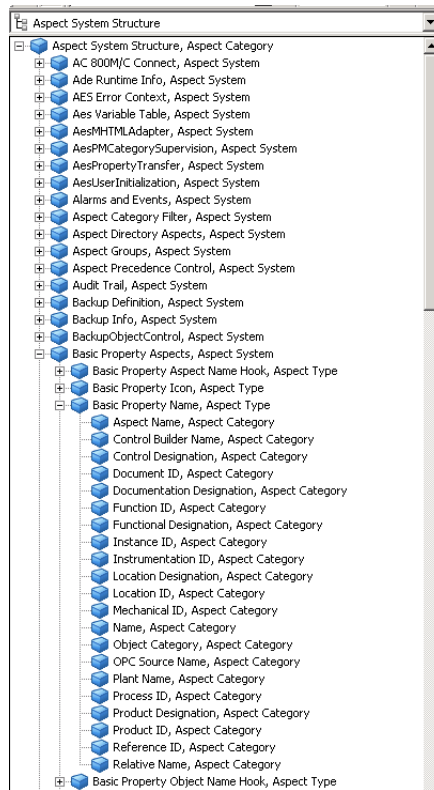


Figure 71. Aspect System Structure

the equipment of the system and is also used to organize the process control environment in the system: Control networks, control projects, applications, controllers, stations, different types of sub-objects (like I/O boards and process objects as well as signals assigned to different objects within this structure).

Control Structure is the base for the control operation of the system. The user can, for example, check consistency and system status.

By configuring the objects with Control Designation aspects you can build and display control designations according to reference designation system standards in this structure.

Figure 72 below gives an example of how the structure could look for AC 800M after having done some configuration work on a control project.

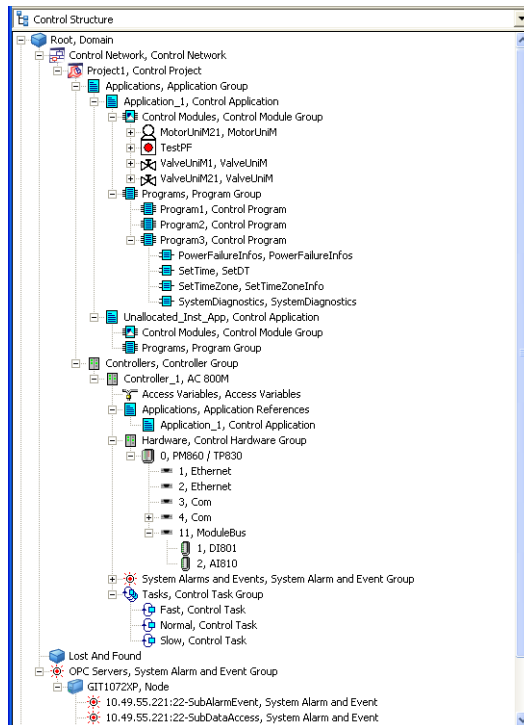


Figure 72. Control Structure for AC 800M

The Lost and Found object collects useless alarms etc. An object under the Lost and Found object is removed in the same way as another object.

The objects:

- System Alarms and Events and their sub-objects.
- OPC Servers and their sub-objects.

are shown in Control Structure automatically for internal Control Builder M administration purposes.

Documentation Structure

The structure should be used for storage of project and plant documents. Build an object structure and add relevant documents to the objects as aspects.

By configuring the documentation objects with Documentation Designation aspects you can build and display document designations according to reference designation system standards in this structure.

Equipment Structure

Equipment Structure is used to organize Batch Management equipment into Equipment Groups.

Functional Structure

Functional Structure describes the functionality of the plant. It is used to structure the plant into systems and subsystems according to how the plant equipment operates together.

By this it forms a base for the identification of reusable control solutions and can also be used to organize displays, alarm sectioning and other functions that are related to the structure of the plant.

It can be used to display functions divided into several different control systems. It can also display aspect objects that are not directly related to any specific control logic, like a heat exchanger (the physical object) included in a control loop.

By configuring the objects with Functional Designation aspects you can build and display functional designations according to reference designation system standards in this structure.

The structure is initialized without any objects. [Figure 73](#) shows an example of how Functional Structure can look when various types of objects have been added.

It is reflected in the Toolbox in Graphics Builder, and also in the corresponding wizard page.

The structure is updated when generic elements are created or modified, or when new primitives are loaded or unloaded.

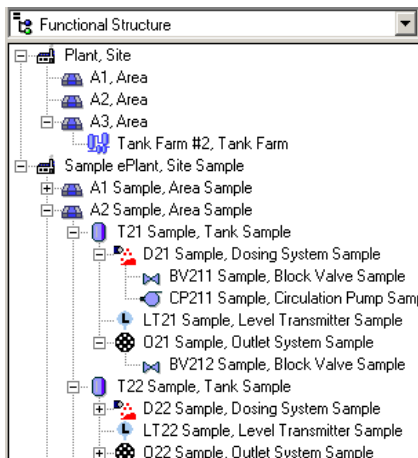


Figure 73. Functional Structure

Graphics Structure

Graphics Structure (Figure 74) holds the graphic primitives and subelements available in the system. This structure is typically not accessed nor modified by a user as it is updated by different tools.

It contains one root object, Graphics Tools. The root object then holds three objects, one holding the subelements and one holding the primitives. The third object holds the other graphical elements.

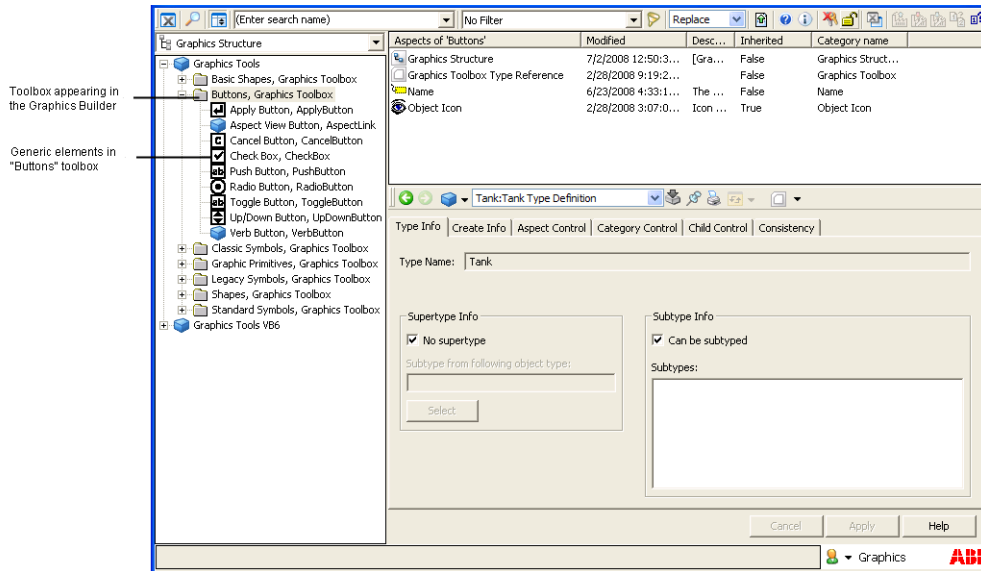


Figure 74. Graphics Structure

Library Structure

Library Structure is used to categorize and store reusable entities beyond object types.

It contains all object type libraries and its versions. Most of the work with libraries can be made in this structure, such as connect used libraries to a library, creating new libraries, create a new version for a released library, change the library state, and archive a library. The object types of these libraries are stored in [Object Type Structure](#).

Library Structure ([Figure 75](#)) also holds the alarm and event list configurations, the view classes, the tools, the workplace frames, and the workplace panels. The alarm and event list configurations can be shared by a number of lists, and define the layout and filter parameters of the lists. Each object must contain one and only one

configuration. The configurations are listed in the Config View of the Alarm and Event List aspect category.

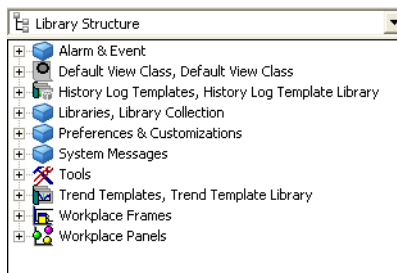


Figure 75. Library Structure

Location Structure

Location Structure is supplied to help the user identify the geography or location of the objects in the plant. This structure must be built with the knowledge of what buildings, rooms and locations that exist in the site. At system start-up this structure contains no objects.

By configuring the objects with Location Designation aspects you can build and display location designations according to reference designation system standards in this structure.

Figure 76 gives an example of how the structure could look when containing objects.

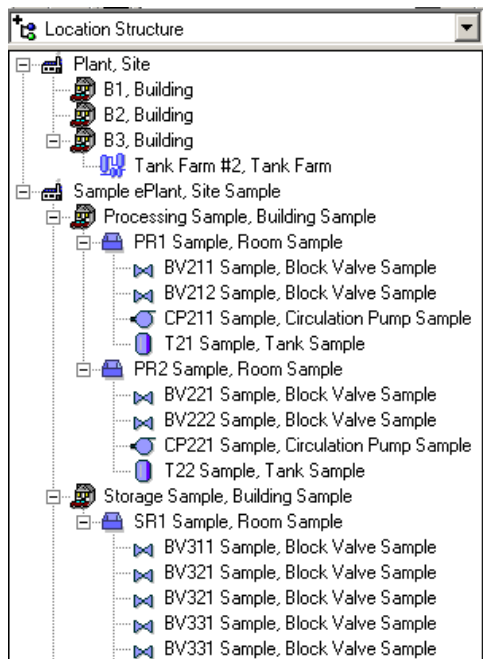


Figure 76. Location Structure

Maintenance Structure

Maintenance Structure (Figure 77) is used to define and work on Backups and System Configuration Versions.

Node Administration Structure

Node Administration Structure (Figure 78) is used for node administration purposes. The main reason for this structure is that all nodes in the system must be found in at least one place and that nodes can be placed in multiple groups. The nodes in Node Administration Structure are typically also found in Control Structure.

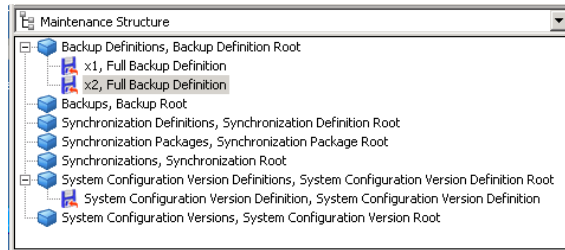


Figure 77. Maintenance Structure

The structure contains two levels under the root: The first level contains node groups and the second level contains nodes. The node group named **All Nodes** has a prebuilt support that disallows manual changes and updates automatically with all nodes in the system.

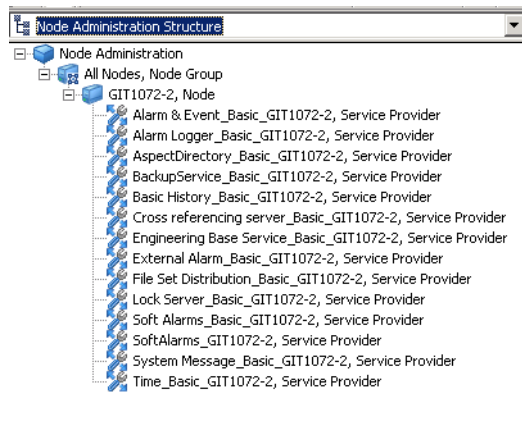


Figure 78. Node Administration Structure

Object Type Structure

Object Types act as types when you create objects. Almost all aspect objects are instances of an Object Type. In the Object Type Structure (Figure 79), the object types defined in the system are organized in Object Type Groups or Library

Versions. For example you find in the Object Type Structure the Library Versions and the object types brought into the system by the AC 800M Connect system extension.

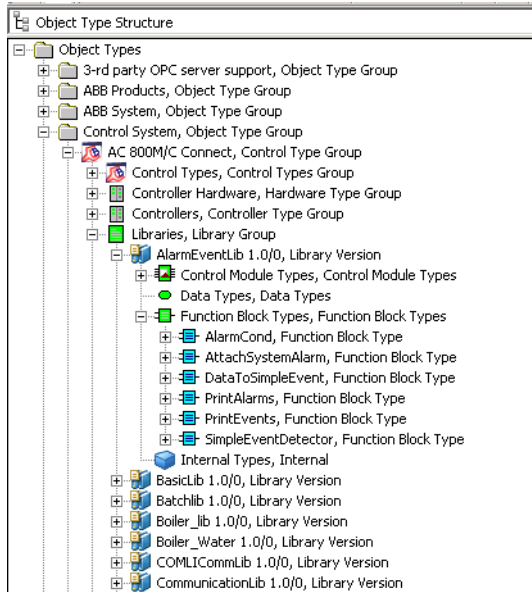


Figure 79. Object Type Structure

Read more about Object Types in [Object Types](#) on page 310 and in subsection on Building Object Types in the *System 800xA Configuration (3BDS011222*)*.

Obsolete Structure

Obsolete Structure contains information about aspect objects that have been deleted by a system function, for example the up-loader in a connectivity package.

An aspect object with the system function name, appended with the time when delete operation was performed, represents each delete operation.

The aspect object has an Obsolete Object aspect, and from the Main view you can open an Import/Export file, which contains the deleted objects and aspects. Deletions made by the user are not logged in this structure.

Procedure Structure

Procedure Structure is used to contain Batch Management receipt procedures, unit procedures, operations, exception procedures, receipt phases, and functions.

Product Type Structure

Product Type Structure contains all certified Industrial IT Products organized according to their Product Suite.

Product Structure

Product Structure contains instances of different product types. The instances contain product information that are specific for an individually delivered product, e.g. serial numbers, test records etc.

Reuse Design Structure

Reuse Design Structure is a tool specific structure created and used by Reuse Assistant. You build Reuse Instructions in this Structure using the Reuse Assistant Architect part of Reuse Assistant.

Scheduling Structure

Scheduling Structure is used to hold Information Management job descriptions of scheduled jobs and running jobs.

Service Structure

Service Structure holds the services, their groups and providers, and the relations between these defined in a system.

A service is a global function in the network, and the objects and aspects in the Service Structure ([Figure 80](#)) control the definition and deployment of the services.

System Structure

System Structure only contains a project object, representing the whole project you are working on and some settings for it.

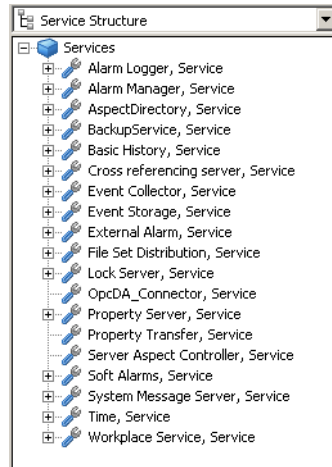


Figure 80. Service Structure

User Structure

User Structure (Figure 81) holds the defined users and user groups allowed to work in the system.

All users must have a related Windows user account. The user aspect object primarily contains information about a user and what the user is allowed to do within the system. It also contains data specific to the user, such as profile configuration values.

The structure contains two levels under the root object: The first level holds groups, and the second holds users. The user group **Everyone** holds all users. Users can be added to groups by dragging and dropping them to different user groups.

User Roles

User Roles adapt your environment depending on your role. You assign user roles when you add a user to a specific Industrial IT group.

The following User Roles exist:

- **Operator** - can control the process.

- **Application Engineer** - can do typical engineering operations.
- **System Engineer** - can do system administrative tasks.

User roles help to customize the environment to suit the needs of each user. This means that the operations the user needs to act on are visible.

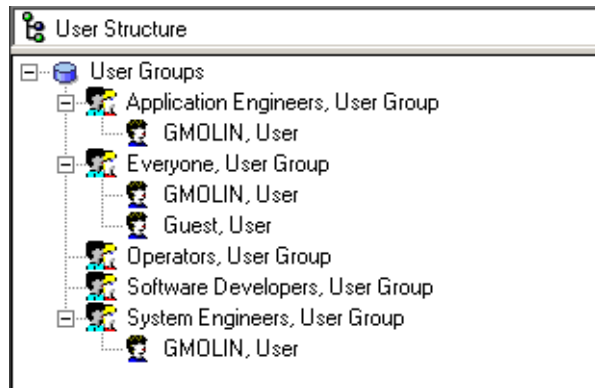


Figure 81. User Structure

The Industrial IT groups may be associated to the corresponding Windows groups. The Windows groups are used for the Security settings. Refer to the *System 800xA Administration and Security (3BSE037410*)* for details.

Workplace Structure

Workplace Structure (Figure 82) holds the workplace objects defined in a system. The structure is used by the users to organize the workplaces in use. A workplace lower in the structure inherits definitions from higher level workplaces.

The workplace objects types of aspect that can be placed here are numerous, and they control the layout and function of a workplace, for example tool bars and menu items.

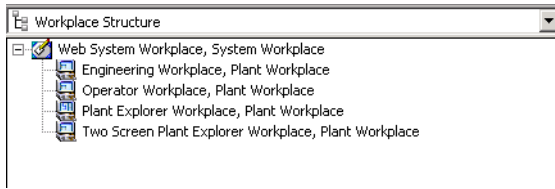


Figure 82. Workplace Structure



Only workplace objects are allowed in this structure.

Aspect Object Architecture

The Aspect Object architecture divides the system topology as shown in [Figure 83](#) into separate functional layers for control, server, and workplace functions respectively. The architecture deals with the workplace and server layers, and defines how the control layer is interfaced. Understanding these layers is essential to understanding the Aspect Object architecture.

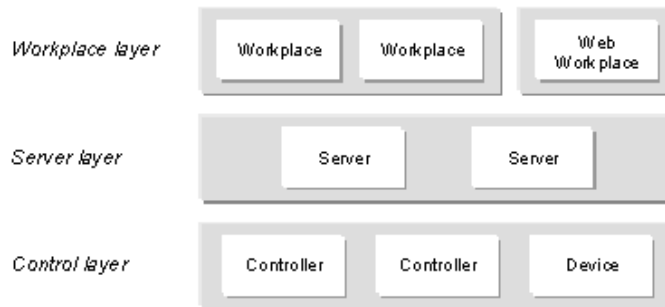


Figure 83. Functionality Layers

Two main concepts are central to the Aspect Object architecture: the concept of Aspect Objects, and the concept of 800xA Services. The Aspect Framework implements these concepts on the Microsoft Windows operating system platform.

Functionality in the server and workplace layers is provided by software components referred to as system applications, or just applications. An important feature of System 800xA is that information and functions are centered on Aspect Objects. To participate in Aspect Object operations, an application must present itself as an aspect system, or possibly as several aspect systems. In essence this means that the application provides COM objects called aspect system objects, which support certain framework-defined interfaces, through which the application can initiate and participate in common operations on objects and aspects.

Functionality in the control layer is connected to the server layer through connectivity components. A connectivity component provides Aspect Object types, access to real time data, and various forms of supporting functionality for different types of controllers and devices.

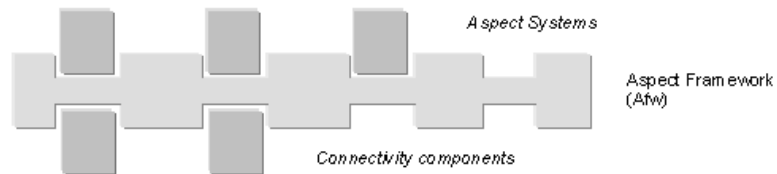


Figure 84. The Afw Aspect Framework

Client/Server Model

Applications are implemented as client applications or services. A service is an entity that provides a certain set of functions in the system. Services run in the server layer. Client applications are applications that utilize the functionality provided by one or more services, e.g. to present some information to a user.

Services must conform to the 800xA Service model. An 800xA Service is partitioned into several service groups, each group handling part of the scope of the service (e.g. part of the object space). For redundancy each group can contain several service providers running on different servers. A service manager initiates and supervises the execution of Afw services.

An 800xA Service provides a service handler. This is a COM object that a client includes and runs as an in-process object to access the service. Using information from the Aspect Directory, the service handler provides one uniform interface to the

service, relieving the client from having to deal with issues such as how the service is partitioned, redundant service providers, etc.

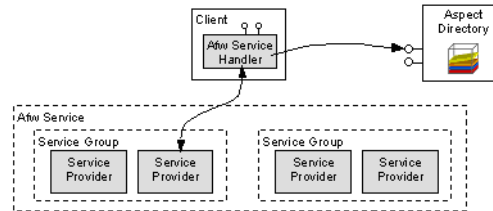


Figure 85. 800xA Service Model

800xA OPC

The OPC (OLE for Process Control) specification is a non-proprietary technical specification that defines a set of standard interfaces based on Microsoft's COM technology. The purpose of the OPC standard interface is to facilitate interoperability between automation and control applications, field devices, and business and office applications.

Traditionally, each application developer was required to write different custom interfaces to exchange data with different data sources. OPC eliminates this need by defining a common interface that permits this work to be done once. The OPC server can be used by many different applications. Different OPC servers are available for different brands of controller, fieldbusses, and other data sources.

OPC plays a vital role in the Aspect Object architecture. It is through OPC compliant interfaces that applications have access to real-time data, historical data, and alarm and event data from controllers, field devices, and other software applications.

The Afw OPC server concept allows many different OPC servers to be placed under a one common server that unifies access to data from different sources. This common server provides one set of OPC compliant interfaces, eliminating the need for client applications to know which OPC server to use for each data item.

There are three 800xA OPC servers: 800xA OPC/DA for real-time data access; 800xA OPC/HDA for access to historical data; and 800xA OPC/AE for access to alarm and event data.

The Afw OPC Servers are designed as Afw Services, and thus consist of two parts:

- An 800xA service handler that runs as an in-process COM object with the client application, and provides a unified OPC interface to all data sources in the system.
- Server components that are accessed by the client component, and perform the actual access to data provided by different data sources.

Aspect Systems

To participate in Aspect Object operations, an application must present itself as an aspect system, or possibly as several aspect systems. Aspect systems provide the functionality that is defined for Aspect Objects. Examples are Control, Graphics, Alarm & Event, History, Reports, Documentation, Simulation, Asset Optimization, Material Tracking, Production Scheduling, etc.

An aspect system provides implementations for one or several aspect types through COM objects, referred to as aspect system objects (ASO). These objects interact with the Aspect Framework through different sets of framework-defined interfaces for common object and aspect operations. ASOs may also expose object properties through OPC.

The Aspect Framework provides COM representations of Aspect Objects and aspects. Aspect systems provide their functionality through aspect system objects. Users identify and access aspect system objects through Aspect Objects and Aspects.

Each aspect system is responsible for storing and maintaining its own data. However, in many cases data must be shared among a group of aspect systems. Unless handled in a correct way, this may cause data consistency problems. The Aspect Object architecture offers two ways to avoid that:

- A shared data item is stored in only one copy by one aspect system, and other aspect systems access it from there, through framework defined interfaces.

- Several aspect systems hold their own copies of a shared data item. When an aspect system updates its copy, it must inform the framework, which in turn informs other aspects systems to update their copies.

An aspect system may provide one or more user interfaces, implemented in a suitable technology.

Many aspect systems provide some degree of configurability, typically by means of some form of configuration tools. To ensure that engineering can be done in an efficient and consistent way, there are certain rules defined for how aspect systems shall handle and share configuration data.

Different aspect systems can be more or less well integrated into the system. To be able to easily describe how well integrated an aspect system is, different integration levels are defined. Depending on which integration level is selected, the aspect system objects must support different sets of framework-defined interfaces for common object and aspect operations.

System Status

Aspect Objects are used to model not only plants, equipment, products, processes and procedures, but also the 800xA System itself, including all its hardware and software components. The status of these system parts is modeled as properties of the corresponding Aspect Objects. To make it possible to present status information for all kinds of Aspect Objects in a structured and uniform way, these object properties and the way they are provided must follow certain rules.

The System Status framework includes interface definitions and guidelines for how system status information shall be provided, and a System Status Viewer aspect system for presentation of system status to a user.

Two levels of system status properties are defined: basic, and additional.

The System Status Viewer aspect system presents an overview of the system status as a tree structure of Aspect Objects, based on any of the object structures, with status indications for those objects that provide system status.

An Aspect Object can provide a summary status for subordinated objects. This is referred to as propagated status. For example, an object representing a controller can show a summary of the status of all its I/O units. The aspect system that provides

system status for the top object, i.e. the controller in this example, is responsible for computing the propagated status.

Applications that provide system status information should also report the occurrence of a fault either directly to the system message service or as an OPC/AE event.

Besides system status properties, an Aspect Object can also provide further object specific status details through specific aspect views. Aspects that provide system status details have the key System Status Details. The System Status Viewer user interface presents a Show Details link to such aspects.

Appendix B Structured Plant Modeling

Introduction



Structured modelling of a plant can be done with or without using reference designations.



For explanation purposes this section uses a collection of example object types called Basic Object Types. You can build a corresponding set of own objects types in a user defined library to support structured plant modelling in your projects according your specific needs.

Functional Relations Between Aspect Objects

The functional relation between aspect objects are presented in a function oriented structure. This structure is built up by placing aspect objects into this structure according to their hierarchical relations.

The following example in [Figure 86](#) shows a number of aspect objects representing a Storage Tank with its drain system and quantity control.

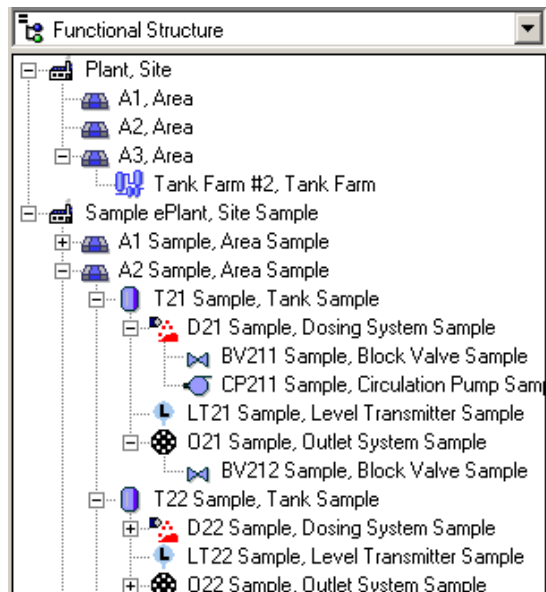


Figure 86. Functional Relation of a Storage Tank

Basic Object Types for Functional Structure

According to standards, it is recommended to build the levels within Functional Structure according to [Figure 87](#).



Device (Equipment Module, Control Loop, Tag) in this context does not mean just the apparatus. It includes also the related I/O, control and Man Machine Interface (MMI) parts. So a better term might be **Device Loop**.

The alternative terms for Device listed in brackets may be used within projects for chemical industry.

The Basic Object Types ([Figure 88](#)) include a set of object types, which can be used to build the basic Functional Structure according to that recommendation.

The Object Types are organized within the **Object Type Structure** window according to the proposed levels.

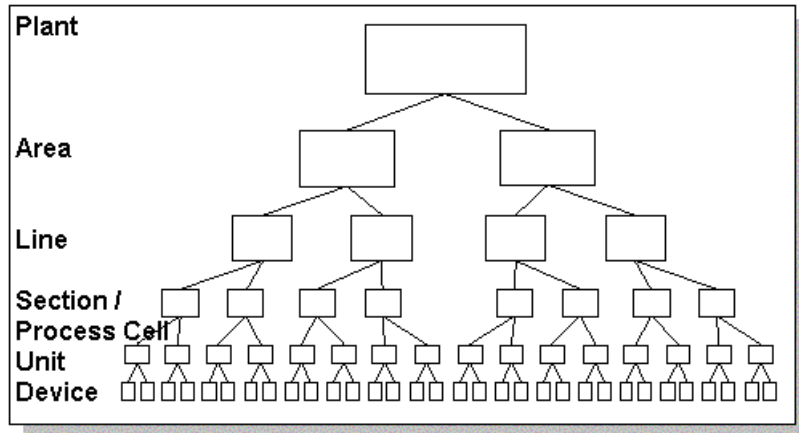


Figure 87. Hierarchy Levels within Functional Structure

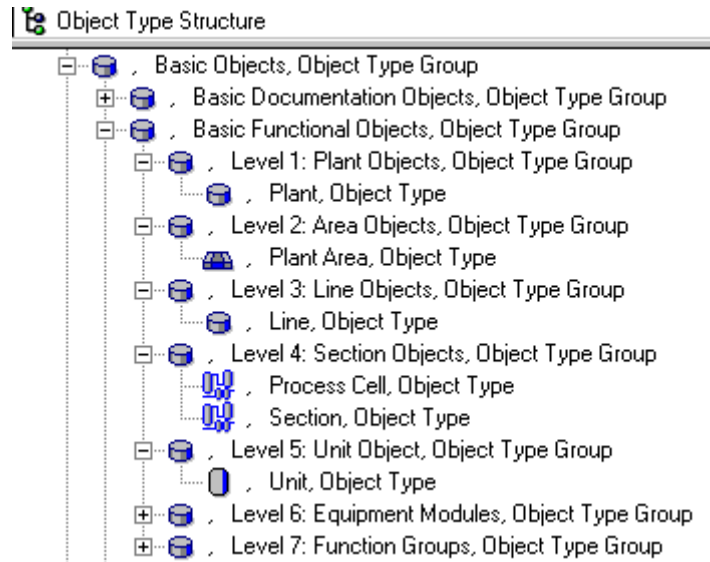


Figure 88. Basic Functional Object Types

Those object types provide the functionality:

- Include sensible icons.
- Include **Functional Designation** aspect.
- Propose sensible object types according to level definitions when creating new children objects.

If those basic object types are not sufficient it is possible to:

- Add your own, project specific object types (for example by copy).
- Derive your own, project specific object types from those by using the **Supertype Reference** mechanism.

Usage of Function Reference Designations

Before building the basic Functional Structures it should be clarified, how to identify objects unambiguously by function reference designations.

Reference designations identify (in opposite to names) the objects in hierarchical way according to international standards and allow easy navigation within big structures.

The function reference designation shall identify objects according to its position within Functional Structure.

Levels are optional and can be skipped, if not relevant.

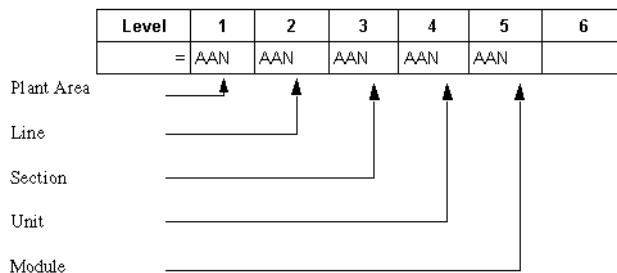


Figure 89. Reference Designations within Functional Structure

Creation of Objects in Functional Structure

To create the top object within a structure:

1. Right-click into the white space of **Functional Structure** area.
2. Select **New Object...** from right mouse button menu.
3. Navigate to plant level object types within **New Object** dialog.
4. Enter name of object to be created.
5. Click **Create**.

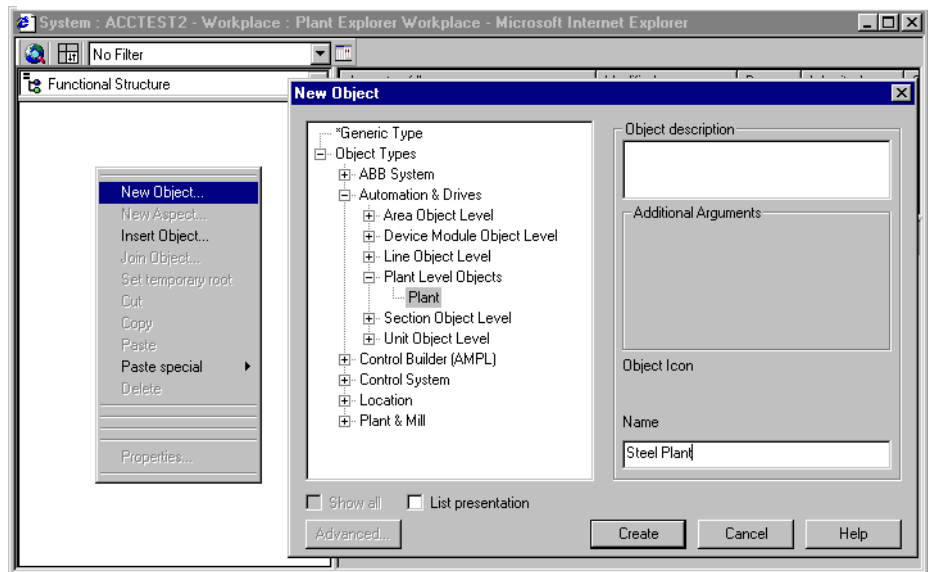


Figure 90. New Object of Type Plant

To create objects in subordinated level:

1. Right-click on object, which serves as parent.
2. Select **New Object...** from right mouse button menu.

3. Select object type from offered list.



Object types are normally configured in that way, that only the most sensible object types are offered when creating new objects. Nevertheless in most cases it is possible to select other object types by enabling the **Show all** check button with **New Object** dialog.

4. Enter name of object to be created.
5. Click Create.
6. Enter function reference designation:
 - Select **Functional Designation** aspect.
 - Enter value in **Name** property.
 - Press **Apply**.

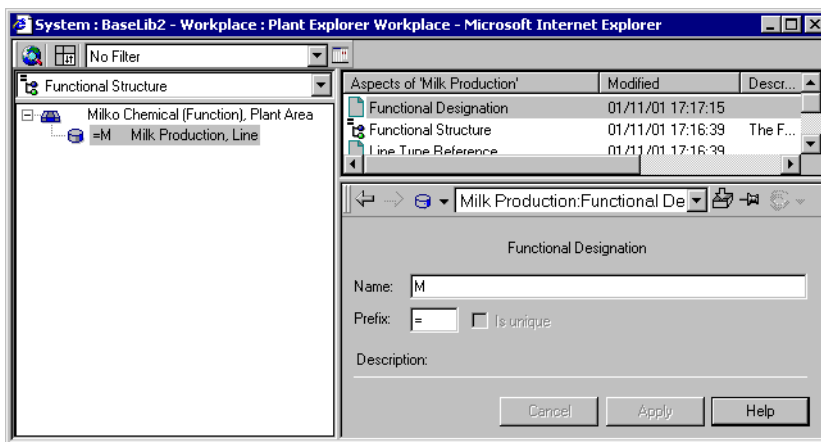


Figure 91. Enter Functional Designation of an Object

7. Continue in that way to complete basic Functional Structure.

A basic Functional Structure you can easily specify and load into the system using an Excel spreadsheet configured for Bulk Data Management, refer to [Figure 92](#). Bulk Data Management enables you to save manual object by object instantiation work.

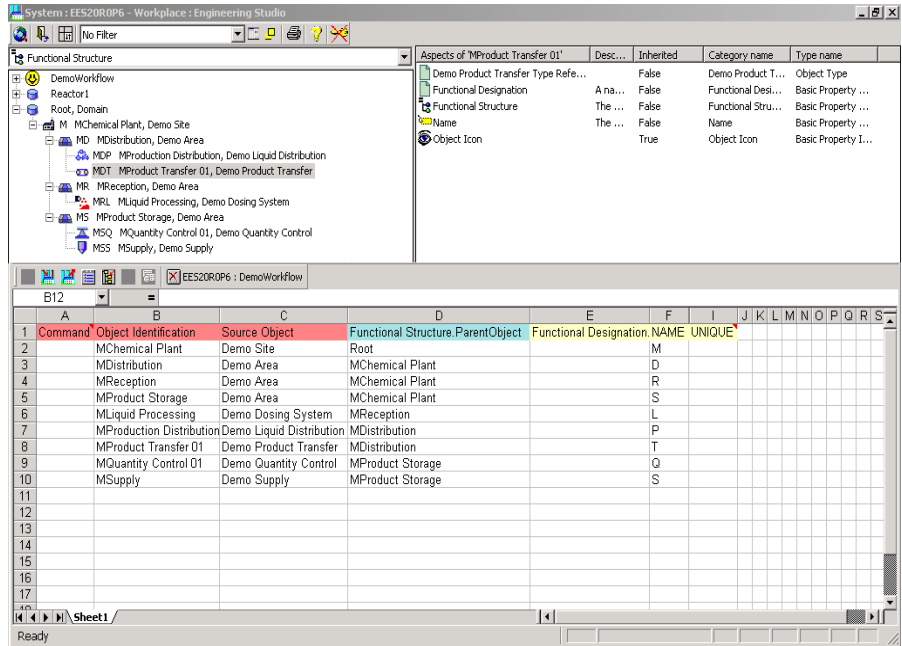


Figure 92. Building Basic Functional Structure with Bulk Data Manager

Relations Between Aspect Objects of the Control System

The relations between aspect objects representing the control system are presented in Control Structure. The structure is built up by creating/placing the relevant aspect objects into this structure.

The basic structure of the control system is specified and broken down into networks, nodes, buses and stations.

Note that as the highest levels also Control System and Plant Network could be introduced additionally.

Basic Object Types for Control Structure

Figure 93 shows general generic structuring recommendations for Control Structure.

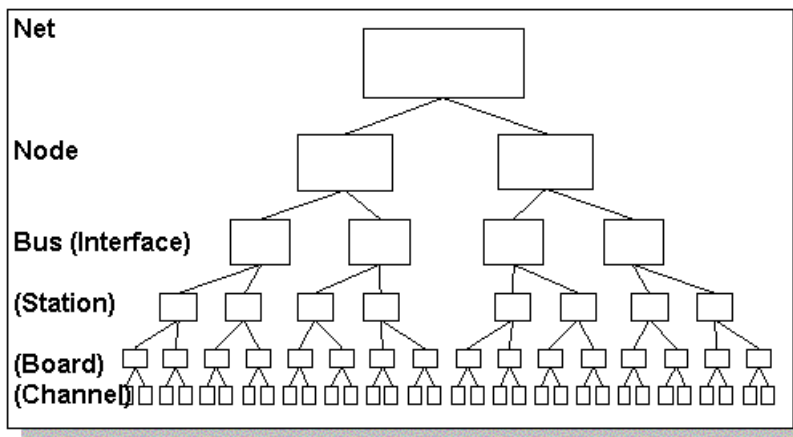


Figure 93. General Generic Hierarchy Levels within Control Structure

In the concrete case the levels depend on the target control system, by default AC 800M.

The AC 800M Connect Object Type Libraries that can be loaded for each system include a set of object types, refer to [Figure 94](#), which can be used to build Control Structure in accordance with the rules imposed by the AC 800M controller configuration with Control Builder M Professional and similar to the recommendation in [Figure 93](#).

Usage of Control Reference Designations

Before building the basic Control Structure it should be clarified, how to identify objects unambiguously in this structure. Reference designations identify (in opposite to names) the objects in hierarchical way according to international standards and allow easy navigation within complex structures. Refer to [Figure 95](#).

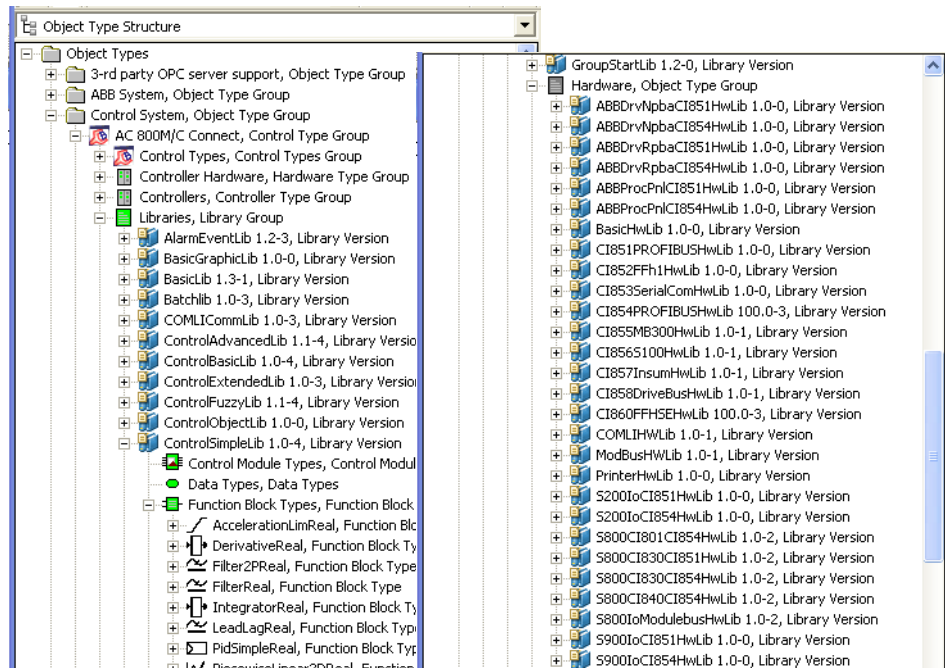


Figure 94. AC 800M Connect Object Types for Control Structure

The control reference designation shall identify objects according to its position within the Control Structure. Levels are optional and can be skipped, if not relevant.



If you do signal engineering (for example following the Functional Planning approach) you make use of the engineering features provided by the AC 800M Signal Extension system extension / object type group CBM_Signals and the corresponding I/O-Allocation function (both delivered with installation package Engineering Studio). In this case you have to specify the relative control designations in Control Designation aspects on Controller (Node), Hardware Unit (Interface/Board) and Signal (Channel) object level.

Control Builder M Professional itself does not need control designations in Control Designation aspects. But it uses designation numbers in the Name aspect and the corresponding path for identification. The CBM_Signal objects the corresponding I/O-Allocation function take care of the mapping between Control Designation and Name aspect.

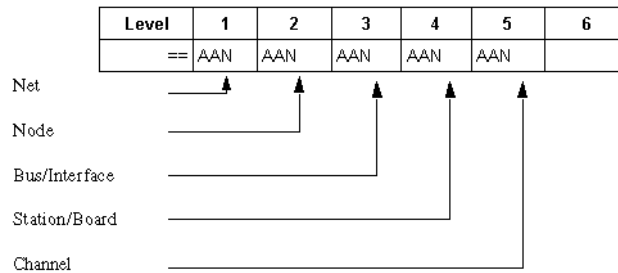


Figure 95. Reference Designations within Control Structure

Creation of Objects in Control Structure

Creation of objects in Control Structure can be performed similar as shown for Functional Structure, taking care of the corresponding object types and filling in the Control Designation aspect.

The Basic Object Types only include object types representing Control System and Plant Network, which are the (optional) levels above the levels shown in [Figure 96](#).

These additional levels are covered by object types loaded with the AC 800M Connect system extension, refer to [Figure 94](#). The hierarchy below and including Control Network in Control Structure is predefined by AC 800M Connect and Control Builder M Professional.

Relations Between Functional Structure and Control Structure

A proposal how to build up Functional Structure in the context of a 800xA system with AC 800M connectivity is shown in [Figure 97](#). How it maps to Control Structure is shown in [Figure 98](#).

In Functional Structure you see

- Relation within a function.
- Relation between functions.

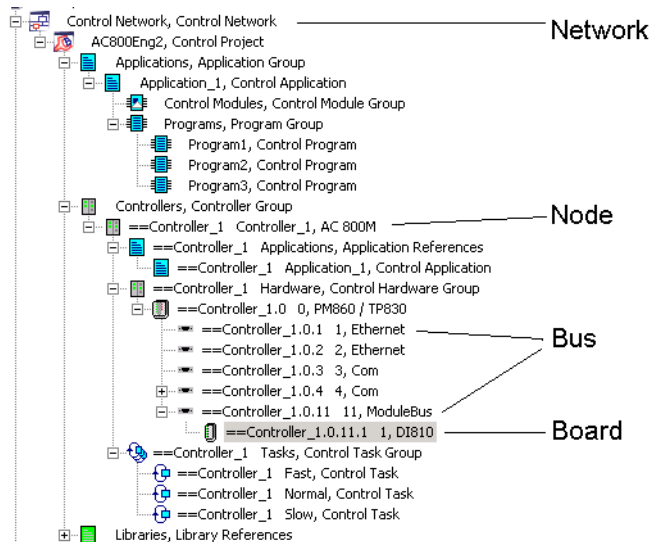


Figure 96. Basic Control Structure Defined by AC 800M Connect

- Relation between Control Module/Function Block and Hardware IO.
- Relations between Control Objects and other Aspect Objects having Graphic Displays.
- Relation between Control Module/Function Block and Variables (Only when using Function Designer).

In Control Structure you see:

- Relations within an Application.
- Structuring of the program code.
- Controller hardware structuring.

Location Relations Between Aspect Objects

Location relations between aspect objects are presented in a location oriented structure. Distinguish between topographic locations like Areas, Buildings, Rooms and mounting locations like Cabinets, Desks, Racks, Slots.

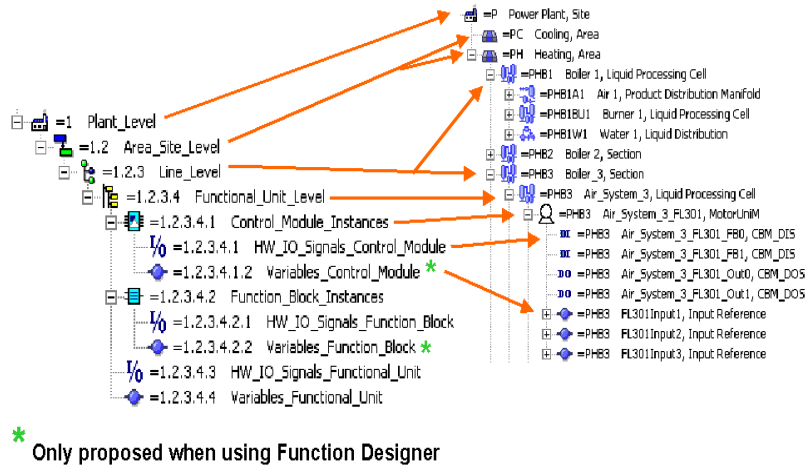


Figure 97. Proposed Levels of Functional Structure

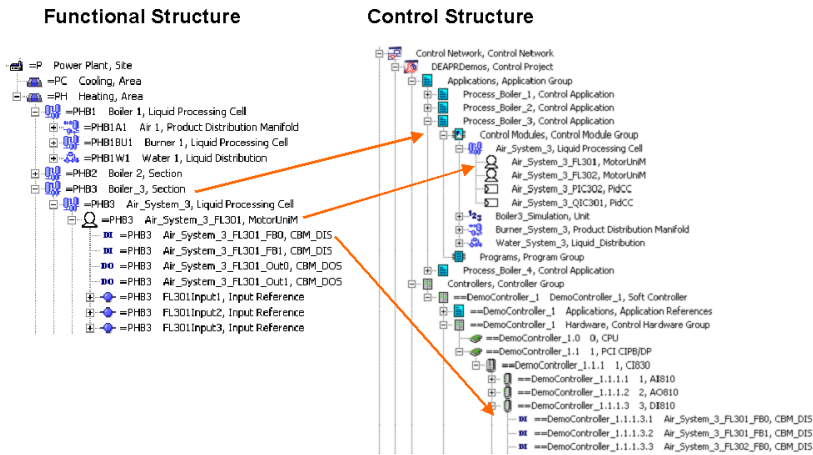


Figure 98. Functional Structure Mapped to Control Structure

The topographic location structure and the mounting location structure can be created independently from each other and connected in a later stage. The connection is performed by placing mounting location structure trees into the

topographic location structure. The mounting location structure contains a number of aspect object structure trees that represent, for example, the cabinets, desks and other units of the plant.

Figure 99 shows the principles of a location oriented structure.

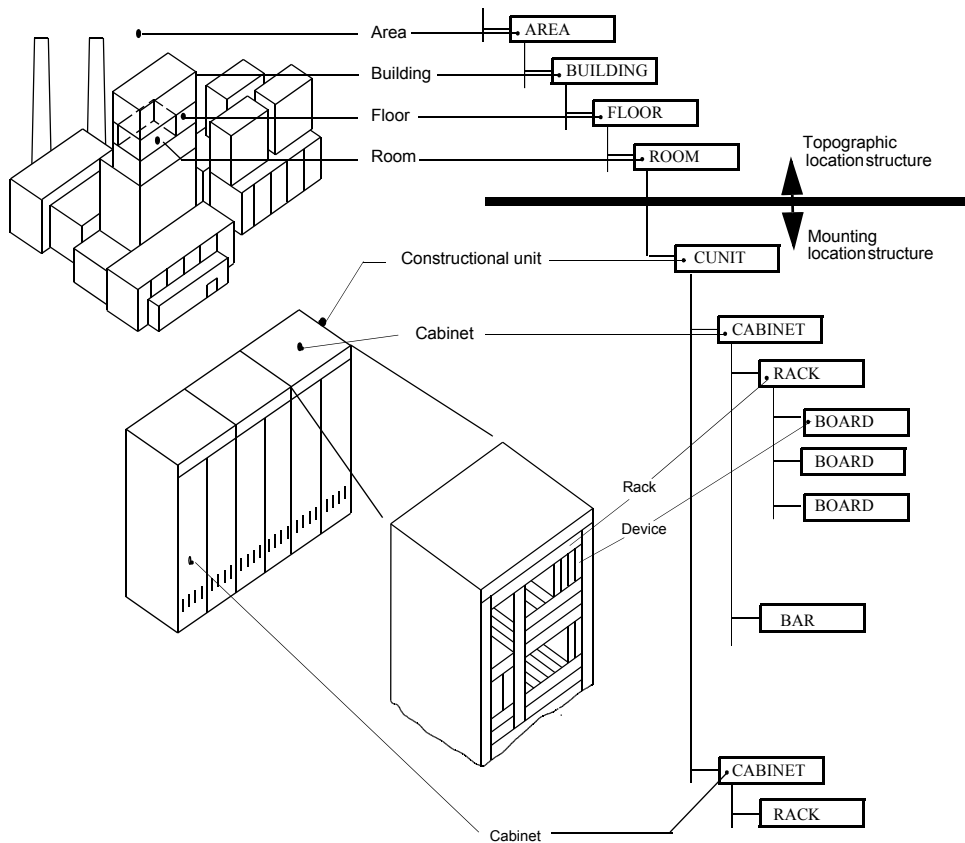


Figure 99. Principle of a Location Oriented Structure

The basic Location Structure in this context includes at least the topographic locations like buildings, floors, rooms, etc. It should be detailed down to the level that is necessary for clarification in the current stage of project work.

If available, topographic layout diagrams and descriptions are added to the objects.

Basic Object Types for Location Structure

Figure 100 shows the topographic location levels proposed to be used.

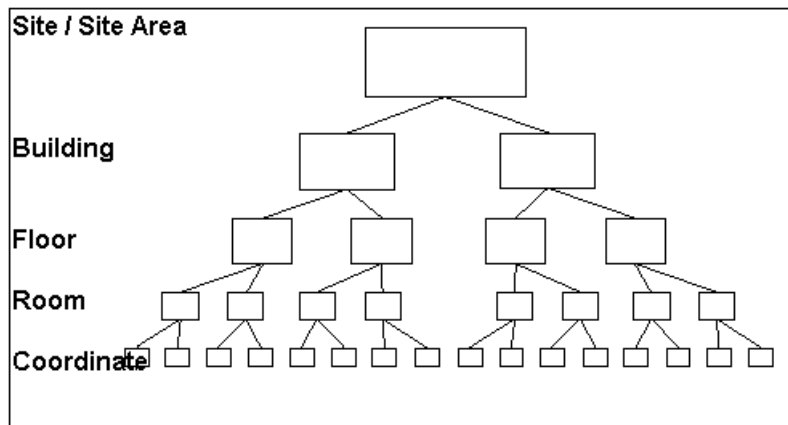


Figure 100. Hierarchy Levels within Location Structure

The Basic Object Types include a set of object types, which can be used to build the basic Location Structure according to that recommendation.

For easy understanding and navigation those Object Types are also organized within the **Object Type Structure** window according to those levels. Refer to Figure 101.

Those object types provide the functionality:

- Include sensible icons.
- Include **Location Designation** aspect.
- Propose sensible object types according to level definitions when creating new children objects.

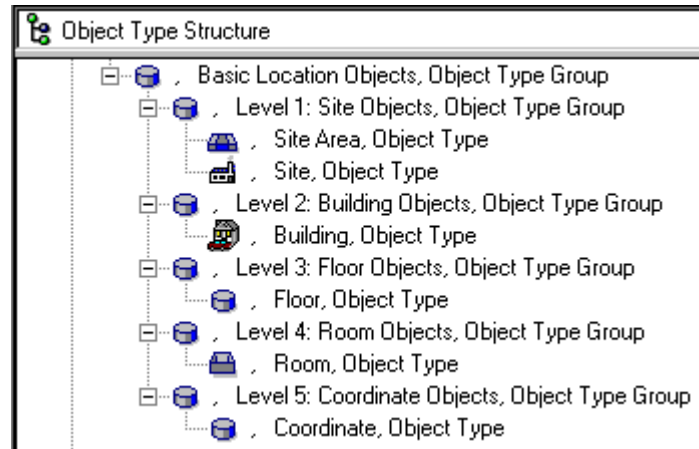


Figure 101. Basic Location Object Types

If those basic object types are not sufficient it is possible to:

- Add your own, project specific object types (for example by copy).
- Derive your own, project specific object types from those by using the **Supertype Reference** mechanism.

Usage of Location Reference Designations

Before building the basic location structures it should be clarified, how to identify objects unambiguously by location reference designations.

Reference designations identify (in opposite to names) the objects in hierarchical way according to international standards and allow easy navigation within big structures.

The location reference designation shall identify objects according to its position within Location Structure.

Levels defined within the location reference designation are optional and can be skipped, if not relevant. Refer to [Figure 102](#).

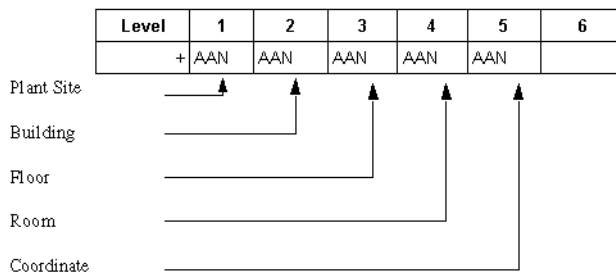


Figure 102. Reference Designations within Location Structure

Creation of Objects in Location Structure

Creation of objects in Location Structure (Figure 103) can be performed similar as shown for Functional Structure, taking care of the corresponding object types and filling in the Location Designation aspect.

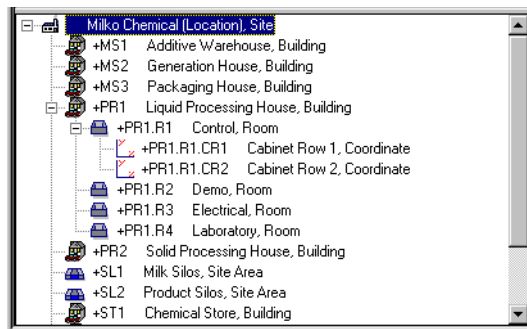


Figure 103. Completion of Location Structure

Relations Between Aspect Objects for Documentation

The documentation of the project is modeled by folder/binder and sub folder/grip/package objects and document objects in Documentation Structure.

Each level is identified by an absolute documentation reference designation and/or object name.

The document objects either are Aspect Objects containing one or more aspects holding documents or are placed Aspect Objects from other structures holding also one or more aspects holding documents.

Basic Object Types for Documentation Structure

Figure 104 shows the documentation structure levels proposed to be used.

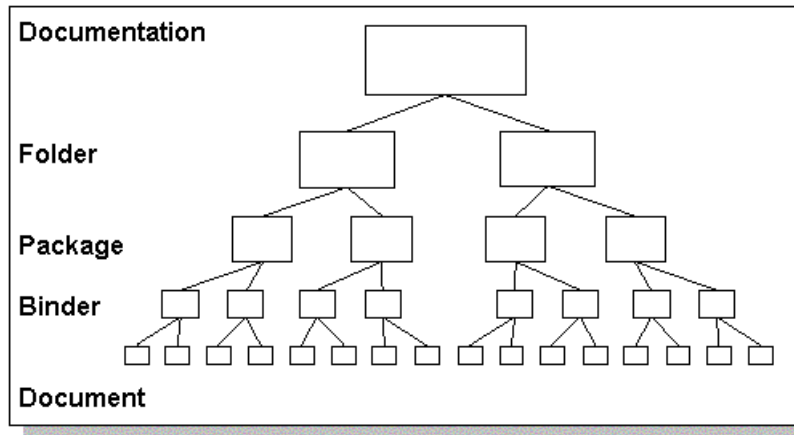


Figure 104. Hierarchy Levels within Documentation Structure

The Basic Object Types include a set of object types, which can be used to build the basic Documentation Structure according to those recommended levels.

For easy understanding and navigation those Object Types are also organized within the **Object Type Structure** window according to those levels. Refer to Figure 105.

Those object types provide the functionality:

- Include sensible icons.
- Include **Documentation Designation** aspect.

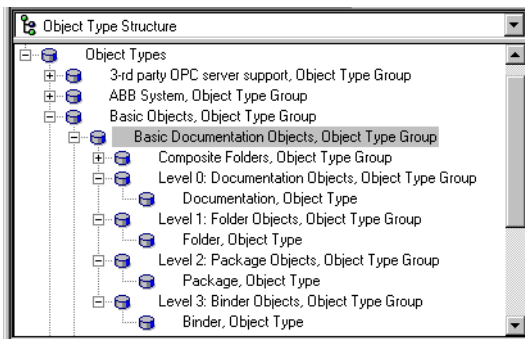


Figure 105. Basic Documentation Object Types

- Propose sensible object types according to level definitions when creating new children objects.

If those basic object types are not sufficient it is possible to:

- Add own, project specific object types (for example by copy).
- Derive own, project specific object types from those by using the **Supertype Reference** mechanism.

In addition to those simple object types also some predefined document folder object types according to ABB internal standard definitions are provided as composite object types (Figure 106).

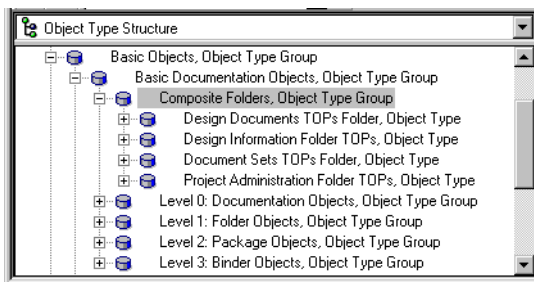


Figure 106. Composite Folders Object Types

Those are ready made folders including:

- Folder names and reference designations.
- Package definitions including package names and reference designations.
- Binder definitions including binder names and reference designations.

Usage of Documentation Reference Designations

Before building the basic documentation structures it should be clarified, how to identify objects unambiguously by Documentation reference designations. The documentation reference designation shall identify objects according to its position within Documentation Structure.

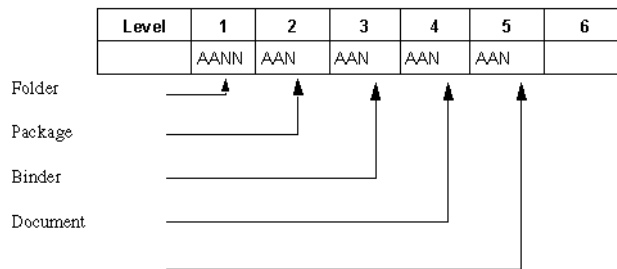


Figure 107. Reference Designations within Documentation Structure

Creation of Objects in Documentation Structure

Creation of objects in Documentation Structure can be performed similar as shown for Functional Structure, taking care of the corresponding object types and filling in the Documentation Designation aspect.

To get ready made folders according to a typical definition you should choose the object types:

- Project Administration Folder.
- Design Information Folder.
- Design Documents Folder.

- Document Sets Folder.

In this way the basic Documentation Structure (Figure 108) can be created in an efficient way.

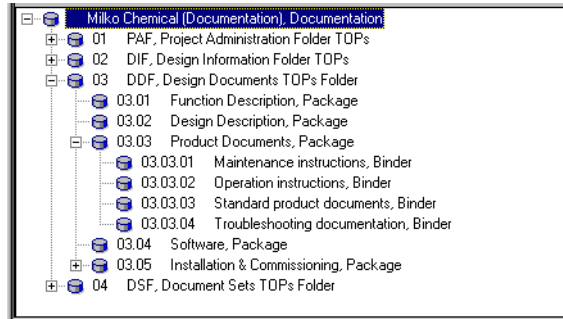


Figure 108. Completion of Documentation Structure

Appendix C Designation System

Introduction

The following example of a designation system only describes rules for unequivocal creation of a Plant / Function designations of a plant. Intention of this example is to highlight important parts of a full designation system.



Note that eventually all designation kinds listed in [Table 35](#) are required for electrical products and signals of a plant.

In general a designation shows the relations between products in the plant and documentation.

Table 35. Designations for a Plant

Prefix	Designation
=	Plant / Function Designation
-	Product Designation
+	Location Designation
:	Terminal Designation
#	Signal Designation

The example designation system is based on the following standards:

IEC 1175

IEC750 = DIN40719, Part 2

DIN19227, Part 1

Function / Plant Designation (=)

Pos.	1	2	3	4	5			6	7	
=	A	A	A	A	N	N	N	.	A(A)	N(N)

- (1) Plant ([Table 36](#))
- (2) Main function ([Table 37](#))
- (3) Sub function ([Table 38](#))
- (4) Technology function group ([Table 39](#))
- (5) Sequential number of technology function group
- (6) Delimiter
- (7) Documentation Group ([Table 40](#))

The designation after the delimiter is only valid for designation handling according to DIN. If you are working according to IEC you can leave this part out.

Example of Function/Plant Designation According to IEC

=MDTF001

- (1) Milko Chemical Plant
- (2) Milk Distribution
- (3) Product Transfer
- (4),(5) Flow measure 001

Example of Function/Plant Designation According to DIN

=MDTF001.L1

- (1) Milko Chemical Plant
- (2) Milk Distribution
- (3) Product Transfer

(4),(5) Flow measure 001

(6),(7) I/O diagram

Table 36. Plant

Code	Description
M	Milko Chemical Plant

Table 37. Main Function

Code	Description
D	Milk Distribution
R	Milk Reception
S	Product Storage

Table 38. Sub Function

Code	Description
L	Liquid Processing
P	Product Distribution Manifold
T	Product Transfer
Q	Quantity Control
S	Milk Supply

Table 39. Technology Function Group

Code	Description
A	Measurement of wet
B	Power supply/distribution
C	Control

Table 39. Technology Function Group (Continued)

Code	Description
D	Drive systems, heaters
E	Power, voltage, current measurement
F	Flow measurement
G	Position measurement
H	Signalization
I	Intercommunication systems
J	Free
K	Fire protection systems
L	Level measurement
M	Group control e.g. emergency stop
N	Control system
O	Free
P	Pressure measurement
Q	Quality measurement
R	Control/position drive/valve
S	Frequency, velocity, rpm measurement
T	Temperature measurement
U	UPS (uninterruptible power supply)
V	Valves
W	Weighing systems
X	Free
Y	Free
Z	Interface plant network

Table 40. Documentation Group

Code	Description
A1	Cover sheet (Cover)
AA	List of contents (Index)
AB	Function description
AC	Mechanical overview
AD	Electrical overview (SLD)
AS	Equipment plan
AX	Interconnection diagrams
B	Terminal diagrams / Cable List
C	Medium voltage supply
D	Low voltage supply, main and control circuits
E	Auxiliary voltage distribution
F	Control voltage distribution
G	Current rectifier
H	DC power circuit
K	Solenoid valves
L	Input / output (control system)
M	Control level HW, Drives
N	Free
P	Function diagrams
Q	Free
R	Monitoring

Table 40. Documentation Group (Continued)

Code	Description
S	Measurement
T	Free

Revision History



The revision index of this System Guide is not related to the 800xA 6.0 System Revision.

The following table lists the revision history of this System Guide.

Revision Index	Description	Date
A	Published for 800xA 6.0.1 release	October 2015
B	Published for 800xA 6.0.3 release	September 2016



A

- AC 800M HI 77
- Access Control 186
- Access Through a Firewall 47
- Acknowledge
 - Alarm 259
- ActiveX Wrapper 297
- Additional aspects 323
- Alarm
 - Acknowledge 259
 - List 259
 - Server 260
- Alarm and Event 252
 - Buffer 265
- Alarm Band 259
- Alarm Blocking 98
- Alarm Classes and Severities 69
- Alarm Expression 260
- Alarm Priorities 95
- Alarms and Events 94
- Allocatable Group 231
- Allocation 231
- AO Alarms 94
- Application Software
 - DownloadApplication Software
 - Test 214
 - Generate 214
- Application/Project Specific Library Solutions 75
- Archive 190
- Archive Guidelines 110
- Archive Rates 109
- Aspect Object property 68
- Aspects 309
- Asset Optimization 213
- Asset Optimization Reporting Requirements 152

- Asset Structure 345
- Audit Trail 33
- Authentication 187
- Authorization 186
- Automation Solution 219

B

- Base Library
 - States 332
- Basic Navigation 42
- Basic Object Types 366, 371, 378, 381
- Basic Structures 204, 315
- Batch Management 213
- Block diagram 272
- Buffer 265
- Built 248
- Bulk Data Applications 284
- Bulk Data Management 278
- Bulk Data Manager 278

C

- Calibration Management 153
- Capacity 34
- Client 200
- CMMS Integration 153
- Code Generation 231
- Communication Board 210
- Component view 227, 241
- Composite Object Types 321
- Condition Monitoring 153
- Configuration Change Management 188
- Configure
 - Alarms and events 262
- Connections via Firewalls 46
- Connectivity Server Data retention on-line 108

Consistency 337
Control Designation 320
Control Hardware Units 238
Control Logic 219
 Add Specific 211
Control Loop 210
Control Module Graphics 235
Control Reference Designations 372
Control Structure 205, 207, 210, 320, 371, 374
Controller 210
Controller as Gateway 139
Controller Code
 Download 214
Controller-level Topology 130
Customization 201

D

Data Exchange 289
Data Flow Order 231
Deploy 335
Design 212
Design Information 206
Designation 316
Device Checks 190
Device Parameter 168
Diagram view 227, 241
Differences 337
Digital signature 188
Direct Integration to 800xA 139
Document Management 293
Document Manager 204
Documentation 380
Documentation Designation 320, 381
Documentation Reference Designations 383
Documentation Structure 204 to 205, 209, 320, 380
 Finalize 213
Double-authentication 33, 187
Download 233
Dynamic Documents 294

E

Electronic Signature 188
Entity 334
Environment 339
Environments
 Aspect versions 341
 Configure-Deploy 340
 Engineering 339
 Evaluate 341
 Load-Evaluate-Go 341
 Previous 341
 Production 339
 Production-Only Aspects 342
Event data 107
Event List 259
Event Storage Server 260
Extend Configuration 213
Extended Configuration Data 289
Extension Library
 States 333
External Alarm 260

F

Field Device Assignment 167
Field-level Topology 130
File Viewer 295
Formal Instance list 324
Formatted Templates 284
FOUNDATION Fieldbus 129
Function Aspect 227
Function Components 227
Function Designer 226
Function Diagram 210
Function Diagrams 228
Function Reference Designations 368
Functional Designation 320, 368, 370
Functional Structure 204, 207, 210, 320, 365, 369

G

Generic OPC 140

-
- Graphic aspect categories
 - Differences 248
 - Graphic Aspects 246
 - Graphic Displays
 - Deploy 214
 - DesignGraphic Displays
 - Build 212
 - Finalize 212
 - Graphic element 249
 - Graphics Builder 244
- H**
- HART Topology 127
- I**
- I/O Board 210
 - I/O Checks 166
 - I/O Signal Allocation 232, 238
 - I/O Signals
 - Allocate 212
 - Assign 211
 - I/O Station 210
 - IEC 61850 131
 - IEC 61850 Connect System Topology 131
 - Import / Export 335
 - Entities 274
 - Import/Export tool 273
 - Indirect Integration to 800xA 139
 - Industry Specific Library Solutions 36
 - Information Management 213
 - Information Manager Server Retention On-Line 109
- L**
- Libraries 35
 - Operations 330
 - Library 201
 - Library Management 330
 - Life Cycle 48
 - List
 - Alarm 259
 - Event 259
 - List Based 289
 - Location Designation 320, 378
 - Location Reference Designations 379
 - Location Structure 205, 209, 211, 320, 375, 380
 - Finalize 213
 - Log Over 188
 - Log over 188
 - Log Templates 104
 - Logging 259
 - Logs 167
 - Long Term Logs 106
 - Loop 197
 - Loop check 169
- M**
- Maintenance Alarms 94
 - Message data access 267
 - Message List 259
 - message log
 - consolidation 269
- N**
- Name 29, 317, 320
 - name 29
 - Name Syntaxes 69
 - Name Uploader Aspect 69
 - Names and Designations 29
 - Naming 321
 - NLS for controller Alarm and Event 68
 - Notification methods 270
- O**
- Object libraries 201
 - Object Type Structure 366, 378
 - Object Types 310
 - Objects and Aspects 307
 - OPC Property Settings 69
 - Operation

SMS and e-mail messaging 269

P

Peer to Peer Communication 69
Performance 34
Permissions 31
Plant Explorer 315, 319
PLC Connect 139
Prefix 320
Procedure function chart 119
Process alarms 94
Process Function 197
Process Functions
 Allocate 211
 Create 210
 ModifyProcess Functions
 Allocate 211
Process Graphics 243
Process Object 210
Process Objects 307
Production Data Logs 107
PROFIBUS DP 124
PROFIBUS PA 127
PROFIBUS Topology 124
Project 200
Project Constants 68
Project Environment 200
Project Library 202
Project Workflow 196

Q

Quick Access Tools 42

R

Re-authentication 33, 187
Recovery Strategy 109
Reference designations 30, 317, 319
Relative designation 319
Replace 336
Report Templates 112

Reservation 338
 Release 337 to 338
 Reserve 338
Reserve 337

S

Scheduling 112
Security 185
Security Policy 45
Security Settings 195
Security Strategy 44
Sequence Bar 259
Sequential Function Chart 235
Server 200
Shared equipment modules 117
Short Term Logs 105
Signal Group 241
Signal Grouping 241
Signal Object 210
Signal Objects 308
Site Engineering
 Workflow 215
SMS and e-mail messaging
 Block diagram 272
 Notification methods 270
 Operation 269
Spread Sheets 277
Standard Libraries 36
Start Situations 25
Structured Plant Modelling 365
Structures 26
 Admin Structure 344
 Aspect System Structure 344
 Control Structure 345
 Documentation Structure 348
 Functional Structure 348
 Graphics Structure 349
 Library Structure 350
 Location Structure 351
 Maintenance Structure 352

- Node Administration Structure 352
- Object Type Structure 353
- Obsolete Structure 354
- Product Structure 355
- Product Type Structure 355
- Reuse Design Structure 355
- Service Structure 355
- System Structure 355
- User Structure 356
- Workplace Structure 357

Structuring of Assets 152

Supertype Reference 368, 379, 382

System 200

System Alarms 94

System Checks 190

System Configuration Version 188

System Objects 308

T

Task Tuning 167

Tasks 70

Test 233

Topology aspect 241

Topology Designer 241

Topology Diagram 241

Topology Status Viewer 241

Trend Logs 105

Trends 252

Typical Solution 201

Typical Solutions 325

U

Unacknowledged Alarm Indicator 259

Update 335

Upload device parameter 168

User Concept 151

User Groups 31

User Roles 356

User strategy 30

Versioning 188

V

W

Web Page 296

Windows Application 297

Workplace Concept 152

Contact us

www.abb.com/800xA
www.abb.com/controlsystems

Copyright © 2016 ABB.
All rights reserved.

3BSE041389-600 B

Power and productivity
for a better world™

