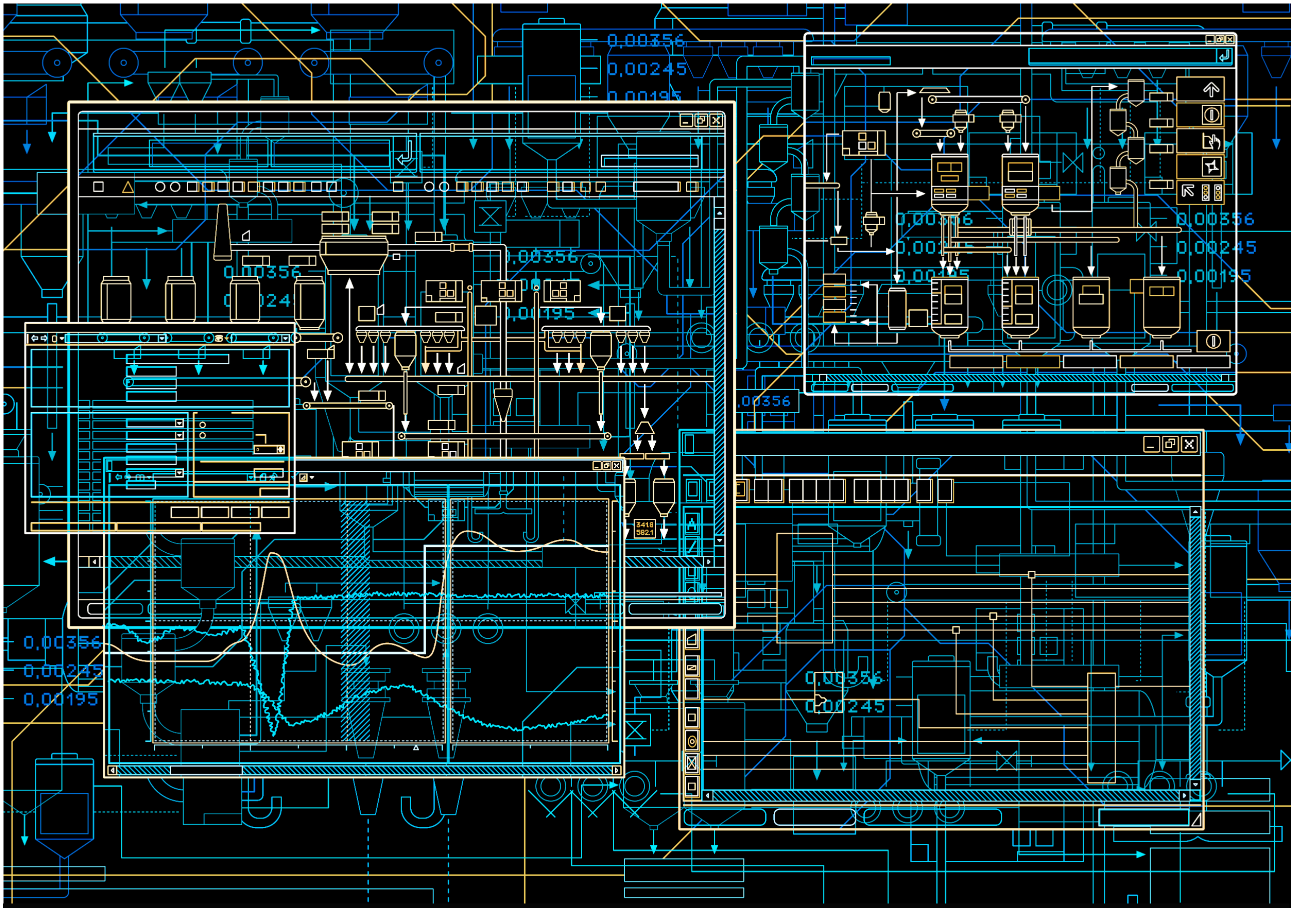


AC 800M

OPC Server

System Version 5.1

Power and productivity
for a better world™

AC 800M

OPC Server

System Version 5.1

NOTICE

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

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

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

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

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

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

TRADEMARKS

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

Release: June 2010
Document number: 3BSE035983-510

TABLE OF CONTENTS

About This Book

Section 1 - Introduction

Section 2 - Start-up and Configuration

Section 3 - OPC Server Operation

Section 4 - Data Access

Section 5 - Alarm and Event

Appendix A - Running the OPC Server and OPC Client on Different Computers

INDEX

About This Book

General

This manual covers configuration issues (completely) for Compact Control Builder (CCB) Users and (partly) for 800xA Users. Therefore, if you run OPC Servers in an 800xA system, only the first part of this manual concerns you, which means exactly up to and including Section 3. Hence, if you wish to run OPC Servers with Compact Control Builder, you are advised to studying the complete manual.

Since the behavior of the OPC Server is different depending on a CCB or 800xA situation, a preset of separate installation instructions can be found in the manual Getting Started[1] (CCB), or System Installation[2] (800xA) respectively.

Section 1, Introduction, introduces *OPC Server for AC 800M* and what should be considered before installing and configuring the product.

Section 2, Start-up and Configuration gives a description on how to set up the system and start the *OPC Server Configuration Panel* for the first time and configure the settings from there.

Section 3, OPC Server Operation deals with subjects applicable for both parts of OPC Server for AC 800M. The different parts of OPC Server is described in Section 4 and Section 5.

Section 4, Data Access describes the OPC Data Access Server.

Section 5, Alarm and Event describes the OPC Alarm and Event Server.

Appendix A, Running the OPC Server and OPC Client on Different Computers

This product meets the requirements specified in EMC Directive 89/336/EEC and in Low Voltage Directive 72/23/EEC.

[1] Getting Started, Introduction and Installation, (3BSE041584*) provides installation instructions for Compact Control Builder Users.

[2] System Installation, (3BSE034678*) provides installation instructions for 800xA Users.

Document Conventions

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

Warning, Caution, Information, and Tip Icons

This publication includes **Warning**, **Caution**, and **Information** where appropriate to point out safety related or other important information. It also includes **Tip** to point out useful hints to the reader. The corresponding symbols should be interpreted as follows:

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

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

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

Information icon alerts the reader to pertinent facts and conditions.

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

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

Terminology

A complete and comprehensive list of Terms is included in the Industrial^{IT} Extended Automation System 800xA, Engineering Concepts instruction (3BDS100972*). The listing included in Engineering Concepts includes terms and definitions that apply to the 800xA system where the usage is different from commonly accepted industry standard definitions and definitions given in standard dictionaries such as ***Webster's Dictionary of Computer Terms*.** Terms that uniquely apply to this instruction may be included here as part of this document.

Term/Acronym	Description
AE	Alarm and Event
COM	Component Object Model, which is the Microsoft binary standard, defines how objects can interact.
DA	Data Access
DCOM	Distributed Component Object Model is an extension of COM. It defines how objects can interact over a network. COM and DCOM are available on most Windows platforms and on several Unix platforms.
MMS	Manufacturing Message Specification
POU	Program Organization Unit
RNRP	Redundant Network Routing Protocol
CCB	Compact Control Builder

Related Documentation

A complete list of all documents applicable to the 800xA Industrial^{IT} Extended Automation System is provided in 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 Introduction

What is OPC Server for AC 800M?

The OPC Server for AC 800M is used for reading run-time data and/or alarms and events from controllers via an OPC interface.

When running with a third part OPC client in an 800xA, it is important to only connect to the 800xA System OPC interface. Thus you must **not** connect to the OPC Server for AC 800M. It is only the 800xA System OPC interface that is adequate enough to handle security issues, band width, performances etc in this type of 800xA configurations with a third part client.

The *OPC Server for AC 800M* consists of the following parts:

OPC Data Access Server

The OPC Data Access Server provides data from controllers programmed with Control Builder through an OPC Data Access 2.0 and 1.0A interface. For more information, see Section 4, Data Access.

OPC Alarm and Event Server

The OPC Alarm and Event Server generates a stream of information, mostly event notifications, and distributes them to the Alarm and Event clients. OPC Alarm and Event 1.02 standard is supported. For more information, see Section 5, Alarm and Event.

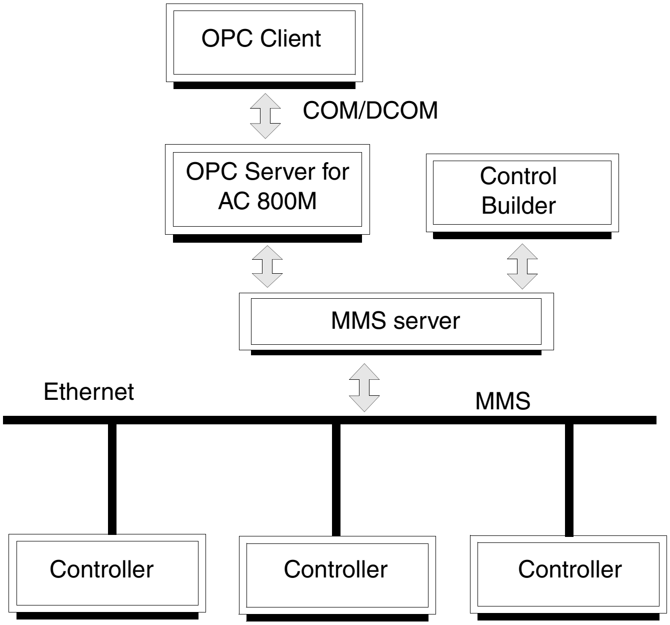


Figure 1. OPC Server configuration in a system.

System Configuration

The following basic rules apply to system configuration and OPC servers:

- One OPC server can receive data or event notifications from multiple control systems,
- A maximum of three OPC servers are allowed to subscribe to Data Access and/or Alarm and Event from one controller,
- A control system can send data or event notifications to multiple subscribing OPC servers,
- Multiple OPC clients can subscribe to data or event notifications from the same OPC server.
- A maximum of five third-party OPC clients are allowed to subscribe to Data Access and/or Alarm and Event from one OPC server,

- A maximum of four Ethernet links (two redundant) are supported via Ethernet cards.
- A third-party OPC client should, if possible, be installed on the same PC as the OPC server. Otherwise, there is a potential risk that communication will not restart after a communication failure (only for CCB Users).

For details regarding the maximum number of supported controllers from which the OPC server subscribes data, refer to *Compact Control Builder AC 800M Product Guide (3BSE041586*)*.

Network Considerations

All PCs must have the same primary network as the OPC Server in order to be able to communicate with this network.

If a Control Builder on PC A has network addresses 172.16.67.41 and 10.46.41.241, and Control Builder on PC B has the addresses 172.16.67.42 and 10.46.41.242, both PCs must have the 172.16 network as their primary network.

The primary network is the network defined as the first network in the network setup.

The OPC server needs access to certain files in order to connect to a controller. These files are stored by the Control Builder that performed the last download. The OPC server can locate these files automatically after a new download.

See File Management on page 16.

Communication Considerations

Communication handling has the lowest priority in a controller. It is therefore important to consider the load on a controller’s CPU if the communication handling is vital to the application. Running close to 100% total load will result in poor capacity and response times for peer-to-peer and *OPC Server for AC 800M* communication. It is hence recommended that the peak total load on the CPU is kept below 100%.

OPC Server for AC 800M Functions

File Management

The behavior of the OPC Server is different depending on a CCB or 800xA situation, because the file management is different. Study the illustrations below in Figure 2 and Figure 3 to learn the different OPC Server behaviors.

- **Control Builder Professional (800xA)**

  Control Builder M Professional can only be run in integrated mode, where all data is stored in the Aspect Directory database. See also Define System on page 29.

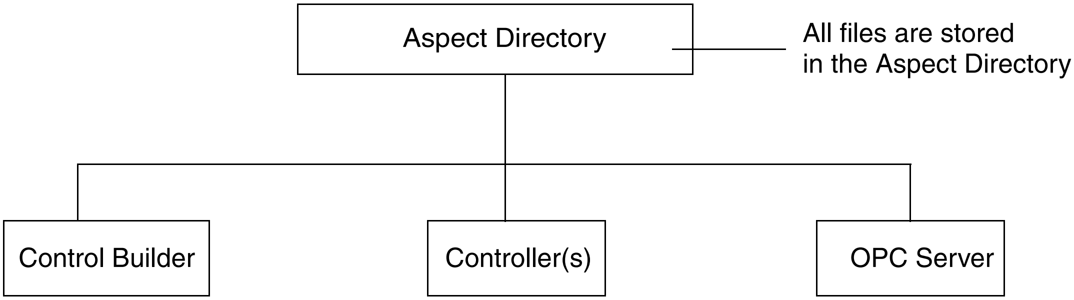


Figure 2. File Management in Control Builder Professional.

- **Compact Control Builder (CCB)**

  An OPC Server has all its data stored in a project folder located on a File Server.

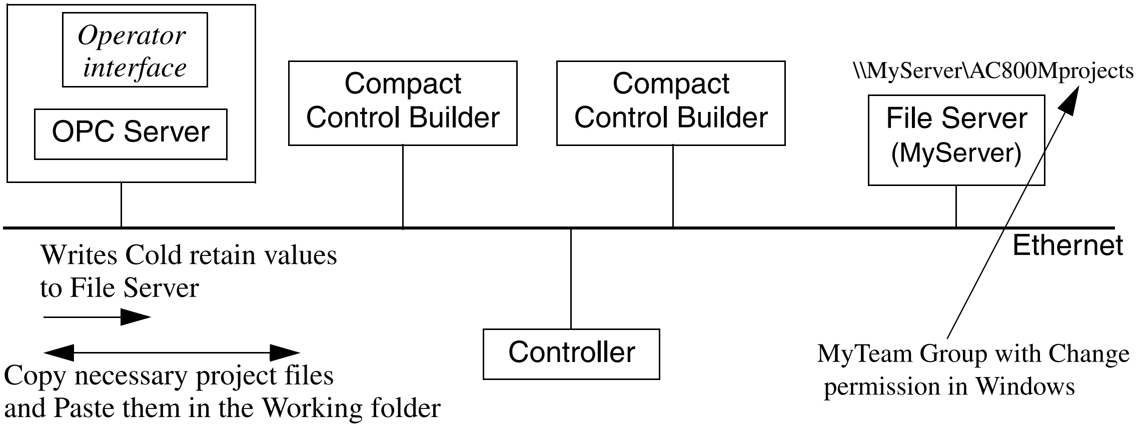

Operator interface
OPC Server
Compact Control Builder
Compact Control Builder
\\MyServer\AC800Mprojects
File Server (MyServer)
Ethernet
Writes Cold retain values to File Server
Copy necessary project files and Paste them in the Working folder
Controller
MyTeam Group with Change permission in Windows

Figure 3. File Management in CCB.

Save Cold Retain Values

Save Cold Retain Values is a function in *OPC Server for AC 800M* that cyclically saves variable values with the attribute cold retain. For more information, see Define System on page 29.

COM and DCOM

COM (Component Object Model) is Microsoft’s binary standard, defining how objects can interact. An object can, for example, be a data structure, a system module or a whole program.

DCOM (Distributed Component Object Model) is an extension of COM. It defines how objects can interact over a network.

If the OPC server is to be run on one PC and the OPC client is to be run on another, DCOM settings must be configured. However, DCOM should only be used if absolutely needed due to the risk for lost connection (see Lost Connection Between OPC Client and OPC Server on page 68).

For more information on DCOM Settings, see Appendix A, Running the OPC Server and OPC Client on Different Computers.

Clock Synchronization

OPC Server for AC 800M contains a function for clock synchronization. If enabled, the OPC server will time synchronize all connected controllers at a certain interval (also, see Clock Synchronization on page 28).

This type of clock synchronization should only be used if there is no clock master synchronization throughout the controller network. The clock master could be an AC 800M controller, but also another type of controller, for example an AC 400 controller.

Prerequisites and Requirements

OPC Server for AC 800M requires Windows 7 or Windows Server 2008 operating system, and ethernet for controller communication.

Intended User

The three beginning sections in this manual covers configuration issues and should be studied by both 800xA and CCB Users. However, the remaining sections with start from Section 4, Data Access, only concern CCB Users and can therefore be ignored by 800xA Users.

Section 2 Start-up and Configuration

This section will give you start-up information and basic information regarding how to configure the OPC Server for AC 800M.

Configuration

After running the *Setup Wizard* it is time to start the *OPC Server Configuration Panel* for specification of controllers to be connected.

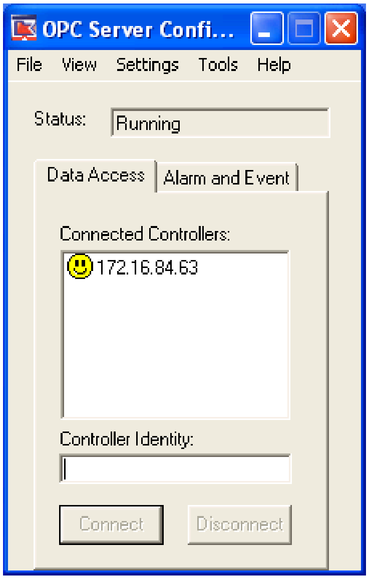
To open the *OPC Server Configuration Panel*:

Start > All Programs > ABB Industrial IT... > .... > OPC Server for AC 800M 5.1 >OPC Server for AC 800M 5.1 .

ℹ *OPC Server for AC 800M* automatically starts when the PC is rebooted. However, this can be changed via **Start > Control Panel > Administrative Tools > Services**. Here, right-click **ABB OPC Server for AC 800M** and select **Properties**. Under the **Startup type** drop-down menu, three options are available – Automatic, Manual, and Disabled. Choose the required option.

OPC Server Configuration Panel

The *OPC Server Configuration Panel* is used to specify the controller(s) to be connected to the OPC Server. When connected, the OPC Server has access to all relevant data in the controller(s). The panel has two controller specification tabs, one for Data Access and one for Alarms and Events.



Status: Running

Connected Controllers: 172.16.84.63

Controller Identity:

Figure 4. The OPC Server Configuration Panel

The OPC Server normally reads the domain files, (created by Control Builder) when the Connect-button is pushed. If the OPC Server for some reason cannot find the files, (after you pushed the Connect-button) you can locate them via a Browse-dialog. However, to select the domain files during online mode (red error icon) double-click the red error icon to open the Browse dialog.

There are three common reasons the OPC Server cannot find the domain files:

1. Wrong Service Account, (no privileges to access aspect directory or Shared network disk) 800xA or CCB respectively.
2. Wrong System connection (800xA only), see Define System on page 29.
3. Wrong path to Shared Network disk (CCB only).

OPC Server Online Help

There is an extended online help function for *OPC Server for AC 800M*, which can be accessed by clicking **Help > Contents...**

Connecting Controllers

Connect Controllers to OPC Data Access Server

To connect controllers to the *OPC Data Access Server*, click the **Data Access** tab and follow the steps below:

1. In the **Controller Identity** entry field, enter the IP address for an available controller.
2. Click **Connect** to connect the controller.
   The controller IP address will now appear in the **Connected Controllers** list.

Each of the controllers connected has an icon indicating the current status of the connection to the controller.

To add more controllers, repeat the procedure above.

If a controller containing a distributed application (an application that exists in more than one controller) is connected, all variables in the application is accessible even if they are owned by another controller. To access the hardware configuration of the other controller, explicitly connect to that controller.

Connect Controllers to OPC Alarm and Event Server

The *OPC Alarm and Event Server* gets data from controllers via subscription. When adding the subscriptions of Alarm and event to a controller, the address of the OPC Server will be stored in the controller. All *Event Notifications* from the controller will be pushed to the OPC Server.

To connect controllers to the *OPC Alarm and Event Server*, click the **Alarm and Event** tab and follow the steps below:

1. In the **Controller Identity** entry field, enter the IP address for an available controller.
2. Click **Connect** to connect the controller.
   The controller IP address will now appear in the **Connected Controllers** list.

Each of the controllers connected has an icon indicating the current status of the connection to the controller.

To add more controllers, repeat the procedure above.

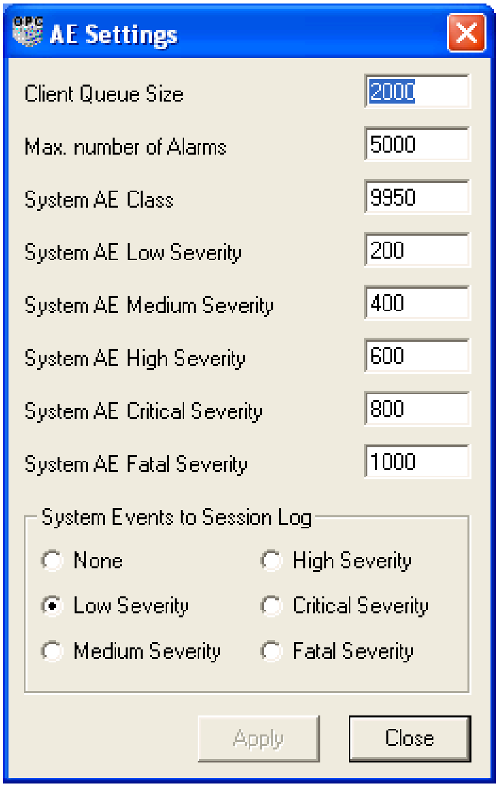
Disconnect Controller

When disconnecting a controller, data in that controller is no longer available to clients of the OPC Server.

To disconnect a controller, select the controller from the **Connected Controllers** list and click **Disconnect**.

Alarm and Event Settings

The following system variables presented in Figure 5, are set to default values and can be changed from the AE Settings dialog. (**Settings > AE Settings...**).



Client Queue Size 2000

Max. number of Alarms 5000

System AE Class 9950

System AE Low Severity 200

System AE Medium Severity 400

System AE High Severity 600

System AE Critical Severity 800

System AE Fatal Severity 1000

System Events to Session Log

○ None ○ High Severity

● Low Severity ○ Critical Severity

○ Medium Severity ○ Fatal Severity

Figure 5. AE Settings dialog box

Client Queue Size

Every OPC Alarm and Event client has a queue in the OPC Server to buffer event notifications. This number specifies the maximum number of items in every queue. Each queue has the same limitations. The default value is 2000, the minimum value is 1500 and maximum value is 5000.

Maximum Number of Alarms

The OPC Server contains a list of all alarms that are not idle (all condition states except the idle state 2) in the connected controllers. This means that you should use the *Max number of Alarm* variable to setup 'The number of alarms that can possible be active at the same time'.

Do not confuse this with your total amount of alarms in the system!

The maximum number of alarms defines the maximum number of alarms in the list, the default value is 5000.

On list overflow, the latest alarm is deleted and a system simple event is generated to announce the overflow.

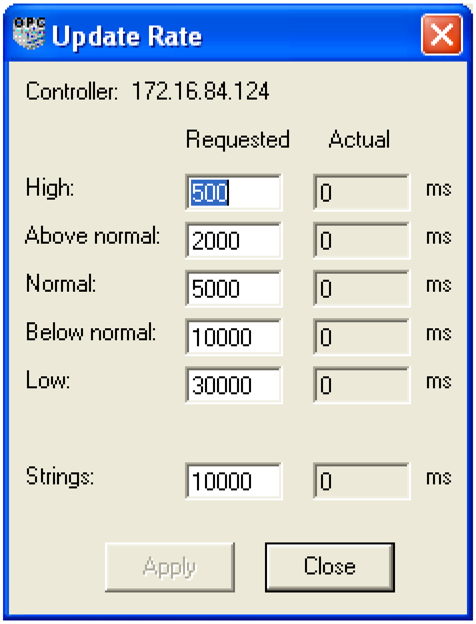
In some cases, in conjunction with large projects, the default value has proven to be too low. It is then recommended to increase the setting from 5000 to, for example, 7500.

Other system variables part from those presented here in the **AE Settings** dialog box, can be configured in the **System Variables** dialog box (**Settings > System Variables**). See Setting System Variables on page 61.

Setting Cache Update Rate

The OPC Server should always have its update rate set twice as fast as the OPC client(s) update rate.

The cache update rate controls how often the OPC Data Access Server updates its internal cache with data from a certain controller. The cache update rate limits the update rate an OPC client can have on an OPC group. To ensure that subscribing OPC clients can access changed values at their requested update rate, the time (value) for cache update rate should be set shorter than the fastest OPC group update rate. To set the update rate, open the Update Rate dialog box by selecting **Settings > Update Rate...** from the *OPC Server Configuration Panel*.



Controller: 172.16.84.124

	Requested	Actual	
High:	500	0	ms
Above normal:	2000	0	ms
Normal:	5000	0	ms
Below normal:	10000	0	ms
Low:	30000	0	ms
Strings:	10000	0	ms

Figure 6. Update Rate Dialog

Settings > Update Rate is only available from the OPC Data Access Server, i.e. when the **Data Access** tab is selected.

To distinguish between fast and slow update rates there are five categories for simple values (*integer*, *real*, *bool*) and a separate category for *strings*. The values in the **Requested** column determines the update rate of the internal cache in the OPC server, i.e. how often items in each category are read from the controller. When *OPC Server for AC 800M* is started, the values are set to default values.

The values in the **Actual** column shows the current update rate. This value will vary around the **Requested** value depending on the load. A zero value in this column means that there is no MMS communication for that category at the moment.

The requested update rates must be given in ascending order, starting at the **High** category. The lowest permitted update rate value for any category is 50 ms.

If any groups has been added, the update rate cannot be changed. The update rate configuration should therefore be made at start-up of the *OPC Server for AC 800M*, before any groups been added.

Example

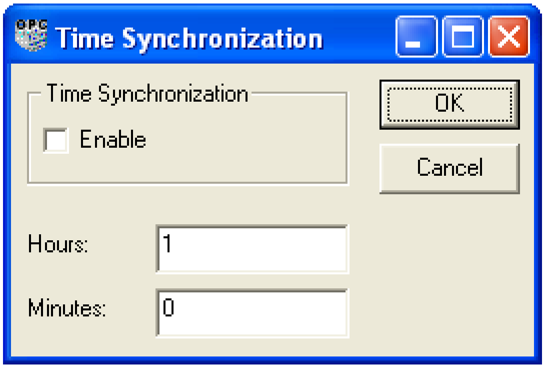
An OPC client creates an OPC group with an update rate of 1000 ms. This means that items added to this OPC group will be reported to the OPC client once every second. In the OPC server, the MMS communication for this group is put in the **High** category, which means that items in this category will be read from the controller every 500 ms.

Clock Synchronization

OPC Server for AC 800M contains a function for clock synchronization. If enabled, the OPC server will synchronize time in all connected controllers, at a certain interval.

This type of clock synchronization should only be used if there is no clock master synchronization throughout the controller network. Such a clock master could be an AC 800M controller, but also another type of controller, for example an AC400 controller.

To clock synchronize from the *OPC Server Configuration Panel*, open the Time Synchronization dialog box by selecting **Settings > Time Synchronization...**



Time Synchronization

□ Enable

Hours: 1

Minutes: 0

Figure 7. Time Synchronization dialog

1. Check the **Enable** check box and set the interval time (one hour is default) in the entry fields for **Hours:** and **Minutes:**.
   Minimum interval time is one minute and maximum is 24 hours
2. Click **OK**.

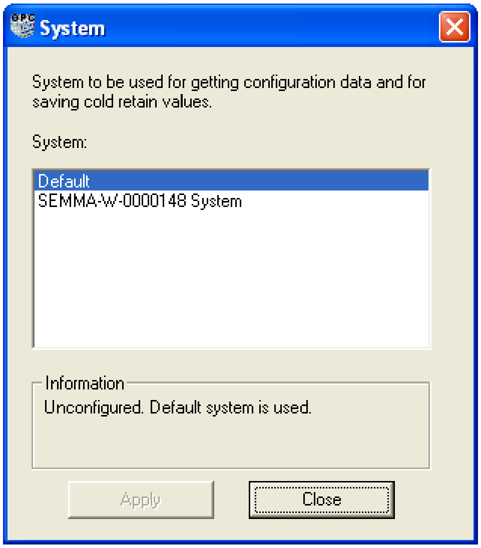
Only controllers with the CPU parameter *CS Protocol Type* set to *MMS* will be synchronized. If the CPU parameter is not set to *MMS*, clock synchronization will be configured from Control Builder.

The accuracy of the time synchronization cannot be guaranteed, due to the fact that it relies on MMS communication. For more information about Clock Synchronization, see the *OPC Server* online help or the Control Builder online help.

Define System

This subsection only describes issues for 800xA and can therefore be ignored by CCB Users.

To select a system and getting configuration data and later on saving cold retain values, open the System dialog box by selecting **Settings > System Connection...** from the *OPC Server Configuration Panel*.


System to be used for getting configuration data and for saving cold retain values.

System:

Default
SEMMA-W-0000148 System

Information
Unconfigured. Default system is used.

Figure 8. System Dialog

Save Cold Retain Values

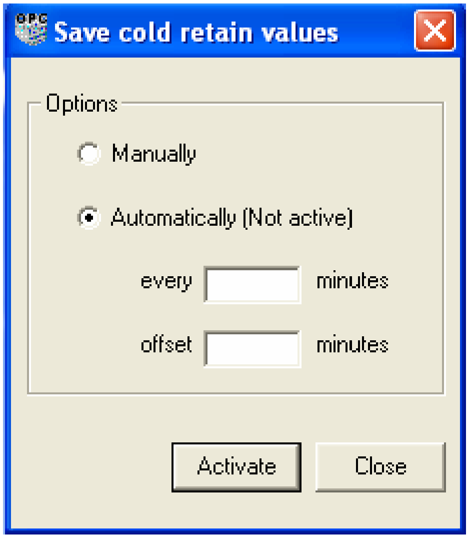
Each time Control Builder enters Offline mode, cold retain values are normally saved. However, there may be long time going between different times of saving these cold retain values, and in case of a power failure more recent cold retain values might be available if using the Save Cold Retain Values function in *OPC Server for AC 800M*.

You can configure the Offset. The Offset is added to the configured cycle time for the first cold retain save after activation.

'Save cold retain values' is a function in the OPC Server for cyclically saving variable values that have been declared with the attribute cold retain.

Saving cold retain values

From the *OPC Server Configuration Panel*, open the Save cold retain values dialog box by selecting **Tools > Save cold retain values...**



Save cold retain values

Options

○ Manually

● Automatically (Not active)

every minutes

offset minutes

Figure 9. Save cold retain values dialog.

To set automatically saved cold retain values, select the option “Automatically”, and set an interval time (recommended every 720 minutes).

To configure offset time for the set interval, set an offset time (recommended value is 0 minutes).

By setting offset time, you can reduce controller load while running redundant OPC Servers. In a redundant configuration, one of the OPC Servers should have offset time set to zero and the other as 360. This setting avoids the situation where two OPC servers fetch cold retain values from the controllers at the same time.

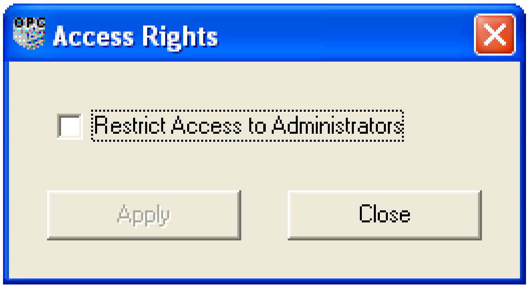
Access Rights

Access right is a setting that aims for protecting the OPC Server configuration panel from unauthorized changes described with start from Connect Controllers to OPC Alarm and Event Server on page 23.

With Access right enabled, all dialog entries will be dimmed, thus locked for unauthorized changes. Please note that only an administrator can disable Access Right and restoring the menu options given from the OPC Server configuration panel.

! Make sure you have administrator privileges before you enable the Access Right.

To enable Access rights, open the Access Right dialog box by selecting **Settings > Access Right...** from the *OPC Server Configuration Panel*.



Access Rights

□ Restrict Access to Administrators

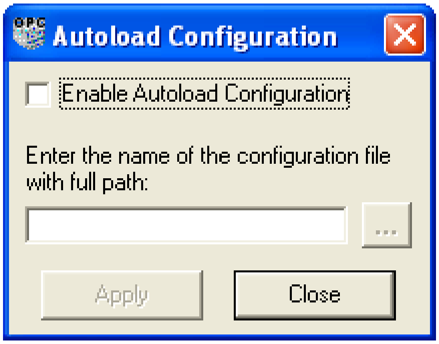
Figure 10. Access Rights dialog

Save Configuration

By clicking **File > Save Configuration** the current configuration settings of the OPC server is saved to a file containing one section for each part of the OPC server. The default extension of the file is .cfg and it is by default saved in the **Files** folder of the OPC server’s working directory.

Automatic Load of OPC Server Configuration

Autoload an OPC Server configuration at start-up can easily be enabled from *OPC Server Configuration Panel*.



□ Enable Autoload Configuration

Enter the name of the configuration file with full path:

Figure 11. Autoload Configuration dialog

The **Enable Autoload Configuration** check box is disabled by default.

1. Open the **Settings > Autoload Configuration Settings...** menu.
2. Check the **Enable Autoload Configuration** check box.
3. Enter the name of the configuration file with full path either by typing it or browsing to it by clicking the **...** button.
4. Click **Apply** to put in effect.

Section 3 OPC Server Operation

General

This section cover information about *OPC Server for AC 800M* that are not specific for the Data Access part or the Alarm and Event part of the OPC Server.

Here you find what is written into the session log, as well as what happens if power failure occur. Some information regarding performance, indication of status and maintenance will also be given.

Session Log

Messages concerning the system are logged in the session log. Types of messages include direct printouts from code or system events. The session log can be opened from *OPC Server Configuration Panel*.

Data Access Printouts to Session Log

The following printouts originate from the OPC Data Access Server.

Table 1. Printouts from Data Access into session log

Printout	Comment
OPC server started	OPC Server has been started
OPC server stopped	OPC Server has been stopped
COM: Waiting for active components	Attempt to shut down OPC Server with OPC clients still connected; followed by a three second delay
OPC: The server had active components when it was shutdown!	OPC Server was shut down with OPC clients still connected (see previous comment)
OPC Data Access: Connected to controller: <IP-address>	Controller is connected to OPC Data Access Server
OPC Data Access: Connected to controller: <IP-address>, warning: no contact	Controller with no contact to OPC Data Access Server has been connected
OPC Data Access: Disconnected from controller: <IP-address>	Controller has been disconnected from OPC Data Access Server
OPC Data Access: Contact lost with controller <IP-address>	Connection between controller and OPC data Access Server has been lost
OPC Data Access: Contact again with controller <IP-address>	Connection between controller and OPC Data Access Server has been re-established
Autoloading configuration...	Autoload configuration in progress
Autoload configuration finished	Autoload configuration is complete
Autoload configuration failed	Autoload configuration has failed
No FTP server available	FTP server is not available

Table 1. Printouts from Data Access into session log (Continued)

Printout	Comment
Saving cold retain files for application <ApplicationName> failed	Saving of cold retain values for the application has failed
Sending cold retain files for application <ApplicationName> failed	Sending of cold retain values for the application has failed

Alarm and Event Code Printouts to Session Log

The following printouts originate from the Alarm and Event part of the code in *OPC Server for AC 800M*.

Table 2. Printouts from code into session log

Type(1)	Printout	Comment
I	OPCAE-~SaveSub. M_OnEventQ is not empty. Size=xxxx	The saved subscription was not empty when deleted. Can occur at timeout
I	OPCAE – Start of subscription. OPC client ID=yyyy 1. Restore events from SaveSub 2. SaveSub not found. No events restored 3. SaveSub not enabled	When an OPC client starts a subscription. One of the three texts is also printed as well, depending on if saved subscription is enabled
I	OPCAE – End of subscription. OPC client ID=yyyy 1. Events are stored in SaveSub 2. SaveSub not created. No events are stored 3. SaveSub not enabled	When an OPC client ends a subscription. One of the 3 texts is also printed as well, depending on if saved subscription is enabled
E	OPCAE – After save in SaveSub. M_OnEventQ is not empty	System error

Table 2. Printouts from code into session log (Continued)

Type[1]	Printout	Comment
E	OPCAE – COPCEventSubscriptionMgt. SystemEventSeverity not found	System error
E	OPCAE – COPCEventSubscriptionMgt. SystemEventClass not found	System error
E	OPCAE – COPCEventSubscriptionMgt: ProcessNewEvent. Refresh failed	System error
E	OPCAE – RemovedSubscribingSystem.m_NrOfSubscribedSystems=xx	System error
E	OPCAE – EventType not supported	System error
E	OPCAE – SaveSub. Error when deleting SaveSub	System error

[1] Type of message tells if it is information (I) or error (E).

System Events to Session Log

System simple events, as well as system condition related events (i.e. system alarm that is going on and off) are also written to the session log. It is possible to filter which system events that should go to the controller log. This is controlled from the system variable *SystemEventsToSessionLog* (see Setting System Variables on page 61 and *OPC Server for AC 800M* Online Help).

The following system simple events and system condition related events may be issued by OPC Alarm and Event Server and written to the session log:

Table 3. System events to session log

Type	Printout	Comment
Event	(6000) Alarm not unique	Generated when OPCAE server discover there are two alarms with the same SourceName/ConditionName combination defined in two different controllers
Event	(6001) Alarm Handler overflow. MaxNoOfAlarms exceeded.	Generated when an alarm item in the Alarm Handler must be detected because of overflow. The size of the Alam Handler is limited by system variable *MaxNoOfAlarms*
Event	(6003) Overflow in queue to OPC client	Generated when there is an overflow in the queue to OPC client
Event	(1011) Alarm definition failed for *<SourceName> <ConditionName>*	The alarm definition in the 1131 application code failed. This message is concatenated with SourceName and ConditionName of alarm

Table 3. System events to session log (Continued)

Type	Printout	Comment
Alarm	(5500) Connection error to DA subscribed controller	Each subscribed controller from OPC Data Access Server have one system alarm. These system alarms are: • activated when an error occurs on this connection • inactivated when all errors are corrected or disappeared
Alarm	(6500) Connection error to AE subscribed controller	Each subscribed controller from OPC Data Access Server have one system alarm (see description above)

Power Failure

The OPC server runs as a service, and can be configured so that configurations are automatically loaded at startup. Groups and items always have to be re-configured after startup.

When an OPC client is stopped abruptly, it is detected by the OPC server, which stop making callbacks. After COM timeout, all memory allocated for the specific client, is released.

See also OPC Data Access Server Behavior at Power Fail on page 42, and OPC Alarm and Event Server Behavior at Power Fail on page 60 for more information.

Error Handling

In addition to standard OPC error handling, serious errors are logged as a log file. Errors are also presented in the *OPC Server Configuration Panel*.

Performance

For specific performance information regarding OPC Data Access Server and OPC Alarm and Event Server, see subsection **Performance** of respective section.

Section 4 Data Access

General

The OPC Data Access Server provides data from controllers programmed with *Control Builder M* through an OPC Data Access 2.0 and 1.0A interface. The program ID is: ABB.AC800MC_OpcDaServer.3

In order to receive data from the controller, it has to be connected to the *OPC Server Configuration Panel*. During compilation, Control Builder saves information about applications, hardware configuration and access variables to files. They are the .cdoXml,.rrsXml domain files for each application and the .ctdXml domain files for each controller. To connect to a specific controller, the OPC Data Access Server is dependent on these domain files. The OPC Server fetch the domain files automatically, when needed. See also File Management on page 16 or OPC Server Configuration Panel on page 21.

Controller and application names must be unique throughout the connected controllers. The OPC Server is not able to provide data for duplicate names. See also Address Space on page 47.

The OPC Sever keeps its name references even when the communication to the controller is lost, consider the following scenario:

Two SoftControllers and which both are configured in the OPC Server. An application is downloaded to the first SoftController, hence the OPC Server provides good data for it. If the Soft Controller will be terminated, the OPC Server will consider this as loss of communication, hence provides bad data.

When the application is downloaded to the second SoftController, the OPC Server will **still** provide bad data for the application!

The explanation is that the OPC Server has references to two applications with the same name, therefore good data will not be provided until the first SoftController is restarted (the OPC Server sees this as regained communication). Once the first

SoftController has been restarted the OPC Server will remove the reference of the former application. Good data will be provided for the application in the second SoftController.

Fast Initial Update

When an OPC client adds an item to a group with reasonably long update rate for example, 30 seconds the OPC server updates the new item with an initial value to the client.

When a new item is added the OPC Server starts to cyclically check if a value has arrived from the controller at each 100 ms, independently of the group update rate determined by the client. As soon as the value has arrived, it sets the quality to good and updates the client. This means that the client does not have to wait until the next regular group update, (another 30 seconds) before it can receive the new added item with good quality.

OPC Data Access Server Behavior at Power Fail

OPC Server Power Failure

The OPC server runs as a service, and can be configured so that configurations are automatically loaded at startup. Groups and items always have to be re-configured after startup. See *Online Help for OPC server, Autoload Configuration Settings*.

OPC Client Power Failure

When an OPC client is stopped abruptly, it is detected by the OPC server, which stop making callbacks. After DCOM time-out, all memory allocated for the specific client, is released.

DCOM Time-out

Callbacks to the client are made on separate threads. Therefore the OPC server never fails when DCOM problems occur. After DCOM time-out, all objects are released according to standard COM.

Error Handling

In addition to standard OPC error handling, serious errors are logged as a log file. Errors are also presented in the *OPC Server Configuration Panel* by double-clicking the icon ☺ . This brings up error information.

What Data Can Be Accessed?

Through the OPC Data Access Server different kinds of runtime-data can be accessed from:

- Applications
- Controllers
- Hardware units

Applications

The Applications branch contains the applications running in the controllers connected to the OPC Data Access Server. You can access variables and function block and control module parameters.

Controllers

The Controllers branch contains the connected controllers to the OPC Data Access Server. Each controller can access hardware variables and access variables.

For each controller, there are four items that reflects the connection between *OPC Server for AC 800M* and the controller:

- HWState
  0 = StateOK
  1 = StateError
- HWStateChangeTime
  Time for last change of HWState
- ErrorsAndWarnings
  0 = StatusOK
  1 = CommunicationError
  2 = FilesMissing
- LatchedErrorsAndWarnings
  0 = StatusOK
  1 = CommunicationError
  2 = FilesMissing

Hardware Variables

You can access attributes in I/O channels and status channels of hardware units. All attributes are read-only. Each channel has a default attribute that is sent when the channel is accessed without specifying an attribute. When you browse the address space, you cannot view the attributes of channels.

Mapping of OPC

Mapping IO Channel Status to OPC Quality

When an OPC client subscribes to an item, that is an I/O channel, a Value, or an I/O Value component of a structured variable of I/O data type (BoolIO, RealIO, DIntIO, DWordIO), the quality of the item will be determined by the corresponding Status component. When the client subscribes to such an item, the OPC server also subscribes internally to the Status component (inside the I/O data type) and uses its value to calculate the quality when the client is updated with a new value.

The quality for *all other items* is decided by the MMS communication against the controller. As long as the OPC server receives value updates from the controller the quality is set to good. Since this functionality, with subscribing to the Status component, means that extra variables will be communicated between the OPC server and the controller, it will increase the load and might affect performance. However, if you do not need this functionality, set the system variable OPCGetQualityFromStatus to false (true by default). In that case item quality is determined by the status of the MMS communication.

Interfaces Supported

All non optional interfaces are supported in accordance to the OPC Data Access specification 2.0. Also, IOPCBrowseServerAddressSpace, and IPersistFile are supported. The following services are **not** supported.

- Public groups
- Deadband
- EU information on OPCITEMATTRIBUTES

States of Status

The OPC Data Access Server can be in one of the following states:

- OPCDA_STATUS_RUNNING
  At least one controller connected and communication is working
- OPCDA_STATUS_NOCONFIG
  No controllers are connected **or** no communication is working to the controller(s) connected.

If no connection is detected when *OPC Server for AC 800M* is started, the state is OPCDA_STATUS_NOCONFIG. When connecting to a controller, the *OPC Server for AC 800M* goes into the OPCDA_STATUS_RUNNING state.If connection is lost to all controllers, the OPC Server re-enters the OPCDA_STATUS_NOCONFIG state.

OPC Data Access Overview

In the OPC Server, a server object is created for each OPC client connected. When an OPC group is created in the OPC client, a group object is created in the server object.

To each OPC group, items (such as variables) are added and update rate is set for the OPC group.

Data access is created by defining OPC groups to which you add items. An item is a data source reference identified by an ItemID. An item has a value, quality and timestamp associated. Item values can be read and written synchronously and asynchronously, or subscribed. You can choose to read a value from the internal cache of the server or directly from the device. The most efficient service when retrieving data is subscription.

For information regarding asynchronous operations, see Limitations Regarding Asynchronous Operations on page 51.

Address Space

The address space for the OPC Server is built just like the tree structure in the Project Explorer of *Control Builder M*. The top level contains two branches, Applications and Controllers. Controller and application names must be unique throughout the connected controllers. Item names are identified by the full path with dot notation, for example Applications.MyApplication.MyProgram.MyDint.

OPC Variables

Variables can be accessed anywhere in the POU structure. Control module and function block parameters can also be accessed. Variables with the attributes *Constant* can only be read and variables with the attribute *Hidden* cannot be seen at all. An item is always of simple datatype.

You can reach access variables, which have been declared accessed from remote systems, in the connected controllers. Access variables can be of simple data types or structured data types.

Table 4. Data Types that are mapped

IEC 61131-3 data type	Variant type
int	VT_I2
dint	VT_I4
uint	VT_UI2
real	VT_R4
word	VT_UI2
dword	VT_UI4
bool	VT_BOOL
string	VT_BSTR
date_and_time	VT_DATE
time	VT_R8 (mapped to ms)

Table 5. Variable Item Properties

ID	Data type of returned VARIANT	Description	Type of property
1	VT_I2	Item canonical data type (VARIANT type of the item value)	Standard OPC property
2	<varies>	Item value (current value)	Standard OPC property
3	VT_I2	Item quality (current quality)	Standard OPC property
4	VT_DATE	Item timestamp (current timestamp)	Standard OPC property
5	VT_I4	Item access rights (readable/writable)	Standard OPC property
6	VT_R4	Server scan rate (time between cache updates)	Standard OPC property
5000	VT_BSTR	Item type name (IEC 61131-3 data type)	Server specific property
5002	VT_BSTR	Item Attribute (retain, cold retain)	Server specific property
5005	VT_BSTR	Item category (parameter or variable)	Server specific property
5006	VT_BSTR	Parameter direction (in, out or in/out)	Server specific property
5007	VT_BSTR	Path (Path to corresponding POU variable)	Server specific property

Item Quality

An item can have one of the following states of quality:

- Good.
- Uncertain.
- Bad.

For example, a non-working item have the status `OPC_QUALITY_BAD`.

For quality **Good**, the following substatus is issued:

– `OPC_QUALITY_LOCAL_OVERRIDE`

This status is issued for I/O channels or items of I/O data type connected to an I/O channel. The value has been Forced.

For quality **Uncertain**, the following substatus is issued:

– `OPC_QUALITY_LAST_USABLE_VALUE`

This status is issued if the OPC Server does not read from the controller in the same pace as it is supposed to update the OPC client.

– `OPC_QUALITY_ENGINEERING_UNITS_EXCEEDED`

Used for I/O channels or items of I/O data type connected to an I/O channel. The value is outside the limits defined for this parameter.

For quality **Bad**, either of the following substatuses are issued:

– `OPC_QUALITY_CONFIG_ERROR`

This status is issued if some problem with the configuration for example, the item in question has been deleted in the controller.

– `OPC_QUALITY_LAST_KNOWN_VALUE`

This status is issued if the OPC Server has no contact with the controller when it is time to update the OPC client with a new value, consequently the OPC Server is giving the OPC client the last known value.

– `OPC_QUALITY_COMM_FAILURE`

This status is issued if communications have failed. There is no last known value available.

- `OPC_QUALITY_OUT_OF_SERVICE`

  Is issued if the active state of the item or the group containing the item is InActive.

- `OPC_QUALITY_UNIT_ERROR`

  Used for I/O channels or items of I/O data type connected to I/O channels. A device failure has been detected. I/O value will be frozen to the last known value.

- `OPC_QUALITY_CHANNEL_ERROR`

  Used for I/O channels or items of I/O data type connected to an I/O channels. A channel error has been detected. The I/O value will be frozen to the last known value.

Writing

Writing can either be done synchronously or asynchronously. If done synchronously, the OPC Server is occupied while writing to an controller is performed and updating of other clients are not possible. If writing is done asynchronously, the OPC Server is able to update other clients while writing to the controller.

The OPC server allows a maximum of 50 simultaneous write calls to the same OPC group. If there is a 51st write call before the first write call is finished, the OPC server will return an error code to the client on the 51st call.

If an OPC client gets status OK, this means the value was successfully written to the controller.

Performance

OPC Server for AC 800M performance varies, considering the number of controllers being used, what type of controller being used, and the number of clients.

For more details about performance of OPC Server for AC 800M, refer to *Compact Control Builder AC 800M Product Guide (3BSE041586*)*.

Time Stamping

Time Stamping of Data Access is done in OPC Server.

Limitations Regarding Asynchronous Operations

The OPC Server has a limitation regarding the number of asynchronous operations it can handle per OPC Group. The following operations are limited to a maximum of 50 uncompleted operations per type:

- IAsyncIO:Read,
- IAsyncIO:Write,
- IAsyncIORefresh,
- IAsyncIO2:Read,
- IAsyncIO2:Write,
- IAsyncIO2:Refresh.

Note that there can be 50 outstanding calls for each of these at the same time in one OPC Group.

If the client continues to call these methods before the OPC Server answers, the OPC Server returns the error CONNECT_E_ADVISELIMIT. This error indicates that the limit is reached. The client should then wait for one of the uncompleted operations of the same type to finish, before trying again.

There is no limit to the number of OPC Items that can be handled in each operation. This means that a client should try to use fewer operations with more OPC items per operation, in order to avoid problems. For instance, if 20 Items are to be written, it is better to send them all in one IAsyncIO2:Write operation, and not use 20 operations with one item in each.

Section 5 Alarm and Event

General

From the OPC Alarm and Event Server, a stream of information is generated which is distributed to the Alarm and Event clients. This information consists mostly of event notifications and such a client can be an event viewer, Alarm/Event Manager, or a logging system. Clients, on the other hand, can send acknowledgements of alarms and requests to the server.

The combination of SourceName/ConditionName of all alarms defined in the connected controllers must be globally unique. A system event is generated if two alarms with the same combination are detected, see also System Events to Session Log on page 37.

OPC Server for AC 800M provides an additional service to the OPC standard via the variable *StandardEnableSource*. However, if you wish to disable this service see the **AE Settings** dialog box under Setting System Variables on page 61.

OPC Alarm and Event Server

The MMS OPC Alarm and Event Server maps the MMS based Alarm and Event communication to the COM/DCOM based OPC Alarm and Event communication. The Alarm and Event Server handles events from many controllers and forwards them to several OPC Alarm and Event clients.

It is possible to have several servers in parallel on different systems. It is also possible that many clients subscribe to the same OPC Server, and that the OPC Server itself subscribes to many controllers.

Interfaces Supported

The Alarm and Event part of *OPC Server for AC 800M* supports the OPC Alarm and Events 1.02 standard. The program ID is: ABB.AC800MC_OpcAeServer.3

The interface contains methods (functions) to acknowledge, enable, and disable alarm conditions, as well as methods (functions) for subscription of events.

OPC clients can also query the Alarm and Event server for information such as condition names used, available vendor specific attributes, etc.

The list of objects with respective interfaces is as follows:

- OPCEventServer object
- OPCEventAreaBrowser object
- OPCEventSubscription object.

OPCEventServer Object

Table 6. Methods of IOPCCommon

Method	**Status**
GetErrorString (dwError, ppString)	Implemented
GetLocaleID (pdwLcid)	Implemented
QueryAvailableLocaleIDs (pdwCount, pdwLcid)	Implemented
SetClientName (szName)	Implemented
SetLocaleID (dwLcid)	Implemented

Table 7. Methods of IOPCEventServer

Method	Status
AckCondition (dwCount, szAcknowledgerID, szComment, pszSource, pszConditionName, pftActiveTime, pdwCookie, ppErrors)	Implemented
CreateAreaBrowser (riid, ppUnk)	Not implemented
CreateEventSubscription (bActive, dwBufferTime, dwMaxSize, hClientSubscription, riid, ppUnk, pdwRevisedBufferTime, pdwRevisedMaxSize)	Implemented
DisableConditionByArea (dwNumAreas, pszAreas)	Not implemented
DisableConditionBySource (dwNumSources, pszSources)	Implemented
EnableConditionByArea (dwNumAreas, pszAreas)	Not implemented
EnableConditionBySource (dwNumSources, pszSources)	Implemented
GetConditionState (szSource, szConditionName, ppConditionState)	Not implemented
GetStatus (ppEventServerStatus)	Implemented
QueryAvailableFilters (pdwFilterMask)	Implemented
QueryConditionNames (dwEventCategory, pdwCount, ppszConditionNames)	Implemented
QueryEventAttributes (dwEventCategory, pdwCount, ppdwAttrIDs, ppszAttrDescs, ppvtAttrTypes)	Implemented
QueryEventCategories (dwEventType, pdwCount, ppdwEventCategories, ppEventCategoryDescs)	Implemented
QuerySourceConditions (szSource, pdwCount, ppszConditionNames)	Implemented

Table 7. Methods of IOPCEventServer (Continued)

Method	Status
QuerySubConditionNames (szConditionName, pdwCount, ppszSubConditionNames)	Implemented
TranslateToItemIds (szSource, dwEventCategory, szConditionName, szSubConditionName, dwCount, pdwAssocAttrIDs, ppszAttrItemIDs, ppszNodeNames, pCLSIDs)	Not implemented

Table 8. Methods of IOPCConnectionPointContainer

Method	Status
EnumConnectionPoints (ppEnum)	Implemented
FindConnectionPoint (riid, ppCP)	Implemented

OPCEventAreaBrowser Object

Table 9. Methods of IOPCEventAreaBrowser

Method	Status
BrowseOPCAreas (dwBrowseFilterType, szFilterCriteria, ppIEnumString)	Not implemented
ChangeBrowsePosition (dwBrowseDirection, szString)	Not implemented
GetQualifiedAreaName (szAreaName, pszQualifiedAreaName)	Not implemented
GetQualifiedSourceName (szSourceName, pszQualifiedSourceName)	Not implemented

OPCEventSubscription Object

Table 10. Methods of IOPCEventSubscriptionMgt

Method	Status
CancelRefresh (dwConnection)	Implemented
GetFilter (pdwEventType, pdwNumCategories, ppdwEventCategories, pdwLowSeverity, pdwHighSeverity, pdwNumAreas, ppszAreaList, pdwNumSources, ppszSourceList)	Implemented
GetReturnedAttributes (dwEventCategory, pdwCount, pdwAttributeIDs)	Implemented
GetState (ppActive, pdwBufferTime, pdwMaxSize, phClientSubscription)	Implemented
Refresh (dwConnection)	Implemented
SelectReturnedAttributes (dwEventCategory, dwCount, dwAttributeIDs)	Implemented
SetFilter (dwEventType, dwNumCategories, pdwEventCategories, dwLowSeverity, dwHighSeverity, dwNumAreas, pszAreaList, dwNumSources, pszSourceList)	Implemented
SetState (bActive, dwBufferTime, dwMaxSize, hClientSubscription, pdwRevisedBufferTime, pdwRevisedMaxSize)	Implemented

Table 11. Methods of IOPCConnectionPointContainer

Method	Status
EnumConnectionPoints (ppEnum)	Implemented
FindConnectionPoint (riid, ppCP)	Implemented

Status

The OPC Alarm and Event Server has the following states:

- OPCAE_STATUS_RUNNING
- OPCAE_STATUS_FAILED (not used for the moment)
- OPCAE_STATUS_NOCONFIG
- OPCAE_STATUS_SUSPENDED

The following applies to OPC Server for AC 800M:

- When the OPC Server is started, the status is set to OPCAE_STATUS_NOCONFIG.
- When connected to at least one controller, the OPC Server status is always OPCAE_STATUS_RUNNING.
- If all controllers are removed from the panel, the OPC Server status returns to OPCAE_STATUS_NOCONFIG.
- The system variable *SetFailedAtNoConn* is used to control the behavior when all connections to controllers are lost (for information on how to set this variable, see Setting System Variables on page 61):
  - The default value of *SetFailedAtNoConn* is True.
  - With *SetFailedAtNoConn* set to True:
    If there are controllers in the panel, but all connections are offline, the OPC Server status is set to OPCAE_STATUS_SUSPENDED. As soon one of the connections is restored, the OPC Server status returns to OPCAE_STATUS_RUNNING.
  - With *SetFailedAtNoConn* set to False:
    The OPC Server status is OPCAE_STATUS_RUNNING as long as there are controllers in the panel, no matter if they have a working connection or not.

Buffer Queues

There is a queue in the Alarm and Event Server for every connected Alarm and Event client (see Figure 12). All data that passes the OPC Server, for example event notifications and acknowledges will be queued. The maximum size of the queues can be configured through the configuration interface. The data in these queues is only held in the primary memory and is lost when the gateway shuts down or when a power failure occurs. It is possible to have a second OPC Server in parallel to increase the data security.

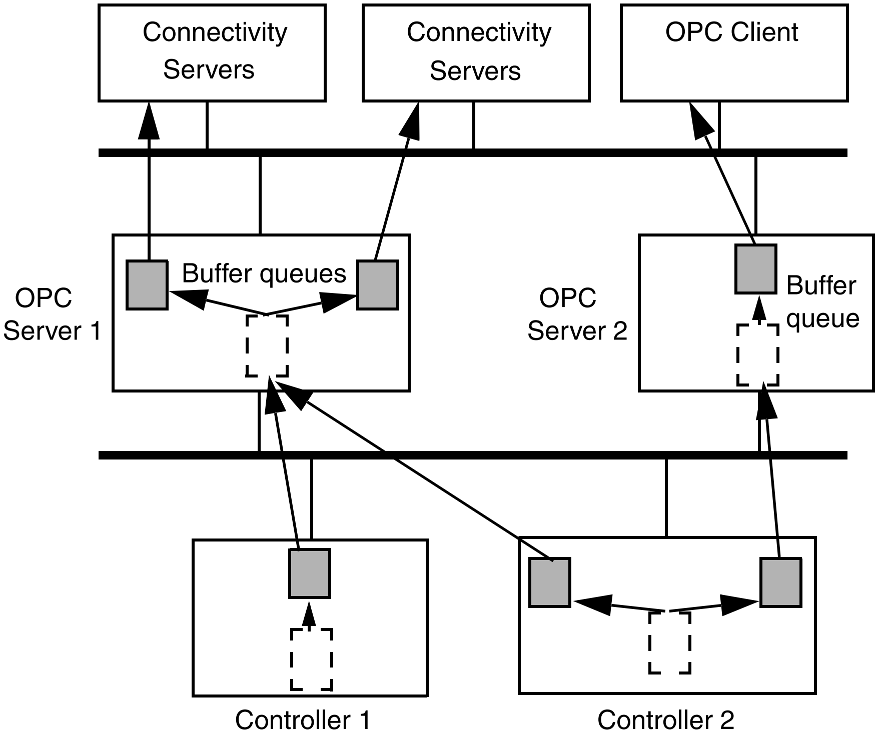


Figure 12. Alarm buffer queues

Note

The Alarm and Events Server has the following limitations.

- Tracking-related events (events generated from operator interactions) are not supported.
- The optional Quality information is not supported.
- The optional SubConditions are not supported. That means that our conditions always only have one SubCondition (Single state conditions).
- The optional Area concept from OPC alarm and event is not supported.

OPC Alarm and Event Server Behavior at Power Fail

If the power fails on the OPC Alarm and Event Server, all events received but not yet sent to respective OPC client, are lost.

When the OPC Server has recovered from power fail, all alarms that are not in idle state is updated in the alarm handler. This means that recreation of alarm lists after power fail is done. As mentioned in previous paragraph, it is **not** possible to recreate condition related or simple events sent from controllers to the OPC Server before the power fail occurred.

OPC Alarm and Event Clients

The OPC Alarm and Event clients are able to perform different tasks, such as filtering out event notifications based on event category.

Setting System Variables

This dialog (see Figure 13) helps you to set system variables other than those given in Alarm and Event Setting dialog. See Alarm and Event Settings on page 24.

- StandardEnableSource
- SendDisabledAE

System variables are set in the System Variables dialog box which can be opened by selecting **Settings > System Variables...** from the *OPC Server Configuration Panel*.

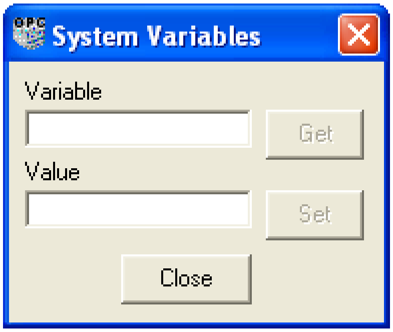


Figure 13. System Variables Dialog

Setting Variables in System Variables Dialog Box

StandardEnableSource

If this choice is selected, the OPC Alarm and Event interface supports standard Enable/Disable by source. This means that all alarms in one object by source are enabled or disabled at the same time.

If this choice is not selected, it is possible to enable or disable single alarms in one object by source.

SendDisabledAE

If this choice is selected, disabled alarms (conditions) are also sent to the OPC Alarm and Event client on refresh.

If this choice is not selected, no disabled alarms (conditions) are sent to the OPC Alarm and Event client on refresh.

If you wish to disconnect the additional service from the OPC Server, thus not sending disabled alarms while refresh then set the variables *StandardEnableSource* = ***true*** and *SendDisabledAE* = ***false***.

For more information about system variables, see the *OPC Server for AC 800M* Online Help.

All the other system variables available, are presented in the **AE Settings** dialog box, (**Settings > AE Settings...**). See Alarm and Event Settings on page 24.

Filtering of Event Notifications

It is possible for OPC clients to filter out event notification based on event category. The OPC standard makes it possible to send vendor-specific information in event notifications. An event notification belongs to a certain event category.

There are six event categories defined within the OPC Alarm and Event Server.

Table 12. Event categories defined in the OPC server

Event category	Vendor specific attributes
Process Simple Event	Class, SourceGUID
Process Condition Event	Class, SourceGUID, AutoDisabled
System Simple Event	Class, SourceGUID
System Condition Event	Class, SourceGUID, AutoDisabled
OPC_SERVER_ERROR	NeedsRefresh
Process Data To Simple Event	XMLData, EventCode, RecipePath, Class, SourceGUID

The Process Simple Event and the Process Condition Event categories are used for simple events respective condition-related events generated from applications.

The categories System Simple Event and System Condition Event are used for system simple events and system alarms.

The OPC_SERVER_ERROR category is a simple event used to tell the OPC client about refresh.

The last category, Process Data To Simple Event, is a simple event generated from an application containing extra user defined data.

The vendor specific attributes are:

- Class – Used to group alarms of the process
- SourceGUID – An unique identifier given to each source to make it easier for the OPC client to identify the source of an alarm
- AutoDisabled – An attribute that indicates if an alarm is in the AutoDisabled state
- NeedsRefresh – Information to the OPC client from the OPC server telling that it is time for refresh
- XMLData – String containing user defined data
- EventCode – Determines batch event type
- RecipePath – Path to recipe

Refreshing OPC Clients

The OPC Server needs to refresh the OPC clients in the following situations:

- After adding a new subscription or deleting a subscription to a controller.
- After a communication error to a controller and when the communication is OK again.
- After overflow in the event subscriptions queue in the controller (the controller setting "AE Event subscription queue size") and when the queue again is filled to less than 75%.
- Five minutes after an alarm was added or removed from the controller, for example after application download.
- After overflow in the list of all alarms that are not idle (the OPC Server setting MaxNoOfAlarms) and when that list again is filled to less than 90%.
- After overflow in queue to OPC client (the OPC Server setting Queue Size) and when the queue is filled less than 75% again.

Before refresh of the OPC clients, the OPC Server refreshes itself first on all alarms that are not in an idle state. When the OPC Server is ready with its update, the refresh of OPC clients start. However, there is a delay of 30 seconds to avoid to many refreshes.

Performance

Response time to OPC clients will slow down considerable if more than 10 OPC clients are connected to the OPC server.

Sending alarms to OPC server is done with the limitations of MMS traffic, but are time stamped correctly.

Appendix A Running the OPC Server and OPC Client on Different Computers

If you are going to run the OPC server on one PC and the OPC client on another PC, the settings required for DCOM must be performed on the OPC Server PC.

DCOM should however only be used if absolutely needed, due to the risk for lost connection as described in Lost Connection Between OPC Client and OPC Server on page 68.

Firewall Settings for OPC Server

To establish a DCOM connection between OPC Server and OPC Client, both running on Windows 7, the Windows Firewall settings need to be modified on the OPC Server as described below:

1. Go to **Control Panel** > **System and Security** > **Windows Firewall**.
2. Click **Advanced Settings** to open the Windows Firewall with Advanced Security window.
3. Click **Inbound Rules**.
4. Click **New Rule**.
5. Use the **New Inbound Rule Wizard** to select and add the following programs (already installed on the PC) to allow the connection:
   - **OpcEnum** available in Windows\System 32 folder
   - **AC800MC_OpcServer** available in Program Files\
     ABB Industrial IT\Control IT\OPC Server for AC 800M 5.1\Bin folder.

These new Inbound Rules ensure that a proper connection is established between OPC Server and OPC Client.

Lost Connection Between OPC Client and OPC Server

Various problems may occur if an OPC Client, running on another PC than the OPC Server, is stopped abruptly (if, for example, the network between the OPC Client and Server is disconnected).

INDEX

Contact us

ABB AB
Control Systems
Västerås, Sweden
Phone: +46 (0) 21 32 50 00
Fax: +46 (0) 21 13 78 45
E-Mail: processautomation@se.abb.com
www.abb.com/controlsystems

ABB Inc.
Control Systems
Wickliffe, Ohio, USA
Phone: +1 440 585 8500
Fax: +1 440 585 8756
E-Mail: industrialitsolutions@us.abb.com
www.abb.com/controlsystems

ABB Industry Pte Ltd
Control Systems
Singapore
Phone: +65 6776 5711
Fax: +65 6778 0222
E-Mail: processautomation@sg.abb.com
www.abb.com/controlsystems

ABB Automation GmbH
Control Systems
Mannheim, Germany
Phone: +49 1805 26 67 76
Fax: +49 1805 77 63 29
E-Mail: marketing.control-products@de.abb.com
www.abb.de/controlsystems

Power and productivity
for a better world™