
**User's
Manual**

**DXA410
DAQOPC for DX/MV Series
User's Manual**

IM 04L01B03-61E

Foreword

This manual explains the functions and operations of DAQOPC. To ensure correct use, please read this manual thoroughly before beginning operation. After reading the manual, keep it in a convenient location for quick reference whenever a question arises during operation.

DAQOPC is an OPC server that provides OPC interface (Data Access interface). The OPC interface was designed by the OPC (OLE for Processing Control) Foundation. This manual describes the specifications of the Data Access interface and vendor-specific section. This manual is for those who understand the specifications of the OPC Foundation.

Notes

- The contents of this manual are subject to change without prior notice as a result of continuing improvements to the instrument's performance and functions.
- Every effort has been made in the preparation of this manual to ensure the accuracy of its contents. However, should you have any questions or find any errors, please contact your nearest YOKOGAWA dealer as listed on the back cover of this manual.
- Copying or reproducing all or any part of the contents of this manual without the permission of Yokogawa Electric Corporation is strictly prohibited.

Copyright

- YOKOGAWA holds the copyright to the software that is on the CD-ROM.

Trademarks

- Microsoft, MS-DOS, Windows, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- Adobe and Adobe Acrobat are trademarks of Adobe Systems Incorporated (Adobe Systems).
- Other company and product names are trademarks or registered trademarks of their respective companies.
- For purposes of this manual, the TM and !!! symbols do not accompany their respective trademark names or registered trademark names.

Revisions

1st Edition November 2000

Notes on Using This Product

Storing the CD-ROM

Please store the original CD-ROM in a safe place. During the actual operation, use the software that is installed on the hard disk.

Agreement

Restriction on Use

Use of this product (this utility software and manual) by more than one computer at the same time is prohibited. Use by more than one user is also prohibited.

Transfer and Lending

Transfer or lending of this product to any third party is prohibited.

Guarantee

Should a physical deficiency be found on the original CD-ROM or this manual upon opening the product package, please promptly inform Yokogawa. The claim must be made within seven days from the date you received the product in order to receive a replacement free of charge.

Exemption from Responsibility

Yokogawa Electric Corporation provides no guarantees other than for physical deficiencies found on the original CD-ROM or this manual upon opening the product package. Yokogawa Electric Corporation shall not be held responsible by any party for any losses or damage, direct or indirect, caused by the use or any unpredictable defect of the product.

Copyright

YOKOGAWA holds the copyright to the software that is on the CD-ROM.

Contents

Notes on Using This Product	ii
Chapter 1 Overview	1-1
1.1 Functional Overview of DAQOPC	1-1
1.2 Overall Configuration and DAQOPC Interface	1-5
1.3 Server Function	1-6
Chapter 2 DAQOPC Operation	2-1
2.1 Flow of DAQOPC Operation	2-1
2.2 Starting the Configurator and Setting Parameters	2-2
2.3 Starting and Stopping DAQOPC	2-7
2.4 Setting Up the DAQOPC Client	2-8
Chapter 3 DAQOPC DA Server Functions	3-1
3.1 Overview of the DAQOPC DA Server Function	3-1
3.2 OPC Interface Functions	3-2
3.3 Error Codes	3-19
Index	Index-1

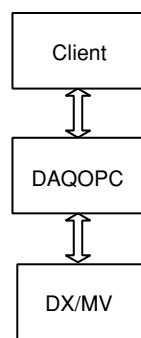
1.1 Functional Overview of DAQOPC

What Is an OPC Interface

By using the OPC interface provided by YOKOGAWA, the user is able to access various data on the DX/MV via the DAQOPC server using an OPC-compliant application that runs on a Windows machine. The user does not have to create a special application program.

The OPC interface is a standard interface that uses OLE (Object Linking and Embedding) for process control. It consists of a server that provides various data on the DX/MV and a general-purpose interface used to access the server.

OPC applies OLE, which is a standard method of communication between applications in the Windows environment, to process control. It allows the exchange of process data between multiple general-purpose Windows applications.

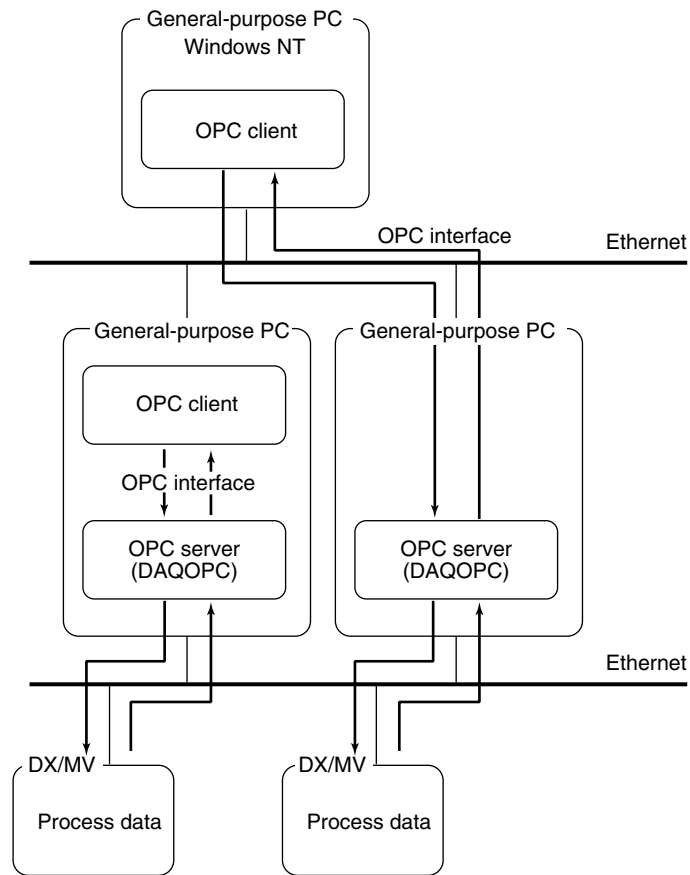


DAQOPC Functions

- Provides Data Access function (DA), the OPC interface.
DA function: Reading and writing of the current value of the process data that uses ItemID as the data identifier.
- Starts and stops the OPC server in sync with the start and shutdown of Windows NT.
- Supports the communication interfaces of the DX/MV which includes Ethernet, RS-232, and RS-422-A.
- Allows a data update rate of up to 1 s. However, data update may not be possible at the specified rate depending on the communication conditions.
- Capable of reading measured value, computed value, and alarm status as an ItemID.
- Capable of reading and writing communication input data as an ItemID (administrator level only).
- Allows access using DX/MV administrator and user levels.
- By setting the communication status with the connected device to an ItemID, the client is able to instruct a recovery after a power failure.
- Retrieves span value, unit, tag, alarm type, and alarm setting as Property IDs of the ItemID.
- The DAQOPC OPC server supports the OPC Data Access Version 1.0A and 2.0.
- The DAQOPC OPC server supports the OPC DA Custom Interface.
- The DAQOPC OPC server supports the OPC DA Automation Interface.
- The DAQOPC OPC server supports the optional OPC DA browser function.
(Browser function is used by the OPC client to browse the contents of the OPC server.)
- The DAQOPC OPC server can connect up to 24 DX/MVs.

Where OPC Interface Fits

OPC client refers to the application that requests data using the OPC interface. OPC server refers to the application that provides the data.

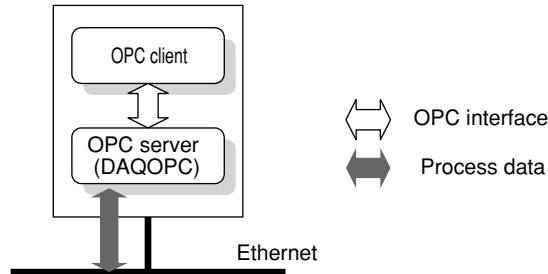


The OPC server and OPC client run on a general-purpose PC. You can use a commercially sold OPC-compliant application or an OPC-compliant application that you've created using VC++ (Visual C++), VB (Visual Basic), or VBA (Visual Basic for Application) as an OPC client.

Server/Client Configuration

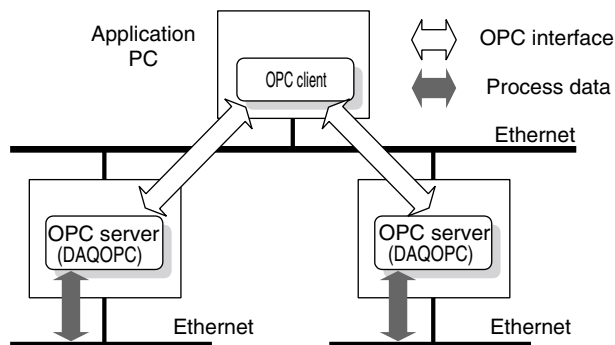
The DAQOPC user (OPC client) may exist on the same PC as the DAQOPC server or on a host computer (Windows NT machine).

When the OPC client exists on the same PC as the DAQOPC server



- Application size : Mid-size
- Performance : Faster than when the OPC client exists on a host computer, because a local OPC server is used.

When the OPC client exists on a host computer



- Application size: Mid- to large-size
- Performance : The performance degrades slightly (20% to 30%) per server when compared to when the OPC client exists on the same PC as the DAQOPC server. This is because the DCOM (Distributed Component Object Model) is used to connect between the client and server.

Hardware Requirements

The following hardware is required for DAQOPC to operate properly.

- PC model : IBM PC/AT compatible (that can run Windows NT).
- CPU : Pentium II 300MHz or faster recommended (Single CPU).
- Main memory : 128 MB or more recommended.
- Free disk space : 130 MB or more.
- Communication device : RS-232 port supported by the OS.
Ethernet NIC (not required if the OPC client and the DAQOPC server exist on the same PC, and the RS-232/RS-422-A interface is used to retrieve data from the DX/MV).
- Peripheral devices : Mouse supported by the OS.

Software Requirements

The following software is required for DAQOPC to operate properly.

- OS: Windows NT Workstation 4.0 or Windows NT Server 4.0 package (DAQOPC).
The OS (Windows NT) is of a same language.
- Service pack: Windows NT Service Pack 4, 5, or 6a.

Miscellaneous

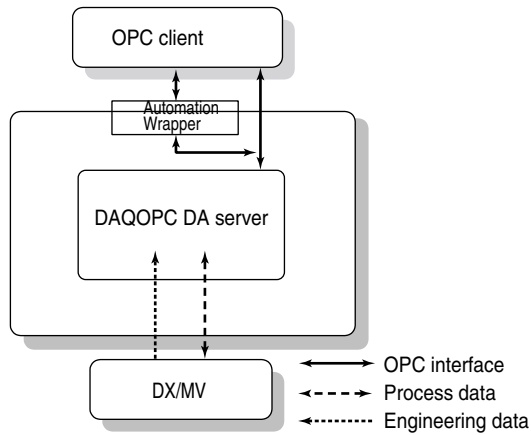
If the communication interface with the DAQSTATION is RS-422-A, an RS-232/RS-422-A converter is required.

1.2 Overall Configuration and DAQOPC Interface

This section will describe the overall construction of DAQOPC and an overview of DAQOPC functions.

Overall Construction of DAQOPC

The overall construction of DAQOPC is shown below. DAQOPC OPC server consists of a server that provides the interface specifications of the OPC DA.



Role of each function

An overview of the interface that DAQOPC supports is indicated.

Process Data		OPC Specification
Current value data	Read/write*	Complies with DA2.0

* Write is only for communication input data only.

Engineering Data		OPC Specification
Tag informatio	Load	Complies with DA2.0
Tag list	Load	Complies with DA2.0

1.3 Server Function

A list of DAQOPC server functions is shown below.

Function Name	Description
OPC DA custom interface	Functions that complies with the custom interface of the OPC specification.
OPC DA automation interface	Functions that complies with the automation interface of the OPC specification.

DA Server

Application capacity

The application capacity of the DAQOPC DA server is shown below.

Item	Application Capacity
Maximum number of clients	100 clients
Maximum number of group objects	1,000 groups
Maximum number of registered ItemIDs	10,000/group
Maximum number of cache update ItemIDs	100,000
Cache update rate (UpdateRate)	1,000 to 3,600,000 ms (1 s to 1 h)

Server name (ProgID)

The server name of the DAQOPC DA server is as follows:

Server name: **Yokogawa.ExaopcDADAQOPCDXMV1**

Note

You can also check the server name (ProgID) in the Version Information dialog box of the DAQOPC configuration window.

OPC Custom Interface and OPC Automation Interface

The OPC specification defines two types of interfaces: OPC custom interface and OPC automation interface. Both of these interfaces have approximately the same data access functions, but they are designed for different client programs.

The DAQOPC OPC server implements both interfaces and supports both types of client programs.

	Custom Interface	Automation Interface
Application	For dedicated applications such as SCADA/MES/analysis software programs	For easy access from script languages
Number of client languages	VC++	Visual Basic
Performance	Good	Poor

The OPC custom interface uses the basic OLE/COM functions directly and operates at a high speed.

In contrast, the OPC automation interface complies with the OLE automation interface that allows access from Visual Basic. The process that facilitates the access from script languages leads to an overhead, and, therefore, the performance is slightly lower than the OPC custom interface.

VB and VC++ versions

If you are creating an application program (OPC client), use the following versions of VB and VC++.

		DAQOPC and User Application on the Same PC	DAQOPC and User Application on Different PCs.
DAQOPC	R1.01	VB6.0, VC++6.0	VB5.0/6.0, VC++5.0/6.0

If you are creating a program using VB, select the Yokogawa Exaopc Data Access Automation Server check box by choosing Project > References.

2.1 Flow of DAQOPC Operation

This section will describe the operations of DAQOPC. For the access method from the OPC client software, see chapter 3, “DAQOPC DA Server Function.”

DAQOPC refers to the setup file at startup and logs onto the DX/MV. The DAQOPC configuration window provides easy-to-use user interface (like a typical Windows application) for displaying and setting the communication parameters of the DX/MV from which DAQOPC is to retrieve data. In addition, the DAQOPC configuration window shows information that is required for the client to connect to DAQOPC (ProgID).

Flow of Operation

The general flow of operation is as follows:

1. Determine the method of connection with the DAQSTATION from which to retrieve the data using the Configurator.
2. Start DAQOPC.
3. Set up the client software if the client is on a different PC.

Changing the Parameter during DAQOPC Operation

You can change system parameters on the DAQOPC configuration window even when DAQOPC is in operation. However, the specified parameters take effect the next time DAQOPC is started.

2.2 Starting the Configurator and Setting Parameters

Configurator

Configurator is used to set the communication conditions between DAQOPC and the DAQSTATION from which to retrieve the data.

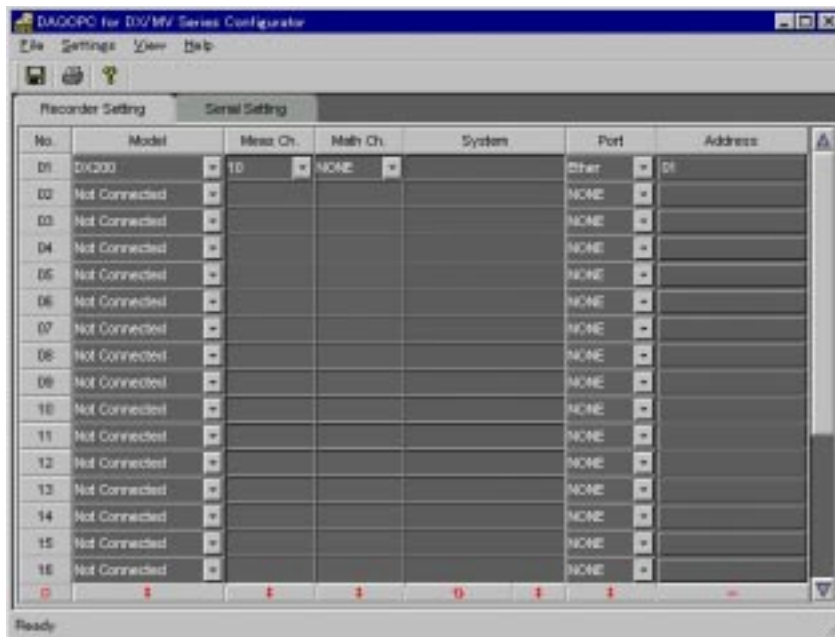
Starting the Configurator

Choose **Start > Programs > DAQOPC for DX_MV > DAQOPC for DX MV Configurator > DAQOPC Configurator**. The window that appears when the program is started is shown below.



Configuring the Recorder

The Recorder Setting tab page is shown below.



The following items are specified on the Recorder Setting tab page.

- Model :Select the type of DX/MV to be connected.
- Meas. Ch :Select the number of measurement channels of the DX/MV to be connected.
- Math Ch :Select the number of computation channels of the DX/MV to be connected.
- System :Not used currently.
- Port :Select the port to be connected. For COM, only the ports that are valid in the serial settings are displayed.
- Address :Only the COM ports that are set to Ethernet or RS-422-A are valid.

When the port is an Ethernet port

Click the address to display the following dialog box.



Specify the following parameters.

- IP Address or Host name :Enter the IP address or host name that is assigned to the DX/MV that is to be connected.
- Login name :Specify the login name.
- Password :Specify the password.
- Login level :Specify the level for the specified login name.

If the parameters are not correct, connection to the DX/MV fails.

If the login level is User, communication input data cannot be specified as an ItemID of DAQOPC.

2.2 Starting the Configurator and Setting Parameters

When the port is COM

Click the address to display the following dialog box.



Set the RS-422-A address.

Serial Setting Tab Page

The following items are specified on the Serial Setting tab page.



Port No. :Select the check box corresponding to the port to be used.

Type :Set the type of port to be used.

Baud Rate :Select the baud rate for the port to be used.

ParityBit :Set the parity for the port to be used.

StopBit :Select the stop bit for the port to be used.

A List of Menu Commands

The functions of menu commands are indicated below.

File > Reload

Reloads the setup file without saving the information that you are currently editing.

File > Save Settings

Saves the information that you are currently editing to the setup file.

File > Print

Prints the recorder settings and serial settings that you are currently editing.

File > Print Preview

Prints the preview of the recorder settings and serial settings that you are currently editing.

File > Printer Setup

Sets the printer.

File > Exit

Exits the Configurator.

Setting > Operation Settings

Displays the Operation Settings dialog box.

View > Recorder Settings

Switches to the Recorder Setting tab.

View > Serial Settings

Switches to the Serial Setting tab.

View > Toolbar

Shows or hides the toolbar.

View > Status Bar

Shows or hides the status bar.

Help > Version Information

Displays the Version Information dialog box.

2.2 Starting the Configurator and Setting Parameters

Working Condition Dialog Box

Set the following items on the Working Condition dialog box.



Time Stamp

Select whether to use the time of the connected recorder or the time of the PC in which DAQOPC is installed for the time stamp that is added when the data is retrieved.

Execute by demo mode

If you select the Demo Mode check box, DAQOPC does not perform communications with the DX/MV and operates as a simulator.

Handling of special data values

Assigns values to special data that is retrieved from the DX/MV. For values over or under the range, you can specify a particular value or assign upper and lower limits of the span. You can also assign values for skip, error, and undefined data. You can assign a value between -100000000 and 1000000000. A decimal point is added to the appropriate position to the value that is actually retrieved.

Handling of data during communication error

Specify the handling of the value when an error occurs while data is being retrieved. If the previous value is used, the value before the communication error is retrieved. If you are specifying a particular value, it is the same as the handling of special data values. If a communication error occurs when DAQOPC initially connects to the DX/MV, the value is always set to 0.

Version Information Dialog Box

The Version Information dialog box shows the following information.

- Version information
- ProgID



2.3 Starting and Stopping DAQOPC

Starting DAQOPC

The DA server process automatically starts as a background process when the Windows NT system is started. At this point, DAQOPC starts communications with the DAQSTATION according to the communication conditions that were specified using the Configurator.

Stopping DAQOPC

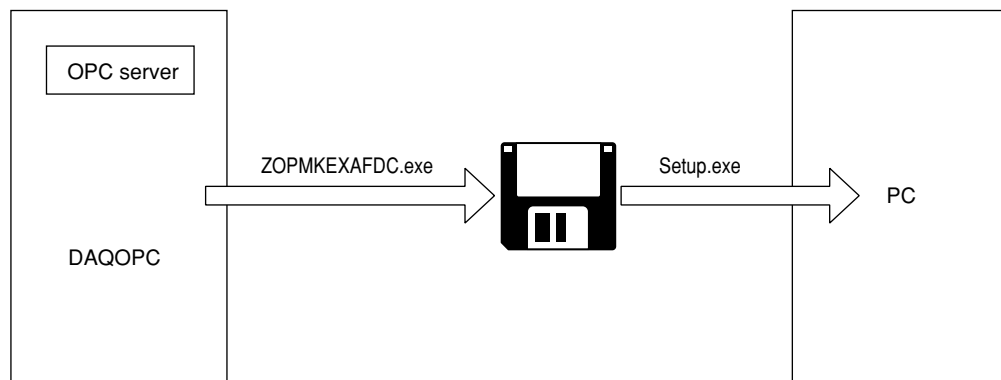
The DA server process automatically stops when the Windows NT system is shut down.

2.4 Setting Up the DAQOPC Client

If you are using DAQOPC remotely from a PC that is connected to the network, you must setup the DAQOPC client.

If you have not set up the DAQOPC client, follow the procedure below to set up the software.

1. Insert a formatted floppy disk into the floppy drive of the PC (PC server) in which you have installed DAQOPC.
2. On the server PC, run Zopmkexafdc.exe in the Program folder in the installation folder of the DAQOPC using a program such as NT Explorer.
3. A dialog box appears when an OPC client setup disk is created.
Click **OK** in the dialog box.
4. Log on to the PC (PC client) in which to setup the OPC client using a user name belonging to the Administrator group.
5. Insert the setup floppy disk that you have just created into the floppy drive of the client PC.
6. Run Setup.exe on the setup floppy disk. The OPC client is set up.



Access Method from the Client Software

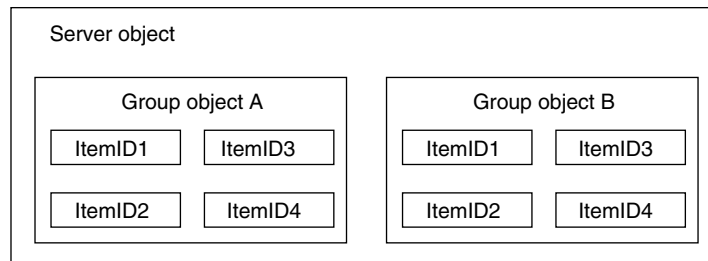
The ProgID used in accessing DAQOPC from the client software is as follows:

Yokogawa.ExaopcDADAQOPCDXMV1

3.1 Overview of the DAQOPC DA Server Function

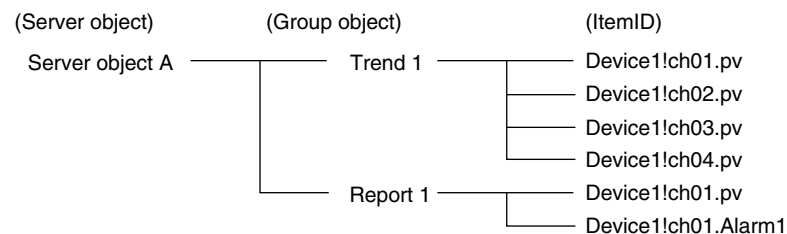
Data Access Method via the OPC Interface

Through the OPC Interface, the OPC client creates a group object in the server object and registers the process data to be accessed as an ItemID in the group object. ItemIDs are registered in the form “tag, tag item.” Using this scheme, you can specify the data acquisition rate for each group object. You can also register the same ItemID in different group objects according to your application.



In the example below, two group objects, Trend 1 and Report 1, are created in the server object A. Device1!ch01.PV through ch04.PV are registered as ItemIDs in the Trend 1 group object. Device1!ch01.PV and Device1!ch04.Alarm1 are registered as ItemIDs in the Report 1 group object.

Trend 1 and Report 1 are accessed as independent group objects. In addition, the value of Device1!ch01.PV exists in Trend 1 and Report 1 in this example.



OPC Interface Compliance

The DAQOPC DA server is an OPC server that provides the interface complying with the following OPC DA Specification Version 2.0 and Version 1.0A.

- OPC DA Custom Interface
- OPC DA Automation Interface

It supports all the standard functions that are defined in these OPC Specifications.

In addition, those items that are defined as options (browse function of items) are also supported.

Note

For details on the OPC Custom Interface and OPC Automation Interface, see the corresponding sections in section 1.3, “Server Functions.”

3.2 OPC Interface Functions

The DAQOPC DA Server supports the standard interface of OPC Version 2.0. It also supports a portion of the interface that is defined to be options. For clients that use the older version of the OPC interface, functions that were supported in OPC Version 1.0A are also supported.

This section will mainly focus on the vendor-specific option interface of the DAQOPC DA Server.

For details on the OPC Interface, see the specifications that are released by the OPC Foundation.

A List of Interfaces

A list of interfaces that are supported is indicated for the OPC Custom Interface and OPC Automation Interface.

Custom Interface

The table below indicates whether or not the DAQOPC DA Server supports various interfaces of the Custom Interface.

Object	Interface Name	Description	Support
OPCServer	IOPCServer	A standard interface of the OPC DA server. Adds and Deletes group objects.	Yes
	IOPCCommon	OPC common interface. Sets the locale and retrieves error strings.	Yes
	IConnectionPointContainer	OPC common interface. Interface for asynchronous communication used to notify the client that the server is going to shut down.	Yes
	IOPCItemProperties	Interface. Retrieves the properties of the ItemID.	Yes
	IOPCBrowseServerAddress Space(optional)	Browses the ItemIDs in the OPC DA server.	Yes
	IOPCServerPublicGroups (optional)	Connects to the public group; deletes the public group.	No
	IPersistFile (optional)	Loads and saves configuration information.	No
OPCGroup	IOPCGroupStateMgt	Manages the group object.	Yes
	IOPCItemMgt	Manages the ItemIDs.	Yes
	IOPCSyncIO	Performs synchronous write and read operations.	Yes
	IOPCAsyncIO2	Performs asynchronous write and read operations.	Yes
	IConnectionPointContainer	Interface for creating connections for asynchronous read and write operations.	Yes
	IOPCPublicGroupStateMgt (optional)	Convert a private group to a public group.	No
	IOPCAsyncIO (old)	Performs asynchronous write and read operations (old interface for compatibility).	Yes
	IDataObject (old)	Performs connection and disconnection to the asynchronous I/O (old interface for compatibility).	Yes
EnumOPCItem Attributes	IEnumOPCItemAttributes	Retrieves a list of ItemIDs that are registered in the group object.	Yes

Yes: Supported, No: Not supported

Automation Interface

The DAQOPC DA Server only supports the automation interfaces that correspond to the custom interfaces that are supported. The table below shows whether or not the interface is supported. (Since those that are not supported may result in error, error handling for Visual Basic must be provided.)

Object	Type	Name	Description	Support
OPCServer	Properties	StartTime	Time when the server was started	Yes
		CurrentTime	Current time	Yes
		LastUpdateTime	Time that was notified last	Yes
		MajorVersion	Major version	Yes
		MinorVersion	Minor version	Yes
		BuildNumber	Build number	Yes
		VendorInfo	Vendor information	Yes
		ServerState	Server state	Yes
		LocaleID	Locale ID	Yes
		Bandwidth	Bandwidth	Yes
		OPCGroups	Collection of group objects	Yes
		PublicGroupNames	Public group name	No
		ServerName	Server name	Yes
		ServerNode	Node name	Yes
	ClientName	Client name	Yes	
	Methods	GetOPCServers	Gets a list of server names.	Yes
		Connect	Established connection.	Yes
		DisConnect	Releases connection.	Yes
		CreateBrowser	Creates a browser object.	Yes
		GetErrorString	Gets the error string.	Yes
		QueryAvailbleLocaleIDs	Gets a list of supported locale IDs.	Yes
		QueryAvailableProperties	Gets a list of properties.	Yes
		GetItemProperties	Gets properties.	Yes
LookupItemIDs	Converts the ItemID of properties.	No		
Events	ServerShutDown	Shutdown event	Yes	

Yes: Supported, No: Not supported

3.2 POPC Interface Functions

Object	Type	Name	Description	Support
OPCBrowser	Properties	Organization	Hierarchical type	Yes
		Filter	Filter specification	Yes
		DataType	Data type	Yes
		AccessRights	Access rights	Yes
		CurrentPosition	Current position	Yes
		Count	Number of collections	Yes
	Methods	Item	Gets the object.	Yes
		ShowBranches	Gets a list of current positions.	Yes
		ShowLeafs	Gets a list of current positions.	Yes
		MoveUp	Moves the hierarchy.	Yes
		MoveToRoot	Moves the hierarchy.	Yes
		MoveDown	Moves the hierarchy.	Yes
		MoveTo	Gets the absolute position.	Yes
		GetItemID	Gets the ItemID.	Yes
GetAccessPaths	Gets access paths.	Yes		
OPCGroups	Properties	DefaultGroupsActive	Default active property	Yes
		DefaultGroupUpdateRate	Default cache update	Yes
		DefaultGroupDeadband	Default deadband	Yes
		DefaultGroupLocaleID	Default locale ID	Yes
		DefaultGroupTimeBias	Default time bias	Yes
		Count	Number of collections	Yes
	Methods	Item	Gets the group object.	Yes
		Add	Adds a group.	Yes
		GetOPCGroup	Gets the group object.	Yes
		Remove	Deletes the group.	Yes
		RemoveAll	Deletes all groups.	Yes
		ConnectPublicGroup	Gets the public group.	No
		RemovePublicGroup	Deletes the public group.	No

Yes: Supported, No: Not supported

Object	Type	Name	Description	Support
OPCGroup	Properties	Name	Group name	Yes
		IsPublic	Public property (always FALSE)	No
		IsActive	Active property	Yes
		IsSubscribed	Asynchronous property	Yes
		ClientHandle	Client handle	Yes
		ServerHandle	Server handle	Yes
		LocaleID	Locale ID	Yes
		TimeBias	Time bias	Yes
		DeadBand	Deadband	Yes
		UpdateRate	Cache update rate	Yes
		OPCItems	Item collection	Yes
	Methods	SyncRead	Performs a synchronous read.	Yes
		SyncWrite	Performs a synchronous write.	Yes
		AsyncRead	Performs an asynchronous read.	Yes
		AsyncWrite	Performs an asynchronous write.	Yes
		AsyncRefresh	Refreshes.	Yes
		AsyncCancel	Cancels asynchronous request.	Yes
	Events	DataChange	Change notification	Yes
		AsyncReadComplete	Asynchronous read response	Yes
		AsyncWriteComplete	Asynchronous write response	Yes
AsyncCancelComplete		Cancel response	Yes	

Yes: Supported, No: Not supported

3.2 POPC Interface Functions

Object	Type	Name	Description	Support
OPCItems	Properties	Parent	Parent group object	Yes
		DefaultRequestedDataType	Default requested data type	Yes
		DefaultAccessPath	Default access path	No
		DefaultActive	Default active property	Yes
		Count	Number of collections	Yes
	Methods	Item	Gets the item object.	Yes
		GetOPCItem	Gets the item object.	Yes
		Remove	Removes the item object.	Yes
		Validate	Checks the item.	Yes
		SetActive	Sets the active property.	Yes
		SetClientHandles	Sets the client handle.	Yes
		SetDataTypes	Sets the data type.	Yes
		AddItem	Adds an item.	Yes
		AddItems	Adds multiple items.	Yes
OPCItem	Properties	Parent	Parent group object	Yes
		ClientHandle	Client handle	Yes
		ServerHandle	Server handle	Yes
		AccessPath	Access path	No
		AccessRights	Access rights	Yes
		ItemID	ItemID	Yes
		IsActive	Active property	Yes
		RequestedDataType	Requested data type	Yes
		Value	Data value	Yes
		Quality	Quality flag	Yes
		TimeStamp	Time stamp	Yes
		CanonicalDataType	Data type assigned by the server	Yes
		EUType	Engineering unit type	Yes
		EUInfo	Engineering unit information	Yes
		Methods	Read	Performs a synchronous read.
	Write		Performs a synchronous write.	Yes

Yes: Supported, No: Not supported

Process Data

The process data of the OPC specification consists of the following three elements.

- Value
- Quality Flag
- Time Stamp

Value

OPC uses a data type called VARIANT.

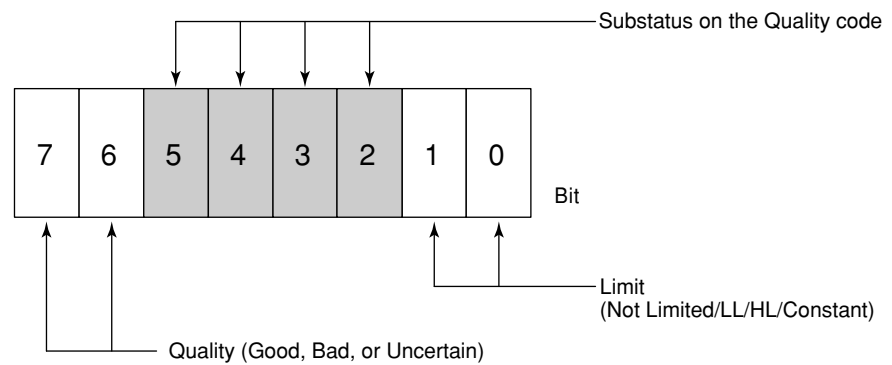
Quality Flag

This flag indicates whether the Value was retrieved correctly. Notifies the reason if erroneous.

The quality flag of DAQOPC follows the system of the OPC specifications. The quality flag system of the OPC specifications consists of the following three contents.

- Quality (Good, Bad, or Uncertain)
- Substatus (details dependent on the Quality)
- Limit (additional information for the substatus)

The quality flag is expressed using the following 8 bits.



DAQOPC sets these three types of information as a quality flag.

3.2 POPC Interface Functions

Quality and Substatus

The Substatus of the Quality code contains information that can be expressed using 4 bits for each Quality (Good, Bad, and Uncertain). The Substatus codes are defined by the OPC specifications. DAQOPC uses these codes to add details. The table below shows the information that the OPC specifications define.

Code	Substatus for Each Data Quality Code		
	Good(3)	Bad(0)	Uncertain(1)
0	Non-specific	Non-specific	Non-specific
1	N/A	Configuration Error	Last Usable Value
2	Local Override	Not Connected	N/A
3		Device Failure	Sensor Not Accurate
4		Sensor Failure	EngineeringUnitsExceeded
5		Last Known Value	Sub-Normal
6		Comm Failure	
7		Out of Service	
8			
9			
10			
11			
12			
13			
14			
15			

Limit

Limit is yet another Quality code information that is added to the combination of Quality and Substatus.

Code	Description
0	Not Limited
1	Low Limited
2	High Limited
3	Constant

Data and Quality code

The data from the DX/MV is converted to the following Quality codes.

Data status	Quality	Substatus	Limit
Normal value	Good	Non-specific	Not Limited
Positive overrange value	Good	Non-specific	High Limited
Negative overrange value	Good	Non-specific	Low Limited
Skip or computation Off	Bad	Out of Service	Not Limited
Error value	Bad	Sensor Failure	Not Limited
Uncertain value	Bad	Out of Service	Not Limited
Communication error	Bad	Comm Failure	Not Limited

By assigning values other than normal values using the Configurator, you can check the data from the DX/MV even without the Quality code.

Time Stamp

Retrieves either the time of the DX/MV or the time of the PC. Using the Configurator, you can change the destination from which to retrieve the time.

Selection on the Configurator	Description
Time of the device (DX/MV)	Retrieves the time of the device.
Time of the PC	Retrieves the time of the PC in which the Server for retrieving the data is installed in UTC.

ItemID

An arbitrary string used to identify items. It is generally called tags.

Defining the ItemID

DAQOPC defines ItemIDs as follows:

Device!tag number.data item

Example: DEVICE1!CH001.PV

The "DEVICE" section is fixed. In the number section that follows, specify the device position that you assigned using the Configurator.

A list of ItemIDs

On DAQOPC, the following items can be accessed using ItemIDs.

ItemID	Description	Canonical data type	Access Rights	Notes
DEVICE!CHxxx.PV	Measured value or computed result	VT_R8	Read Only	Measurement and computation channels only
DEVICE!CHxxx.ALARMx	Alarm status of the specified channel	VT_I2	Read Only	Measurement and computation channels only
DEVICE!CHcxx.INPUT	Value of the communication input data	VT_R8	Read/Write	Communication input data only
DEVICE!COMMON.STATUS	Device status	VT_I2	Read/Write	
DEVICE!COMMON.PROP	Property read status	VT_I2	Read/Write	

3.2 POPC Interface Functions

The retrievable process values are the following two values.

- **Measured value and computed value**

Syntax Device name!channel number.pv

Example: Device1!Ch01.PV

For the device name, enter the name of the device that you specified using the Configurator

- **Alarm value**

Syntax Device name!channel number.ALARM1

Example: Device1!Ch01_ALARM1

You can specify ALARM1 through ALARM4 for the alarm.

Selectable values are as follows:

- **Input value**

Syntax Device name!communication input data number.INPUT

Example: Device1!ChC01.INPUT

Add the letter C as in ChC01 for the communication input data.

Item name PV

Specifiable tag: Existing measurement and computation channels only

Item name ALARM

Specifiable tag: Existing measurement and computation channels only

Item names are ALARM1, ALARM2, ALARM3, and ALARM4.

The alarm value is 0 if there is no alarm occurring. The value is set to the alarm type (alarm type value 1 through 8 as written in property ID: 6004), which is an alarm item property, if an alarm is occurring (see page 2-14).

Item name INPUT

Specifiable tag: Existing communication input data only

The input range is the range that can be input to the DX/MV.

The server does not check whether the input value is within the allowed input range.

Item name STATUS

Specifiable tag: COMMON

Indicates the status of the specified device (see below).

Operating normally.	0
The specified device does not exist.	1
Failed to allocate memory area.	2
Failed to open communications.	3
Communication error.	4

When the status is 3 and 4, and 0 is written, the DAQOPC server attempts to recover the communication of the specified device.

Writing 0 when the status is 0, 1, or 2 is void.

Item name PROP

Specifiable tag: COMMON

Indicates the property read status of the specified device (see below).

The properties of the specified device have not been read.	0
The properties of the specified device have been read.	1
Performs a read operation to the specified device.	2

When 2 is written, the properties are retrieved from the specified device. Normally, the properties are read when the DAQOPC is started. When a retrieve request for a property value is received, the server returns a value based on this value rather than actually requesting the device. In addition, the server uses the decimal position information of the measured and computed values that are retrieved at the start of DAQOPC. Therefore, if you change the range and other settings on the specified device, make sure to execute a reread.

Access path

OPC defines access path for distinguishing the COM (Component Object Model) ports, but the DAQOPC DA Server does not use it.

Blob

OPC defines a parameter called Blob to increase the efficiency of ItemID searches, but the DAQOPC DA Server does not use it.

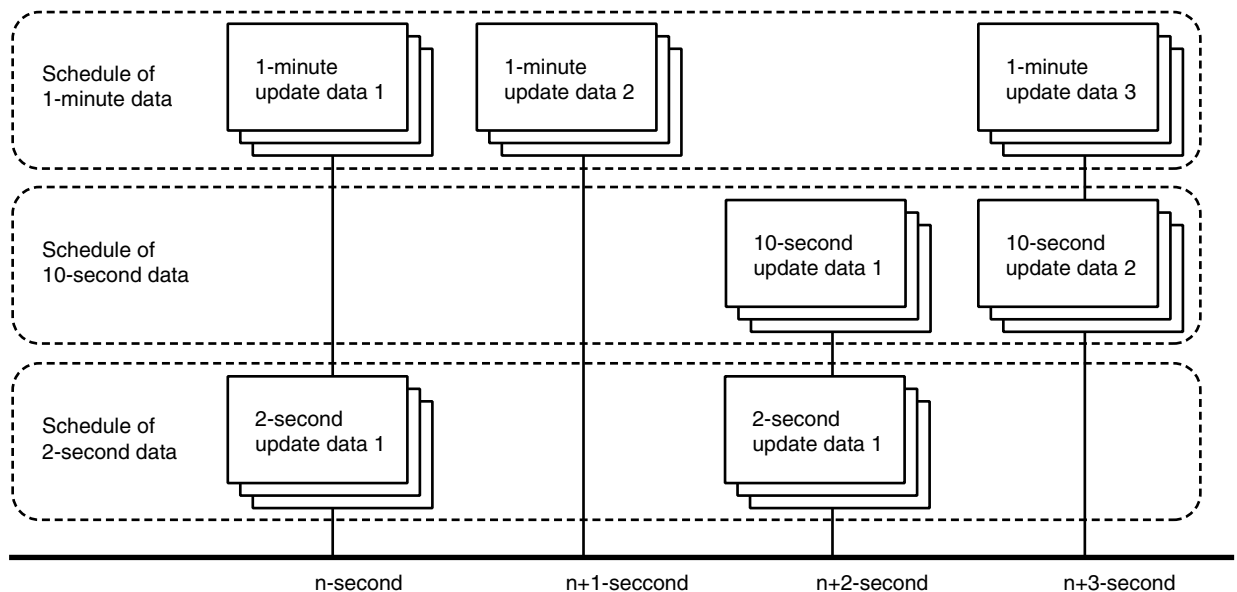
Data acquisition operation

The data acquisition operation complies with the OPC specifications. However, if the acquisition request from the client software exceeds the specifications of the maximum number of data points of DAQOPC, the process is temporarily suspended. Therefore, the desired response will not be obtained.

Cached data update

The cached data is scheduled in units of the update rate as follows:

1. The cached data is divided into update rate communication units (the maximum number of data points that can be transferred each time: 7968).
2. If the data and the update rate are the same, it is scheduled once to avoid duplication.
3. Each communication unit is scheduled as follows:
 - Each communication unit maintains its update rate.
 - Scheduling is dispersed so that duplication of communication units within the same second with the same update rate is avoided as much as possible up to a range of one minute.
 - Scheduling is performed so that duplication within the same second with communication units of other update rates is minimized.
 - The communication unit of each update rate is scheduled with the start point at each 0 s of the system time. If multiple communication units of the same update rate exist, they are scheduled in order every 1 s with the update rate as the maximum width.



Data change notification

DAQOPC rules regarding the data change check for the data change notification are given below.

- For data of which the type is VT_R8 (measured and computed values (PV)), if the ratio of the difference between the current cached data and the new value is greater than the deadband, cached data is updated.
- For data of which the type is VT_I2 (alarm status (ALARMx), device status (STATUS), and property read status (PROP)), if there is a difference between the current cached data and the new value, cached data is updated.

Group Object Properties (IOPCGroupStateMgt)

“Groups” are provided as means by which the OPC client can efficiently access the process data via the DA server. The OPC client is able to register arbitrary number of items in a group, and, in general, this group is handled as a unit for process data access. The handling of group object properties that are vender-specific is indicated below.

Group name

For the DAQOPC DA Server, if no name is specified when creating a group object, a group object with the following name is created.

Name of the created group: **CSGroup1, CSGroup2, ..., CSGroupN**

where **N** is an integer string.

Item Properties (IOPCItemProperties)

Item properties are property values that can be retrieved by specifying the ItemID and property ID. The following types of data can be acquired.

Specifying the data item PV (measurement channel and computation channel)

ItemID is specified as follows:

Device name!measurement or computation channel.PV

Example: DEVICE1!CH001.PV

Property ID	VARIANT type	Description
1	VARTYPE	“Item Canonical Data Type” Allows retrieval of VARIANT data type of items.
5	VT_I4	“Item Access Rights” 1: Read only. Access level of item.
6	VT_R4	“Server Scan Rate” Example “250” Allows retrieval of the DX/MV scan rate in the ms order.
100	VT_BSTR	“EU Unit” Example “°C” Allows retrieval of the engineering unit specified on the DX/MV.
102	VT_R8	“High EU” Example “2.0000” Allows retrieval of the upper limit of the span or scale that is specified on the DX/MV.
103	VT_R8	“Low EU” Example “-2.0000” Allows retrieval of the lower limit of the span or scale that is specified on the DX/MV.

3.2 POPC Interface Functions

Specifying the data item ALARM (measurement channel and computation channel)

ItemID is specified as follows:

Device name!measurement or computation channel.ALARM1 or ALARM 2-4

Example: DEVICE1!CH001.ALARM1

Property ID	VARIANT type	Description
1	VARTYPE	<p>“Data Type”</p> <p>Allows retrieval of VARIANT data type of items.</p>
5	VT_I4	<p>“Item Access Rights”</p> <p>1: Read only.</p> <p>Access level of item.</p>
6	VT_R4	<p>“Server Scan Rate”</p> <p>Example “250”</p> <p>Allows retrieval of the DX/MV scan rate in the ms order.</p>
100	VT_BSTR	<p>“EU Unit”</p> <p>Example “++”</p> <p>Allows retrieval of the engineering unit specified on the DX/MV.</p>
6004	VT_I2	<p>“Alarm Type”</p> <p>0: Alarm OFF 5: RH alarm 1: H alarm 6: RL alarm 2: L alarm7: DelayH alarm 3: DH alarm 8: DelayL alarm 4: DL alarm</p> <p>Allows retrieval of the type of alarm specified on the DX/MV. If the range is set to SKIP or the computation channel is OFF, Alarm OFF is output for all channels.</p>
6005	VT_R8	<p>“Alarm Trigger”</p> <p>Example “1.583”</p> <p>Allows retrieval of the alarm trigger value specified on the DX/MV. It is set to 0 when the alarm is OFF.</p>

Specifying the data item INPUT (communication input data)

ItemID is specified as follows:

Device name!communication input data.INPUT

Example: DEVICE1!CHC01.INPUT

Property ID	VARIANT type	Description
1	VARTYPE	"Data Type" Allows retrieval of VARIANT data type of items.
5	VT_I4	"Item Access Rights" 3: Read and write. Access level of item.
102	VT_R8	"High EU" Allows the retrieval of the upper limit that can be input.
103	VT_R8	"Low EU" Allows the retrieval of the lower limit that can be input.

Specifying the data item STATUS (COMMON)

ItemID is specified as follows:

Device name!COMMON.STATUS

Example: DEVICE1!COMMON.STATUS

Property ID	VARIANT type	Description
1	VARTYPE	"Data Type" Allows retrieval of VARIANT data type of items.
5	VT_I4	"Item Access Rights" 3: Read and write. Access level of item.

Specifying the data item PROP (COMMON)

ItemID is specified as follows:

Device name!COMMON.PROP

Example: DEVICE1!COMMON.PROP

Property ID	VARIANT type	Description
1	VARTYPE	"Data Type" Allows retrieval of VARIANT data type of items.
5	VT_I4	"Item Access Rights" 3: Read and write. Access level of item.

3.2 POPC Interface Functions

Specifying the tag number (measurement channel)

ItemID is specified as follows:

Device name!measurement channel

Example: DEVICE1!CH001

Property ID	VARIANT type	Description
101	VT_BSTR	"Item Description" Example "Funace 1" Allows retrieval of the tag strings specified on the DX/MV.
6003	VT_BSTR	"Input Range" Example "TC, TypeK" Allows the retrieval of range strings of measurement channels.

Specifying the tag number (computation channel)

ItemID is specified as follows:

Device name!computation channel

Example: DEVICE1!CH031

Property ID	VARIANT type	Description
101	VT_BSTR	"Item Description" Example "Funace 1" Allows retrieval of the tag strings specified on the DX/MV.

Specifying the tag number (communication input data)

There are no item properties that can be retrieved.

Specifying the tag number (COMMON)

There are no item properties that can be retrieved.

Specifying the device

ItemID is specified as follows:

Device name

Example: DEVICE1

Property ID	VARIANT type	Description
6001	VT_BSTR	"Model" Example "DX120" Allows the retrieval of the DX/MV model on the DX/MV.
6002	VT_I2	"Login Level" 1: Administrator level 2: User level Allows the retrieval of the login level to the DX/MV.

Locale ID (LCID)

The locale ID (LCID) is a language identifier that is used when the OPC server returns values as strings. It is used in alarms, status, and other parameters that include engineering units.

In general, the DAQOPC DA server ignores the locale ID.

The unit and tag strings of the DX/MV are passed without any modification.

For error strings (**IOPCServer::GetErrorString**, **IOPCCommon::GetErrorString**), the DAQOPC Server only supports the locale or the Windows NT system. However, all error strings that are defined by the OPC are returned in English (Windows NT system errors are returned in the local language).

Structures Defined by the OPC

The DAQOPC DA server handles the vendor-specific section of the structures that are defined by the OPC specifications as follows:

Item property structure (OPCITEMATTRIBUTES)

- Engineering Unit

The engineering unit is the range information of the item ID.

Server status structure (OPCSERVERSTATUS)

- Bandwidth

On the DAQOPC DA server, the bandwidth is set to the number of ItemIDs that are registered in the cache update.

- VendorInfo

“Yokogawa Electric Corporation” is set on the DAQOPC server.

Support for OPC Optional Specifications

The OPC optional functions that DAQOPC supports are described below.

Browsing (retrieves a list of ItemIDs in the server)

- **Hierarchy**

IOPCBrowseServerAddressSpace is an interface used to retrieve a list of tags. There are two methods available for viewing the server-side list, FLAT or HIERARCHICAL. For the DAQOPC, use the HIERARCHICAL method specifying the device name, tag number, and data item in that order.

- **ItemID retrieval through GetItemID**

The following string is returned for GetItemID.

- When a device name is specified

Device name

Example: Device1

(Cannot be used in IOPCItemMgt::AddItems)

- When a tag number is specified

Device name!tag number

Example: Device1!ch01

(Cannot be used in IOPCItemMgt::AddItems)

- When a data item is specified

Device!tag number.data item

Example: Device1!ch01.PV

(Can be used in IOPCItemMgt::AddItems)

3.2 POPC Interface Functions

- BrowseOPCItemIDs filter

If you specify OPC_FLAT for the filter type, the operation is the same as when both OPC_BRANCH and OPC_LEAF are specified. Sorting is not performed on the list, and there is no limit in the number of data points that can be entered in the list.

Therefore, the larger the number of tag numbers, more memory space is consumed. In such case, you can use the following types of filters to suppress the amount of memory that is consumed at any given time.

[szFilterCriteria] : DAQOPC ignores this parameter.
[dwAccessRightsFilter] : DAQOPC supports this parameter.

- BrowseAccessPaths

Since AccessPaths is not used, this is not supported.

3.3 Error Codes

The error codes that DAQOPC specifies are shown below.

Number	Message	Description
0xC0049001	"Fail to allocate memory."	Failed to allocate memory area.
0xC0049002	"Fail to read configuration file."	Failed to load the communication setup file.
0xC0049003	"Selected device doesn't exist."	A nonexistent device is specified by the Configurator.
0xC0049004	"Fail to open device."	Failed communications at startup.
0xC0049005	"Communication error."	A communication error occurred.
0xC0049006	"Irrelevant value."	Invalid value was input for communication input data.
0xC0049007	"Fail to get time stamp from PC."	Failed to get the time stamp from the PC.

Index

A

Access path 3-11
Automation Interface 3-3

B

Blob 3-11
Browsing 3-17

C

Cached data update 3-11
Custom Interface 3-2

D

DA Server 1-6
DAQOPC configuration window 2-1
Data Access Method via the OPC Interface 3-1
Data acquisition operation 3-11
Data change notification 3-12
Defining the ItemID 3-9

E

Error Codes 3-19

F

Flow of DAQOPC Operation 2-1

G

Group name 3-12
Group Object Properties 3-12

H

Hardware Requirements 1-4
Hierarchy 3-17

I

Item Properties 3-13
Item Properties (IOPCItemProperties) 3-13
Item property structure 3-17
ItemID 3-9

L

Locale ID 3-17

O

OPC Automation Interface 1-7
OPC Custom Interface 1-7
OPC interface 1-1
OPC Interface Compliance 3-1
OPC Interface Functions 3-2
Overall Construction of DAQOPC 1-5

P

Process Data 3-7

Q

Quality Flag 3-7

S

Server Function 1-6
Server status structure 3-17
Server/Client Configuration 1-3
Setting Up the DAQOPC Client 2-8
Software Requirements 1-4
Starting and Stopping DAQOPC 2-7

T

Time Stamp 3-7, 3-9

W

What Is an OPC Interface 1-1