Experion PKS

OPC Client Interface Reference

EPDOC-XX78-en-431A
February 2015

Release 431

Honeywell

Document	Release	Issue	Date
EPDOC-XX78-en-431A	431	0	February 2015

Disclaimer

This document contains Honeywell proprietary information. Information contained herein is to be used solely for the purpose submitted, and no part of this document or its contents shall be reproduced, published, or disclosed to a third party without the express permission of Honeywell International Sàrl.

While this information is presented in good faith and believed to be accurate, Honeywell disclaims the implied warranties of merchantability and fitness for a purpose and makes no express warranties except as may be stated in its written agreement with and for its customer.

In no event is Honeywell liable to anyone for any direct, special, or consequential damages. The information and specifications in this document are subject to change without notice.

Contents

Planning considerations for installing and configuring OPC Client controllers

This reference provides the information you need to set up, configure, and test OPC Client controller communications with the server.

Revision history

Revision	Date	Description
A	February 2015	Initial release of document.

How to use this guide

These are the steps for connecting and configuring an OPC Client controller. Complete each step before commencing the next.

Steps	Go to
Install and set up the OPC server	Documentation supplied by the OPC server manufacturer
Install any OPC server configuration files on the OPC client computer	Documentation supplied by the OPC server manufacturer
Test communications between the OPC client computer and the OPC server computer	Testing OPC Client communications with the server
Use Quick Builder to define channels	• OPC Client channel and controller reference • "Build channels" topic in the *Quick Builder User’s Guide*
Use Quick Builder to define controllers	• OPC Client channel and controller reference • "Build controllers" topic in the *Quick Builder User’s Guide*
Download channel and controller definitions to the server	"Downloading items" topic in the *Quick Builder User’s Guide*
Use Quick Builder to define points	Defining an OPC Client address for a point parameter

Related topics

“OPC Client support” on page 7
“Other documentation for OPC” on page 8
“OPC-specific terms” on page 9
“Description of OPC” on page 10
“OPC Client interfaces and methods” on page 13
“Architectures for OPC Client” on page 14
“OPC Client controllers, channels, and points” on page 15
“Testing OPC Client communications with the server” on page 30

“OPC Client channel and controller reference” on page 17
“Defining an OPC Client address for a point parameter” on page 26

OPC Client support

The OPC Client Interface is an OPC client that supports communications to OPC servers meeting the specification of the OPC Data Access Standard, version 1.0a or 2.0.

The OPC Client Interface supports both local and remote OPC servers.

Other documentation for OPC

You should read the documentation supplied by the OPC server manufacturer before creating any OPC Client controllers.

OPC-specific terms

This section contains OPC-specific terms.

CLSID A class identifier that identifies an object. An object registers its CLSID in the system registration database so the object can be loaded and programmed by other applications.

group A number of items with the same update rate and deadband.

item A single data source of the OPC server.

OPC OPC stands for OLE for Process Control. It is a set of standards that define sets of COM interfaces (based on Microsoft's COM/OLE technology) to be observed by OPC clients and servers. This set of standards was established by the OPC Foundation to foster greater interoperability between automation and control applications, field systems and devices, and business and office applications. When the term OPC is used in this Reference, it refers specifically to the OPC Data Access Standard.

ProgID A programmatic identifier. A registry entry that can be associated with a class identifier (CLSID). The format of a ProgID is <*Vendor*>.<*Component*>.<*Version*>, separated by periods and with no spaces. The ProgID identifies a class, but with less precision. The ProgID is used to identify the OPC server from other COM/DCOM components on the same computer. For more details, see any setup and installation instructions provided by the OPC server manufacturer.

update rate The internal update rate of the items in the OPC server.

Description of OPC

OPC provides data from a data source and communicates the data to any client application in a standard way, thereby eliminating the requirement for an application to have specific knowledge about a particular data source, such as internal structure and communications protocols.

An OPC server and an OPC client can reside either on the same computer (local server) or on different computers (remote server).

How OPC data transfer works

OPC server data is available to the OPC client as *items*. To receive items from the OPC server, the OPC client must gather one or more items into a *group*.

The OPC client requests the OPC server to create a group with a client-specified maximum (at most) update rate and a deadband. The OPC client then requests the OPC server to add items to the group. The update rate and the deadband of a group apply to all items in that group.

Tip

There is no concept of hardware in the OPC Data Access Standard; there are just items. An OPC server may represent a piece of hardware as an item and the item's value may indicate the state of the hardware. However, whether or not such representation is available is server-specific.

Although the OPC client can specify any update rate for a group, the OPC server decides whether the request is honored.

Although the OPC client can specify the deadband for a group, the OPC server decides whether the deadband request is honored.

Callback

Generally, the OPC server sends data to OPC clients through callbacks. After a group has been created, the OPC server creates a cache for the group items. The cache is updated according to the group's update rate. The OPC server sends only updated values to the OPC client for items in the group if there has been significant change since the last cache update (based on the group's deadband).

This method of data update significantly reduces traffic between the OPC client and the OPC server as there is no need for periodic read requests to the OPC server. An OPC client gets data when there is a significant change. The level of change required to trigger an update from the OPC server is defined by the OPC client.

Explicit read request

The OPC client can also send explicit read requests to the OPC server, independent of the server callbacks. The OPC client specifies whether the data should come from the OPC server's internal cache or from the field/ hardware device. This method of scanning is less efficient than callback.

OPC Quality

OPC quality and sub status parameters are returned with the item value. In the Experion OPC Client Interface, use the configuration on the OPC channel's Main properties tab to have 'stale' driven by this quality.

OPC quality values are stored in Experion history archives.

The OPC quality and sub status parameters are available to external OPC clients connected to the Experion OPC DA Server.

OPC Quality flags

OPC Quality flags represent the quality state for an item's data value. The low eight bits of the Quality flag are defined in the form of three bit fields: Quality, Substatus, and Limit status. The eight Quality bits are arranged as follows:

```
QQSSSSLL
```

Quality BitField

Table 1: Quality BitField

QQ	Bit value	Define	Description
0	*00SSSSLL*	Bad	The value is not useful for reasons indicated by the Substatus.
1	*01SSSSLL*	Uncertain	The quality of the value is uncertain for reasons indicated by the Substatus.
2	*10SSSSLL*	N/A	Not used by OPC.
3	*11SSSSLL*	Good	The quality of the value is good.

Substatus BitField

Table 2: Substatus BitField for Bad Quality

SSSS	Bit value	Define	Description
0	*000000LL*	Non-specific	The value is bad but no specific reason is given.
1	*000001LL*	Configuration Error	There is a server-specific problem with the configuration. For example, the item in question has been deleted from the configuration.
2	*000010LL*	Not Connected	The input is required to be logically connected to something but is not. This quality may reflect that no value is available at this time, for reasons such as the data source did not provide the value.
3	*000011LL*	Device Failure	A device failure has been detected.
4	*000100LL*	Sensor Failure	A sensor failure has been detected. In some situations, the Limits bit field can provide additional diagnostic information.
5	*000101LL*	Last Known Value	Communications have failed. However, the last known value is available. Note that the age of the value may be determined from the *TIMESTAMP* in the *OPCITEMSTATE*.
6	*000110LL*	Comm Failure	Communications have failed. The last known value is not available.
7	*000111LL*	Out of Service	The block is off scan or otherwise locked. This quality is also used when the active state of the item or the group containing the item is InActive.
8–15		N/A	Not used by OPC.

Table 3: Substatus BitField for Uncertain Quality

SSSS	Bit value	Define	Description
0	*010000LL*	Non—specific	The value is uncertain but no specific reason is given.

SSSS	Bit value	Define	Description
1	*010001LL*	Last Usable Value	Whatever was writing to this value has stopped doing so. The returned value should be regarded as stale. This substatus is associated with the failure of some external source to write something to the value within an acceptable period. Note that the age of the value may be determined from the *TIMESTAMP* in the *OPCITEMSTATE*. This substatus differs from a Bad Quality value with Substatus 5 (Last Known Value). That substatus is associated specifically with a detectable communications error on a fetched value.
2–3		N/A	Not used by OPC.
4	*010100LL*	Sensor Not Accurate	Either one of the following scenarios has occurred: • The value has pegged at one of the sensor limits, in which case the Limits field should be set to 1 or 2. • By some form of internal diagnostics, the sensor is known to be out of calibration, in which case the Limits field should be set to 0.
5	*010101LL*	Engineering Units Exceeded	The returned value is outside the limits defined for this parameter. Note that in this case the Limits field indicates which limit has been exceeded but does not necessarily imply that the value cannot move farther out of range.
6	*010110LL*	Sub-Normal	The value is derived from multiple sources and has less than the required number of Good sources.
7–15		N/A	Not used by OPC.

Table 4: Substatus BitField for Good Quality

SSSS	Bit value	Define	Description
0	*110000LL*	Non-specific	The value is good. There are no special conditions.
1–5		N/A	Not used by OPC.
6	*110110LL*	Local Override	The value has been overridden. Typically this means the input has been disconnected and a manually entered value has been forced.
7–15		N/A	Not used by OPC.

Limit BitField

Table 5: Limit BitField

LL	Bit value	Define	Description
0	*QQSSSS00*	Not Limited	The value is free to move up or down.
1	*QQSSSS01*	Low Limited	The value is pegged at some lower limit.
2	*QQSSSS10*	High Limited	The value is pegged at some higher limit.
3	*QQSSSS11*	Constant	The value is constant and cannot move.

Related topics

"Main properties for an OPC Client channel" on page 18

OPC Client interfaces and methods

The following table lists the interfaces and methods used by the OPC Client Interface.

Table 6: Interfaces and methods

Interface	Methods
IOPCCommon	SetClientName
IOPCServer	AddGroup GetStatus RemoveGroup
IOPCItemMgt	AddItems RemoveItems ValidateItems
IOPCSyncIO	Read Write
IOPCAsyncIO	Read Refresh
IOPCAsyncIO2	Read (OPC DA 2 only) Refresh2 (OPC DA 2 only)

The OPC Client Interface implements the IAdviseSink, IOPCDataCallback and IOPCShutdown interfaces.

Architectures for OPC Client

For the OPC Client Interface to communicate with an OPC server, the OPC server must be installed on an appropriate computer. (The OPC Client Interface and the OPC server can reside on the same computer or on different computers.)

To install an OPC server, install and set up the OPC server as specified in the documentation supplied by the OPC server manufacturer.

Valid OPC interface configurations

The server OPC interface supports single-channel communications.

Single channel communications

A single-channel configuration.

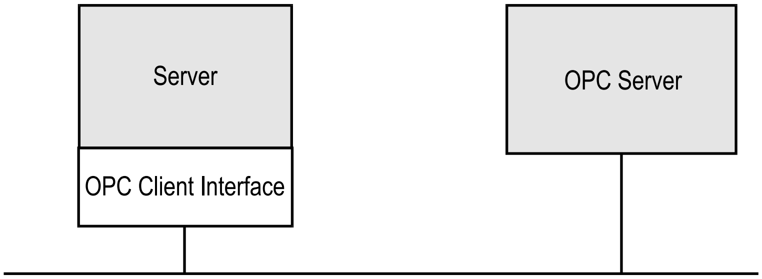

Figure 1: OPC Interface single-channel configuration

Alternate data source

The OPC Interface also supports an alternate data source. If the OPC Interface connection to the primary data source fails, the OPC Interface connects to the alternate data source.

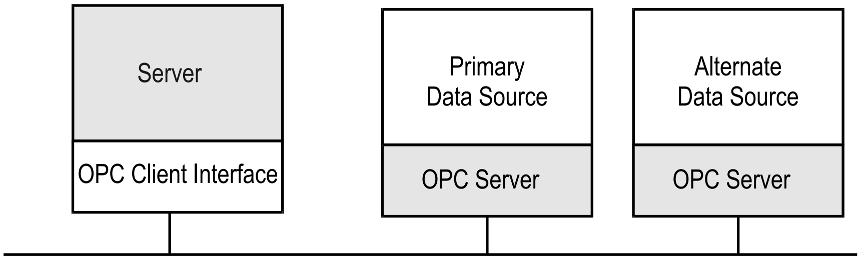

Figure 2: OPC redundant alternate data source configuration

OPC Client controllers, channels, and points

An OPC Client controller is Quick Builder's mechanism for configuring one or more groups with the same deadband. Unlike most other types of controller, an OPC Client controller is an abstract entity, and does not represent a physical device.

Similarly, a point parameter is Quick Builder's mechanism for defining an item on an OPC Client controller. During configuration, you map each item to an appropriate point parameter on an OPC Client controller. Only floating point number data types are supported.

An OPC Client controller manages groups as follows:

- A group is created for each scan period used by the point parameters (items) defined on the controller. (That is, all point parameters that have the same scan period are placed in the same group.) Note that if a scan period of 0 is used, the item is placed in a group with an update rate equal to the slowest scan rate of the server system.
- All points defined on the controller have the same OPC deadband. (Note that the OPC deadband is not the same as the alarm and control deadbands that can be specified for analog points.)
- The maximum number of items that can be configured in a controller is 735.

An OPC Client channel forms the interface between one or more OPC Client controllers and an OPC server. You configure an OPC Client channel in the same way as other channels; however, two properties have special significance as far as OPC is concerned:

- **Diagnostic Scan Period**. This determines how often the OPC client tests the status of the OPC server to ensure that it is OK.
- **Background Scan Period**. This applies to all controllers configured under the channel and determines how often the OPC client explicitly reads all items configured on a controller if that controller is configured with background scanning enabled. This is independent of the callbacks.

OPC Client channel and controller reference

This section describes the configuration and addressing information specific to OPC Client channels and controllers.

In addition to the information contained in this reference, and for help to build channels and controllers, see the section titled "Building controllers or channels" in the *Quick Builder User's Guide*.

Tip

An OPC Client controller is an abstract entity that is used as a container for OPC groups. An OPC Client controller does not represent a real controller or a real piece of hardware.

Related topics

"Main properties for an OPC Client channel" on page 18

"Main properties for an OPC Client controller" on page 21

"Planning considerations for installing and configuring OPC Client controllers" on page 5

Main properties for an OPC Client channel

The Main tab defines the basic properties for an OPC Client channel.

For information about how to create a channel, see the topic titled "Building controllers and channels" in the *Quick Builder User's Guide*.

Tip

If you specify an alternate host name, Quick Builder builds a redundant OPC channel. Refer to the *Configuration Guide* for your server for information about monitoring redundant channels.

Property	Description
Name	The unique name of the channel. A maximum of 10 alphanumeric characters (no spaces or double quotes). Note: In Station displays, underscore characters (_) appear as spaces.
Description	(Optional) A description of the channel. A maximum of 132 alphanumeric characters, including spaces.
Marginal Alarm Limit	The communications alarm marginal limit at which the channel is declared to be marginal. When this limit is reached, a high priority alarm is generated. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one channel, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*. A channel barometer monitors the total number of requests and the number of times the controller did not respond or response was incorrect. The barometer increments by two or more, depending on the error, and decrements for each good call. To calculate an acceptable marginal alarm limit, use the formula: Square root of the number of controllers on the channel × Marginal Alarm Limit defined on those controllers (Normally, you specify the same value for all controllers on a channel). For example, if there are 9 controllers on the channel and their Marginal Alarm Limit is set to 25, the value would be 3 (which is the square root of 9) x 25 = 75.
Fail Alarm Limit	The communications alarm fail limit at which the channel is declared to have failed. When this barometer limit is reached, an urgent alarm is generated. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one channel, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*. Set this to double the value specified for the channel Marginal Alarm Limit.
Connect Timeout	Amount of time, in seconds, the server waits to connect to the OPC server before abandoning the connection. The default is 20 seconds. **Attention** When connecting to redundant third party OPC Servers via Redirection Manager (RDM), it is recommended that the connect timeout be set to 40 seconds.
Read Timeout	Amount of time, in seconds, the server waits for a reply from the OPC server after a synchronous read request. The default is 2 seconds. **Note:** When connecting to legacy third party control networks, it is recommended that the Read Timeout be set to 5 seconds.

Property	Description
Host Name (preferred)	The name of the computer on which the preferred OPC server software resides. **Note:** The OPC Server should not be installed on the Honeywell server. It should be installed on another machine. The exception to this is the MatrikonOPC Server for Allen-Bradley PLCs, which may be installed locally. In this case the name must be *LocalHost*. This is to support server redundancy, so that if the Honeywell servers fails over, it always connects back to the MatrikonOPC server on the local machine. **Note:** DCOM configuration is required when connecting to an OPC server located on another computer. This is required for the MNGR account to connect properly. For information about configuring DCOM, see the *Supplementary Installation Tasks Guide*. **Note:** When you use Quick Builder to configure non-scanned parameters that connect to a remote OPC server, your Windows user account must also reside on the remote OPC server. In addition, the account must be a member of the *Product Administrators* group on the remote OPC server. If your Windows user account does not exist or does not have sufficient privileges on the remote OPC server, you will not be able to select parameters from the list of parameters stored on the server. Username and passwords must match on both machines.
Host Name (alternate)	(Optional) The name of the computer on which the alternate OPC server software resides. If it is on the same computer as the OPC Client Interface, the name must be *LocalHost*. **Note:** If you specify an alternate host name, Quick Builder builds a redundant channel OPC connection. See the *Configuration Guide* for the server for information on monitoring the status of redundant channels.
Diagnostic Scan period	The amount of time, in seconds, between diagnostic scans. The diagnostic rate must be set to one of the valid server scan periods. The default is *60* seconds. This value is used as the rate for sending synchronous requests for checking the OPC server's current status.
Background Scan Period	The background scanning period (in seconds). This must be set to one of the valid server scan periods (defaults to *60*). This value is used as the rate for sending explicit read requests of all items configured under a controller if that controller is also configured with background scanning enabled. **Note:** When connecting to legacy third party control networks, it is recommended that the background scan period be set to *10* minutes.
ProgID	The ProgID for the OPC server that is to be connected.
OPC server host time can drift	It is recommended that this property always be selected. Time drift of OPC server computer clocks (UTC time) is being seen more and more regularly. Note: Moving in or out of daylight savings time is not a time drift as the UTC time does not change.
Last Known Value substatus is stale	If selected, the point parameter is processed as stale if the quality substatus provided by the OPC Server is *Last known value*. If selected, the point parameter is processed as bad if the quality substatus provided by the OPC Server is *Last known value*. See the topic titled "OPC Quality" for more information.
Last Usable Value substatus is stale	If selected, the last usable value of a point parameter is processed as stale. If not selected, the last usable value of a point parameter is processed as bad. See the topic titled "OPC Quality" for more information.
Item Type	The type of channel specified when this item was created.
Last Modified	The date and time the channel properties were modified.
Last Downloaded	The date and time the channel was last downloaded to the server.

Property	Description
Item Number	The unique item number currently assigned to this channel, in the format `CHNcc`, where `cc` is the channel number. You can change the item number if you need to match your current server database configuration. The number must be between `01` and the maximum number of channels allowed for your system. For more information about setting the maximum value, see the topic titled "Adjusting sizing of non-licensed items" in the *Supplementary Installation Tasks Guide*.

Related topics

"OPC Quality" on page 10

Main properties for an OPC Client controller

The **Main** tab defines the basic properties for an OPC Client controller.

For information about how to create a controller, see the topic titled "Building controllers and channels" in the *Quick Builder User's Guide*.

Property	Description
Name	The unique name of the controller. A maximum of *10* alphanumeric characters (no spaces or double quotes). Note: In Station displays, underscore characters (_) appear as spaces.
Description	(Optional) A description of the controller. A maximum of *132* alphanumeric characters, including spaces.
Channel Name	The name of the channel on which the controller communicates with the server. (You must have already defined a channel for it to appear in this list.)
Marginal Alarm Limit	The communications alarm marginal limit at which the controller is declared to be marginal. When this limit is reached, a high priority alarm is generated. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one controller, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*. A controller barometer monitors the total number of requests and the number of times the controller did not respond or response was incorrect. The barometer increments by two or more, depending on the error, and decrements for each good call. The default value is *25*.
Fail Alarm Limit	The communications alarm fail limit at which the controller is declared to have failed. When this barometer limit is reached, an urgent alarm is generated. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one controller, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*. Set this to double the value specified for the controller Marginal Alarm Limit. The default is *50*.
Background Scan	Specify whether background scanning is performed in addition to normal callback operation. Select **Enabled** if you do not want the default (**Disabled**). The background scan period is configured under the channel.

Property	Description
Deadband	The OPC deadband, which applies to all items referenced by point parameters belonging to this controller. Note that although the OPC client can specify the deadband for a group, the OPC server decides whether the deadband request is honored. The deadband Indices are defined as: *0* = 0.000% *1* = 0.001% *2* = 0.002% *3* = 0.005% *4* = 0.010% *5* = 0.020% *6* = 0.050% *7* = 0.100% *8* = 0.200% *9* = 0.500% *10* = 1.000% *11* = 2.000% *12* = 5.000% *13* = 10.000% *14* = 20.000% *15* = 50.000%
Diagnostics item	Item in the OPC Server that represents the status of the physical controller. The controller will be marked as failed when the item has the same value as the **Failure Value** (i.e., Zero or Non Zero). Click the ellipses button to display the OPC Browser, and select the diagnostic item. Maximum 65 characters.
Failure Value	The value that indicates an error condition. **Zero** This is the default value. When diagnostics item has a value of 0, OPC scan task will fail the controller. For example, the *IsConnected* item used in MatrikonOPC Server for Allen-Bradley PLCs is set to 1 when connected, or 0 when disconnected. **Non Zero** When diagnostics item has a value that is not zero, OPC scan task will fail the controller. For example, the *IsFaulty* item provided by some OPC Servers is set to 0 when connection is established or 1 when error occurred.
SOE Enabled	Enables Sequence of Events option for status points managed by an OPC controller. When the point's value changes, OPC Scan Task generates a new event containing the field timestamp (up to ms) and the new value.
Item Type	The type of controller specified when this item was created.
Last Modified	The date and time the controller properties were modified.
Last Downloaded	The date and time the controller was last downloaded to the server.

Property	Description
Item Number	The unique item number currently assigned to this controller, in the format `RTUnnnnn`. You can change the item number if you need to match your current server database configuration. The number must be between `01` and the maximum number of controllers allowed for your system. For more information about setting the maximum value, see the topic titled "Adjusting sizing of non-licensed items" in the *Supplementary Installation Tasks Guide*.

OPC Client points reference

This section describes how to configure points for an OPC Client controller using Quick Builder.

In addition to the information contained in this reference, and for help to build points, see the section titled "Building and configuring points" in the *Quick Builder User's Guide*.

Related topics

"Defining an OPC Client address for a point parameter" on page 26
"Optimizing OPC Client scanning performance" on page 28

Defining an OPC Client address for a point parameter

A point parameter is Quick Builder's mechanism for "mapping" a single item on an OPC Client controller. When configuring a point parameter, note that:

- The point parameter's **Scan Period** is used as the item's update rate. Note that if a scan period of 0 is used, the item is placed in a group with an update rate equal to the slowest scan rate of the server system. By default, the slowest scan rate is one hour.
- Periodic scanning is based on OPC callbacks, not on OPC synchronous read requests.

Related topics

"Planning considerations for installing and configuring OPC Client controllers" on page 5

Address syntax for OPC Client controllers

For source and destination addresses the format for an OPC Client controller address is:

```
ControllerName Address
```

Part	Description
`ControllerName`	The name of the OPC Client controller.
`Address`	The address in the controller where the value is recorded.

If you would like help with the address, you can use the Address Builder. To display the Address Builder, click ... next to **Address**.

Attention

Experion OPC clients do not support Access Path when defining an address of an OPC item. The OPC standard states that access path is optional for a client to support. If the client does not specify the access path then the server should choose the access path to use.

Address syntax

The format for the address is:

```
[RockwellPath]OPCItemName [DataFormat]
```

Part	Description
`RockwellPath`	The Rockwell path is not normally required. However, it is required for OPC servers developed using the Rockwell OPC Toolkit. You need to see the vendor's documentation to determine whether a Rockwell path is required. The Rockwell path generally represents a device driver or topic (if the server was previously developed as a DDE server). **Attention** If you do specify a Rockwell path: • You must include the square brackets. (In this particular case, they do not simply indicate an optional part of the syntax.) • There is no space between the Rockwell path and the `OPCItemName`.

Part	Description
OPCItemName	The address in the controller where the value is recorded. You need to see the vendor's documentation for the syntax. For example, if you are using the system OPC server to access the PV of a point named *sinewave*, the address would be *sinewave.PV*. A maximum of 64 characters. Any printable ASCII character that you can type, including the space character, is supported.
DataFormat	The data format. If you do not specify a data format, the default is *IEEEFP*.

Data format definitions

If you want to use a user-defined data format, you must define the format on the server. See the section titled "About user-defined data formats" in the *Server and Client Configuration Guide* for more information.

Format	Counts
IEEEFP	IEEE single-precision floating point
U3BCD	0–999 BCD
U4BCD	0–9999 BCD
U4095	0–4095
U999	0–999
U9999	0–9999
U100	0–100
U1023	0–1023
U8B	0–255
S16B	-32768–32767
S8B	-128–127
S9999	-9999–9999
U16B	0–65535
U15B	0–32767
U14B	0–16383

Optimizing OPC Client scanning performance

The maximum amount of data that can be acquired from an controller is influenced by the rate of sending scan packets to the controller. An understanding of the OPC scan packets will help you configure points so that optimal data acquisition performance can be achieved by maximizing the amount of data acquired with each scan packet.

The scan packets that have been built can be listed by using the utility **lisscn** (list scan). Listing scan packets helps verify the scanning strategy.

For more information about **lisscn**, see the section titled "Command Reference" in the *Server and Client Configuration Guide*.

OPC Client scan packets

OPC groups are collections of items with the same callback period and deadband. In order to reduce the number of OPC groups (and hence OPC scan packets) you should:

- Assign all points with the same deadband to the same controller.
- Reduce the number of different scan periods used for points on a controller.

Troubleshooting OPC Client issues

This section describes troubleshooting tasks for OPC Client that you can perform either on the server or from any Station.

Related topics

"Testing OPC Client communications with the server" on page 30

"Troubleshooting OPC Client communication errors" on page 31

Testing OPC Client communications with the server

You use the OPC client test utility, **opctst**, to test the:

- Connection to an OPC server
- Creation of OPC groups on a server
- Addition of OPC items into groups
- Reading and writing of values to OPC items on the OPC server computer

Prerequisites

Before testing:

- Complete any OPC server setup and configuration as required. See the setup and installation instructions supplied by the OPC server manufacturer.
- Install any OPC server configuration files on the OPC client computer as required. See the setup and installation instructions supplied by the OPC server manufacturer.
- Ensure you are logged on using the *mngr* account as this is the account that the OPC Client Interface uses to communicate with the OPC server.

To run the opctst utility

1. Open a Command Prompt window.
2. Type **opctst** and then press Enter.
3. When the Windows application starts, select the OPC menu option and perform each of the menu operations in turn (that is, Initialize COM, Connect to OPC server, and so on).

Related topics

“Planning considerations for installing and configuring OPC Client controllers” on page 5

Troubleshooting OPC Client communication errors

If you experience difficulty getting the OPC Client Interface to communicate with an OPC server, refer to the Experion PKS Server Log file (accessible via an icon in the Diagnostic Tools folder). The log file gives an indication as to the cause of any OPC communications problems.

Notices

Trademarks

Experion®, PlantScape®, SafeBrowse®, TotalPlant®, and TDC 3000® are registered trademarks of Honeywell International, Inc.

OneWireless™ is a trademark of Honeywell International, Inc.

Other trademarks

Microsoft and SQL Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Trademarks that appear in this document are used only to the benefit of the trademark owner, with no intention of trademark infringement.

Third-party licenses

This product may contain or be derived from materials, including software, of third parties. The third party materials may be subject to licenses, notices, restrictions and obligations imposed by the licensor. The licenses, notices, restrictions and obligations, if any, may be found in the materials accompanying the product, in the documents or files accompanying such third party materials, in a file named third_party_licenses on the media containing the product, or at http://www.honeywell.com/ps/thirdpartylicenses.

Documentation feedback

You can find the most up-to-date documents on the Honeywell Process Solutions support website at:

http://www.honeywellprocess.com/support

If you have comments about Honeywell Process Solutions documentation, send your feedback to:

hpsdocs@honeywell.com

Use this email address to provide feedback, or to report errors and omissions in the documentation. For immediate help with a technical problem, contact your local Honeywell Process Solutions Customer Contact Center (CCC) or Honeywell Technical Assistance Center (TAC) listed in the "Support and other contacts" section of this document.

How to report a security vulnerability

For the purpose of submission, a security vulnerability is defined as a software defect or weakness that can be exploited to reduce the operational or security capabilities of the software.

Honeywell investigates all reports of security vulnerabilities affecting Honeywell products and services.

To report a potential security vulnerability against any Honeywell product, please follow the instructions at:

https://honeywell.com/pages/vulnerabilityreporting.aspx

Submit the requested information to Honeywell using one of the following methods:

- Send an email to security@honeywell.com.

  or

- Contact your local Honeywell Process Solutions Customer Contact Center (CCC) or Honeywell Technical Assistance Center (TAC) listed in the “Support and other contacts” section of this document.

Support

For support, contact your local Honeywell Process Solutions Customer Contact Center (CCC). To find your local CCC visit the website, https://www.honeywellprocess.com/en-US/contact-us/customer-support-contacts/Pages/default.aspx.

Training classes

Honeywell holds technical training classes on Experion PKS. These classes are taught by experts in the field of process control systems. For more information about these classes, contact your Honeywell representative, or see http://www.automationcollege.com.

Index