G2-OPC Client Bridge

User's Guide

Version 2020

G2-OPC Client Bridge User's Guide, Version 2020

May 2020

The information in this publication is subject to change without notice and does not represent a commitment by Gensym Corporation.

Although this software has been extensively tested, Gensym cannot guarantee error-free performance in all applications. Accordingly, use of the software is at the customer's sole risk.

Ignite Technologies, Inc.
401 Congress Ave., Suite 2650
Austin, TX 78701 USA
Telephone: +1-800-248-0027
Email: success@ignitetech.com

Part Number: DOC079-830

Contents

Preface

Describes this document and the conventions that it uses.

About this Guide

This guide:

- Explains how to install the G2-OPC Client Bridge, and how to establish connections to multiple OPC Servers simultaneously and communicate with them through the G2-OPC Client Bridge process.
- Describes the system requirements for this release.
- Gives information on bridge features.

Audience

To use the G2-OPC Client Bridge, you must have at least a limited knowledge of G2 and OPC (OLE for Process Control) technology.

Conventions

This guide uses the following typographic conventions and conventions for defining system procedures.

Typographic

Convention Examples	Description
g2-window, g2-window-1, ws-top-level, sys-mod	User-defined and system-defined G2 class names, instance names, workspace names, and module names
history-keeping-spec, temperature	User-defined and system-defined G2 attribute names
true, 1.234, ok, “Burlington, MA”	G2 attribute values and values specified or viewed through dialogs
Main Menu > Start KB Workspace > New Object create subworkspace Start Procedure	G2 menu choices and button labels
conclude that the x of y ...	Text of G2 procedures, methods, functions, formulas, and expressions
new-argument	User-specified values in syntax descriptions
text-string	Return values of G2 procedures and methods in syntax descriptions
File Name, OK, Apply, Cancel, General, Edit Scroll Area	GUIDE and native dialog fields, button labels, tabs, and titles
File > Save Properties	GMS and native menu choices
workspace	Glossary terms

Convention Examples	Description
`c:\Program Files\Gensym\`	Windows pathnames
`/usr/gensym/g2/kbs`	UNIX pathnames
`spreadsh.kb`	File names
`g2 -kb top.kb`	Operating system commands
`public void main()` `gsi_start`	Java, C and all other external code

Note Syntax conventions are fully described in the *G2 Reference Manual*.

Procedure Signatures

A procedure signature is a complete syntactic summary of a procedure or method. A procedure signature shows values supplied by the user in *italics*, and the value (if any) returned by the procedure *underlined*. Each value is followed by its type:

g2-clone-and-transfer-objects
(*list*: class item-list, *to-workspace*: class kb-workspace,
delta-x: integer, *delta-y*: integer)
-> *transferred-items*: g2-list

Related Documentation

G2 Core Technology

- *G2 Bundle Release Notes*
- *Getting Started with G2 Tutorials*
- *G2 Reference Manual*
- *G2 Language Reference Card*
- *G2 Developer's Guide*
- *G2 System Procedures Reference Manual*

- *G2 System Procedures Reference Card*
- *G2 Class Reference Manual*
- *Telewindows User's Guide*
- *G2 Gateway Bridge Developer's Guide*

G2 Utilities

- *G2 ProTools User's Guide*
- *G2 Foundation Resources User's Guide*
- *G2 Menu System User's Guide*
- *G2 XL Spreadsheet User's Guide*
- *G2 Dynamic Displays User's Guide*
- *G2 Developer's Interface User's Guide*
- *G2 OnLine Documentation Developer's Guide*
- *G2 OnLine Documentation User's Guide*
- *G2 GUIDE User's Guide*
- *G2 GUIDE/UIL Procedures Reference Manual*

G2 Developers' Utilities

- *Business Process Management System Users' Guide*
- *Business Rules Management System User's Guide*
- *G2 Reporting Engine User's Guide*
- *G2 Web User's Guide*
- *G2 Event and Data Processing User's Guide*
- *G2 Run-Time Library User's Guide*
- *G2 Event Manager User's Guide*
- *G2 Dialog Utility User's Guide*
- *G2 Data Source Manager User's Guide*
- *G2 Data Point Manager User's Guide*
- *G2 Engineering Unit Conversion User's Guide*
- *G2 Error Handling Foundation User's Guide*
- *G2 Relation Browser User's Guide*

Bridges and External Systems

- *G2 ActiveXLink User's Guide*
- *G2 CORBALink User's Guide*
- *G2 Database Bridge User's Guide*
- *G2-ODBC Bridge Release Notes*
- *G2-Oracle Bridge Release Notes*
- *G2-Sybase Bridge Release Notes*
- *G2 JMail Bridge User's Guide*
- *G2 Java Socket Manager User's Guide*
- *G2 JMSLink User's Guide*
- *G2 OPCLink User's Guide*
- *G2 PI Bridge User's Guide*
- *G2-SNMP Bridge User's Guide*
- *G2 CORBALink User's Guide*
- *G2 WebLink User's Guide*

G2 JavaLink

- *G2 JavaLink User's Guide*
- *G2 DownloadInterfaces User's Guide*
- *G2 Bean Builder User's Guide*

G2 Diagnostic Assistant

- *GDA User's Guide*
- *GDA Reference Manual*
- *GDA API Reference*

Customer Support Services

You can obtain help with this or any Gensym product from Gensym Customer Support. Help is available online, by telephone and by email.

To obtain customer support online:

➔ Access Ignite Support Portal at `https://support.ignitetech.com`.

You will be asked to log in to an existing account or create a new account if necessary. Ignite Support Portal allows you to:

- Register your question with Customer Support by creating an Issue.
- Query, link to, and review existing issues.
- Share issues with other users in your group.
- Query for Bugs, Suggestions, and Resolutions.

To obtain customer support by telephone or email:

➔ Use the following numbers and addresses:

United States Toll-Free	+1-855-453-8174
United States Toll	+1-512-861-2859
Email	support@ignitetech.com

1

Introduction

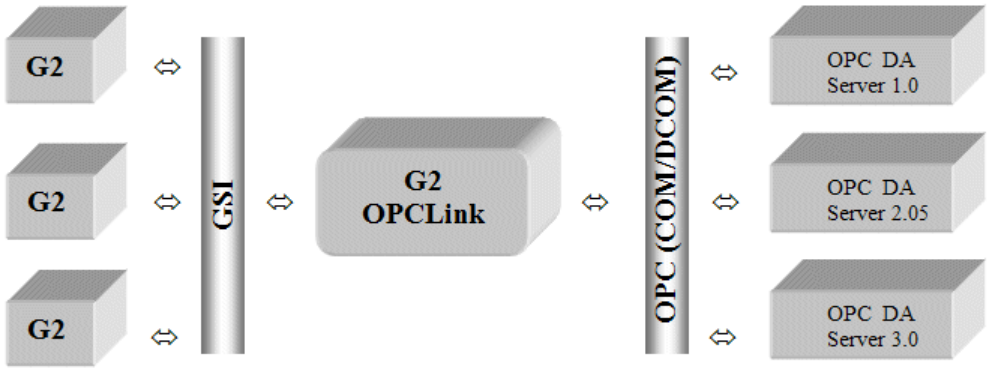
Describes the G2-OPC Client Bridge.

Introduction

The G2-OPC Client Bridge allows G2 applications to access the data points monitored by any Data Access OPC Server. In installations already using several OPC Data Access servers, the bridge allows access to all OPC Server data points from within G2.

The G2-OPC Client Bridge presents OPC items as native G2 objects. Each OPC item is represented by a gsi-opc-item object, which allows G2 to access and reference the data in the same way as other G2 variables. In addition, the G2-OPC Client Bridge provides remote procedure calls (RPCs) for managing and browsing an OPC Server. This data can be used for trend analysis and sophisticated inferencing by the G2 inference engine.

System Architecture



Release Compatibility

The G2-OPC Client Bridge has specific compatibility requirements with G2 and G2 Gateway.

G2

The G2-OPC Client Bridge knowledge base supplied with the bridge, `launch.kb`, is compatible with G2 Version 6.0 or later on Windows platforms only.

G2 Gateway

It is not necessary to have a separately installed G2 Gateway to run this version of the bridge.

Note To achieve a successful installation, you must use only software components provided with this release.

Using Previous Knowledge Bases

A knowledge base built to use previous versions of the G2-OPC Client Bridge will not be fully compatible with this version. You must replace the G2-OPC Client Bridge executable, `G2opc.exe`, and knowledge base, `launch.kb`, with files provided with this release.

2

Installing and Running the G2-OPC Client Bridge

Describes how to install and run the G2-OPC Client Bridge.

Introduction

To retrieve data from OPC Server(s) in real time, you can use the G2-OPC Client Bridge in local and distributed configurations, as follows:

- In local configurations, the G2-OPC Client Bridge and OPC Server(s) all run on a single machine, in which case, the installation process does not need any specific settings.
- In distributed configurations, the G2-OPC Client Bridge and OPC Server(s) run on two or more machines cooperatively, where the bridge initially resides on a remote machine (the client computer) on the network and uses the DCOM mechanism to access server(s) directly. To enable distributed configurations, you must configure various settings on both the remote server and the local client computer.

This chapter describes how to configure DCOM Config Utility settings for computers on which the G2-OPC Client Bridge and OPC server(s) are running.

System Requirements

The G2-OPC Client Bridge requires:

- G2 Version 2015
- OPC Servers compliant with OPC Data Access Server Version 1.0 through 3.0 and all intermediate versions.
- Windows XP, Windows 2003, Windows 2000, Windows 7, Windows 8.1, Windows 10, Windows Server 2008 / 2008R2, or Windows Server 2012 R2
- A minimum of 128 MB.

Installing the G2-OPC Client Bridge

To install the G2-OPC Client Bridge, you must configure the client and server for DCOM on Windows..

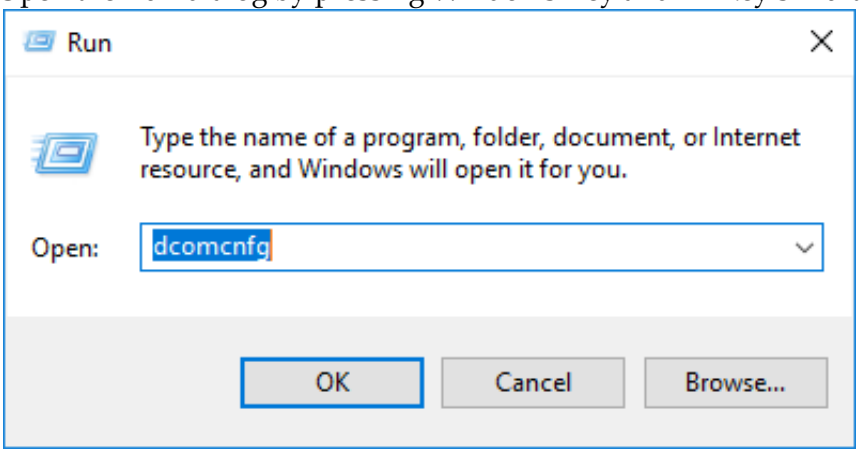
Configuring the Client for DCOM

To configure the client for DCOM, first, you set up the client, then you register the OPC server or servers.

Setting up the Client

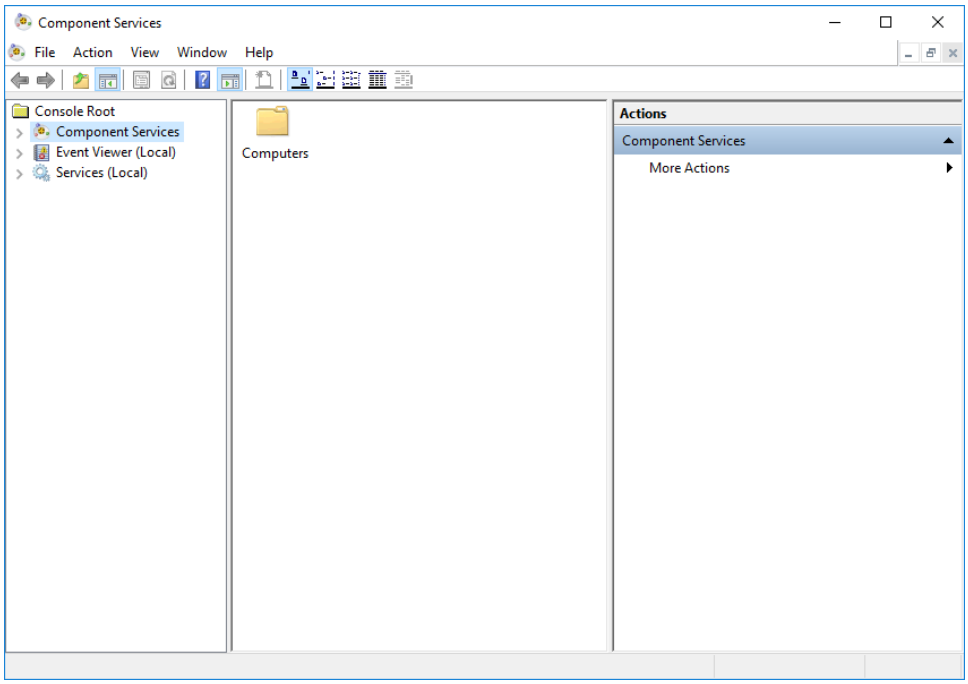
To set up client:

1 Open the Run dialog by pressing Windows Key and 'R' key simultaneously



2 Run the Component Services management tool by typing dcomcnfg, then clicking OK to run .

The Component Services management tool looks like this:



3 Open the computer COM properties by expanding to the Component Services->Computes, right clicking on My Computer then selecting 'Properties'

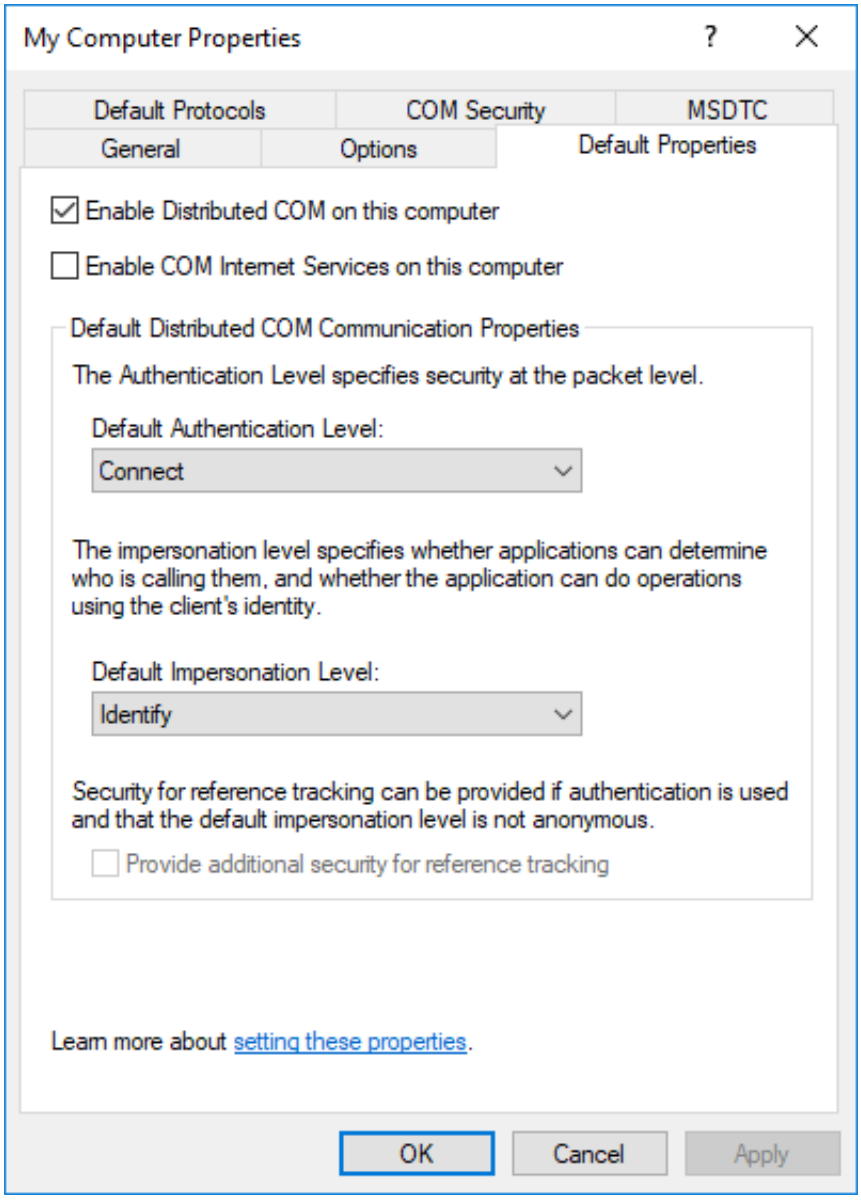
4 Click the Default Properties tab and configure the dialog, as follows:

a Ensure that the Enable Distributed COM on this computer is enabled.

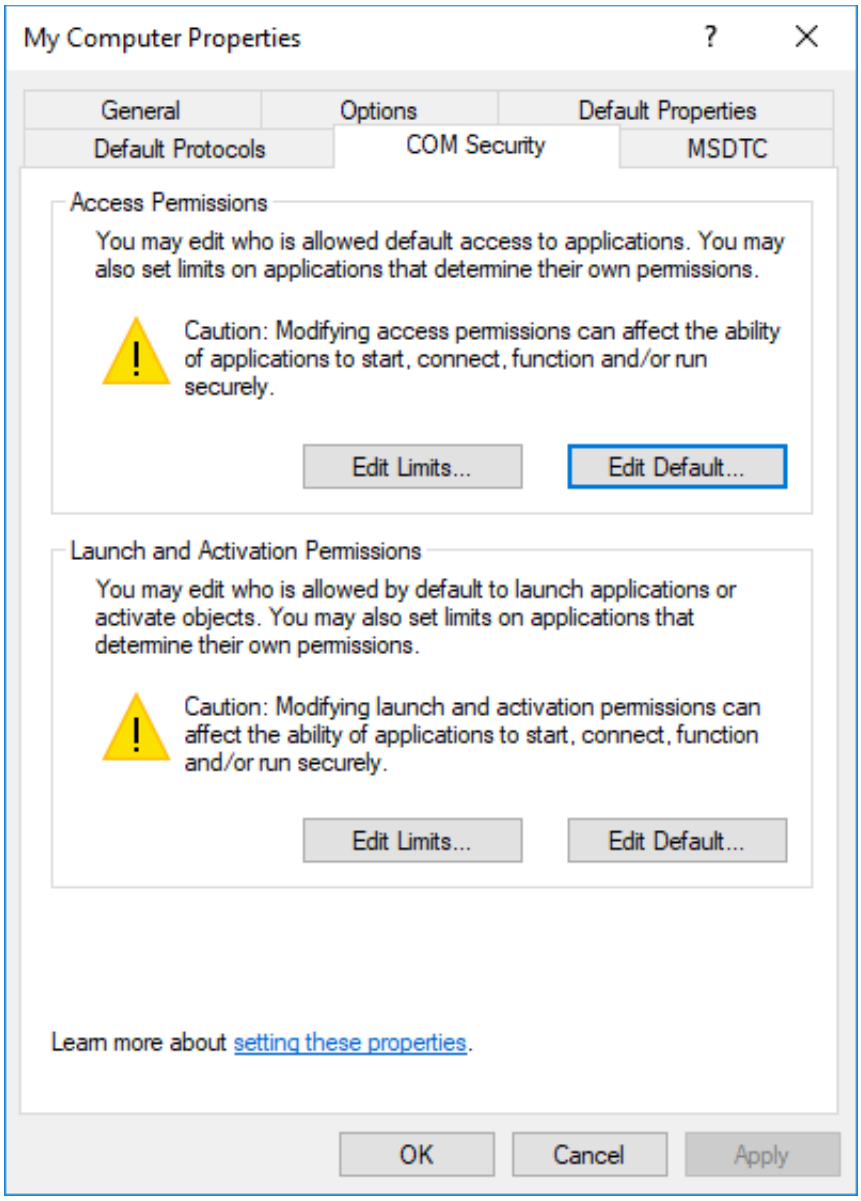
b Configure the Default Authentication Level to be Connect.

c Configure the Default Impersonation Level to be Identity.

The Default Properties tab should look like this:



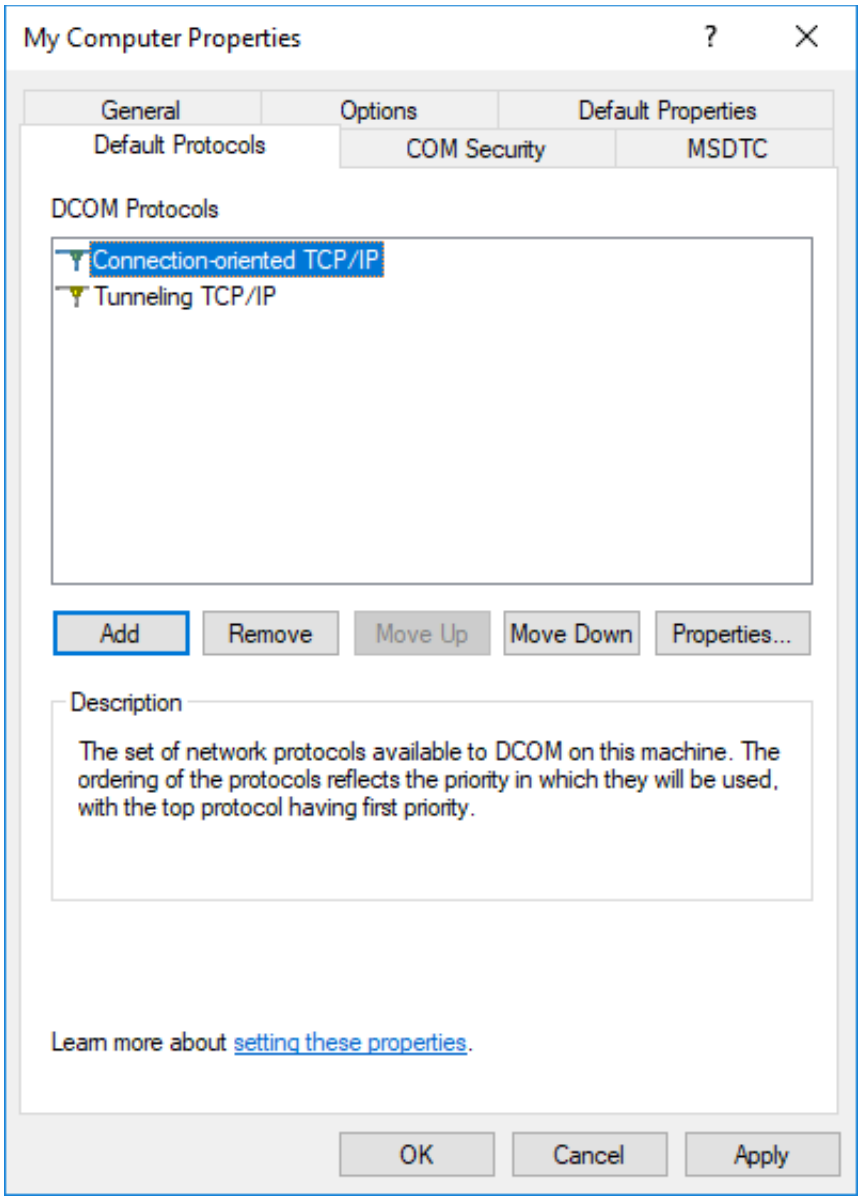
5 Click the COM Security tab:



This is where you configure who has access to the G2-OPC Client Bridge from remote OPC servers. You only need to be concerned with the Default Access Permissions button on this tab.

6 Under Default Access Permissions, click the Edit Default button and configure the users of remote OPC Servers whom you want to be able to make callbacks to this machine when the G2-OPC Client Bridge performs subscription-based read operations.

7 Click the Default Protocols tab:



This is where you set which of the installed network protocols on the client computer to use for DCOM.

8 Configure the DCOM Protocol to use Connection-oriented TCP/IP.

Registering OPC Server(s)

You must now register the OPC Server(s) that the G2-OPC Client Bridge will connect to and specify their locations on the named remote servers. You can use one of two techniques, depending on the client environment.

To register the OPC Server(s), using a customized registry file:

➔ Prepare and apply a customized `reg` file on the client computer, as described in the Microsoft registry documentation.

To register the OPC Server(s) automatically:

1 Install the OPC Server on the client computer, to automatically register the server in the registry.

2 Remove the entries for InProc and OutOfProc servers, then add an entry for RemoteServerName.

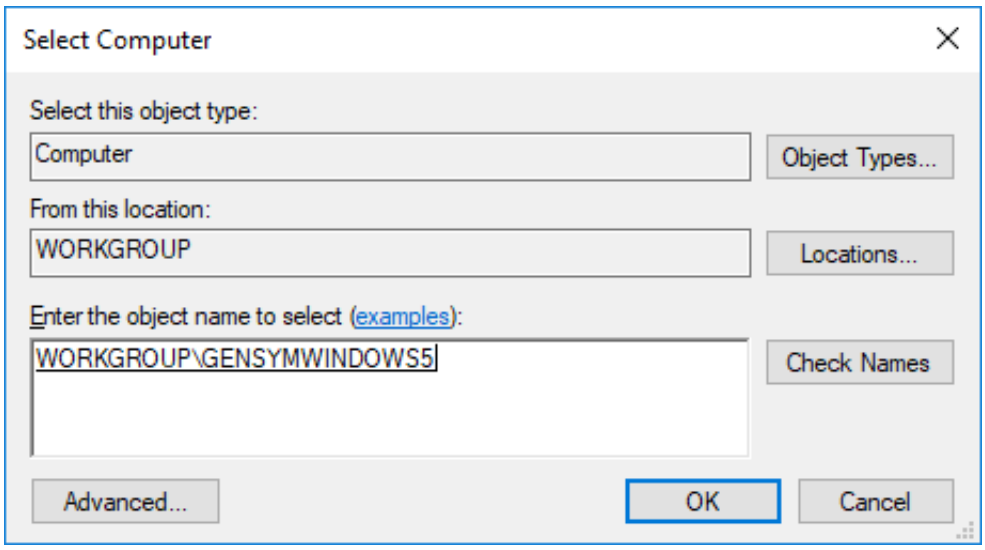
The DCOM configuration utility uses this technique, but you must still configure the client computer manually.

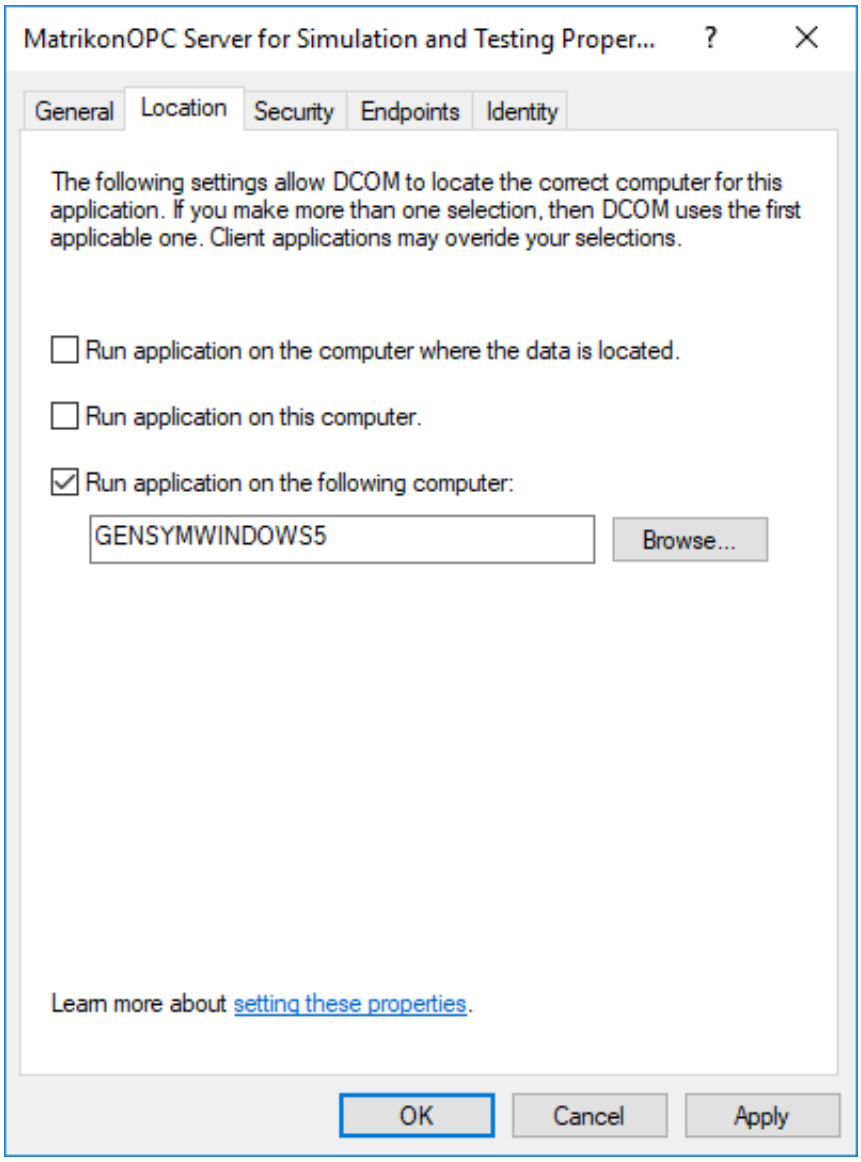
Note In either case, the desired registry information for the OPC Server must not include the LocalServer32 or InProcServer32 keys, and must include the AppID key and the associated RemoteServerName.

If the server application name does not show up in the DCOM configuration control panel, make sure that there is an AppID key associated with the object server's CLSID in the Registry, as follows:

```
[HKEY_ROOT_CLASSES\CLSID\{clsid}]
"AppID"="{clsid}"
```

For example, when installing the OPC Server on the client computer, to access remotely "MatrikonOPC Server", enable the Run application on the following computer option and then either browse for the computer or enter the computer name




MatrikonOPC Server for Simulation and Testing Proper...
General
Location
Security
Endpoints
Identity
The following settings allow DCOM to locate the correct computer for this application. If you make more than one selection, then DCOM uses the first applicable one. Client applications may overide your selections.
Run application on the computer where the data is located.
Run application on this computer.
Run application on the following computer:
GENSYMWINDOWS5
Browse...
Learn more about setting these properties.
OK
Cancel
Apply

Configuring the Server for DCOM

To configure the server for DCOM, first, you set up the server, then you configure DCOM settings for the server.

Setting up the Server

To set up the server:

1. Launch the Component Services management tool on the computer your target OPC Server is running.
2. Open My Computer Properties
3. Configure the Default Properties tab as you did on client side.
4. Click the Default Security tab.

   This is where you tell the operating system who you will allow to access OPC Servers on this machine (Default Access Permissions), who you will allow to launch OPC Servers on this machine (Default Launch Permissions), and who you will allow to configure OPC Servers on this machine (Default Configuration Permissions).
5. Click the Default Access Permissions tab.

   On this tab, when you click Add you will be presented with a dialog that lets you browse the local machine and domain (if applicable and logged into a domain) for users and groups to which to grant permissions.
6. Click the Default Security tab and click Edit Defaults under Default Launch Permissions.

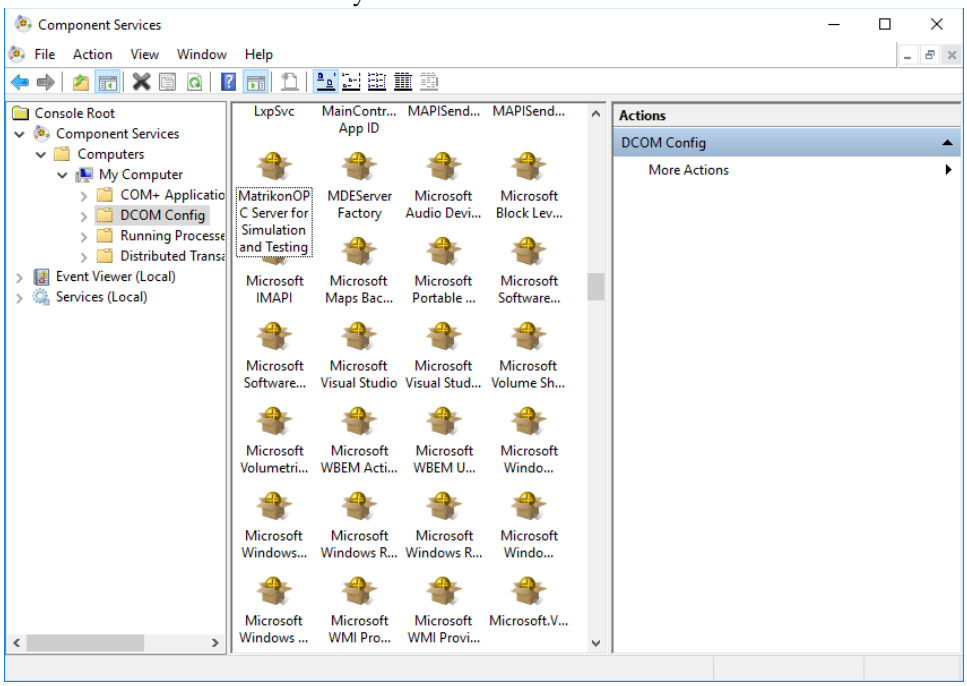
   On this tab, you define who can actually start your OPC Server on this computer. You add users/groups the same way you did when configuring Access Permissions.
7. On the Default Security tab, click Edit Defaults under Default Configuration Permissions.

   If you are setting up DCOM for the first time, we recommend that you do not change these settings.
8. Configure the Default Protocols tab as you did on the client side.

Configuring DCOM Settings for Your OPC Server

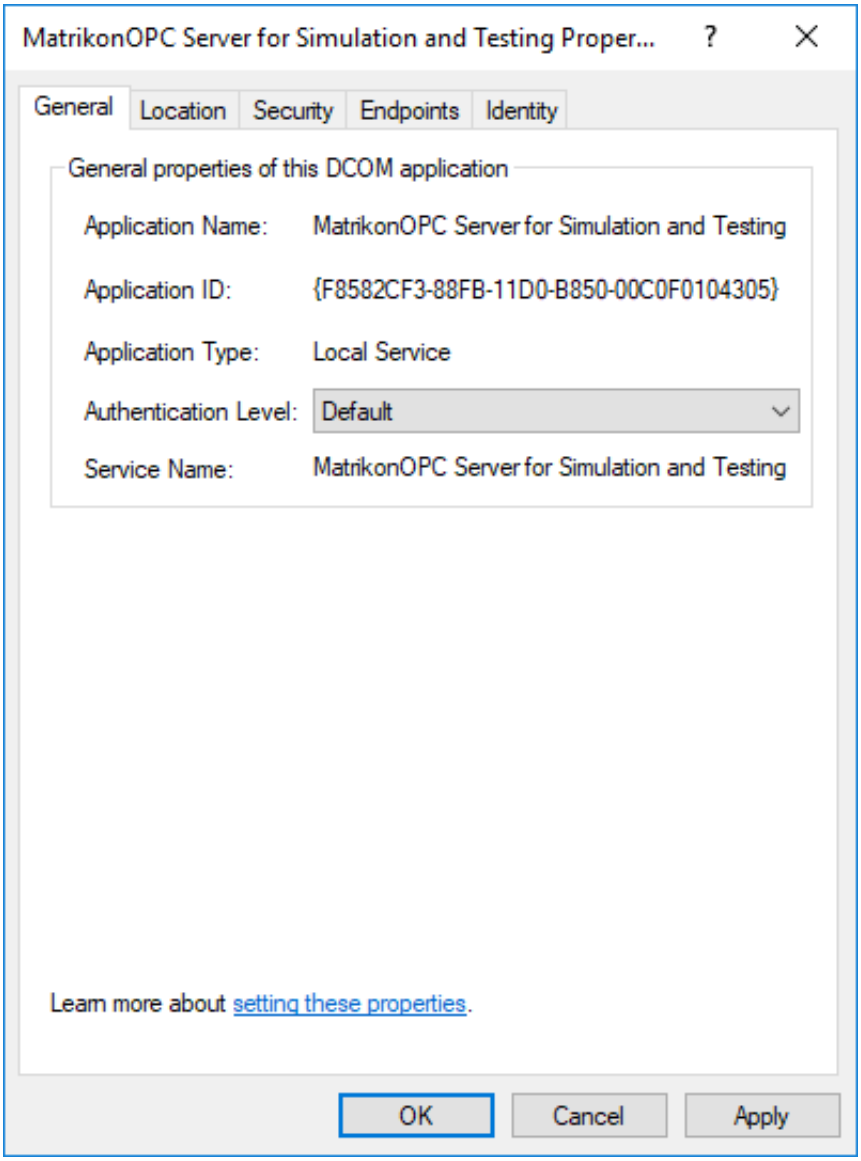
To configure DCOM settings for your OPC Server:

1 Open the DCOM Config subfolder under My Computer and browse until you find the OPC Server of your choice.

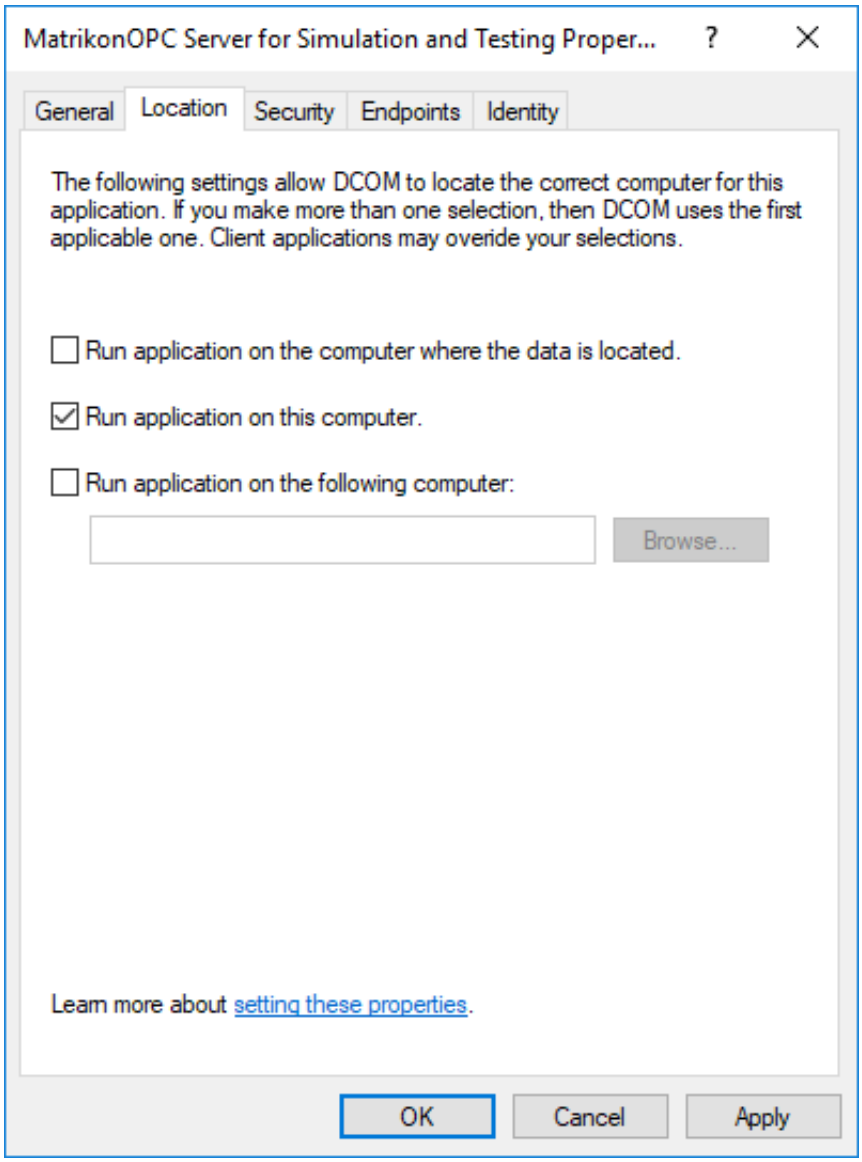


2 Right-click the server and choose Properties to configure server-specific settings, as follows:

a On the General tab, we recommend that you leave the Authentication Level to be Default:



b On the Location Tab, ensure that Run application on this computer is the only check box that is checked:

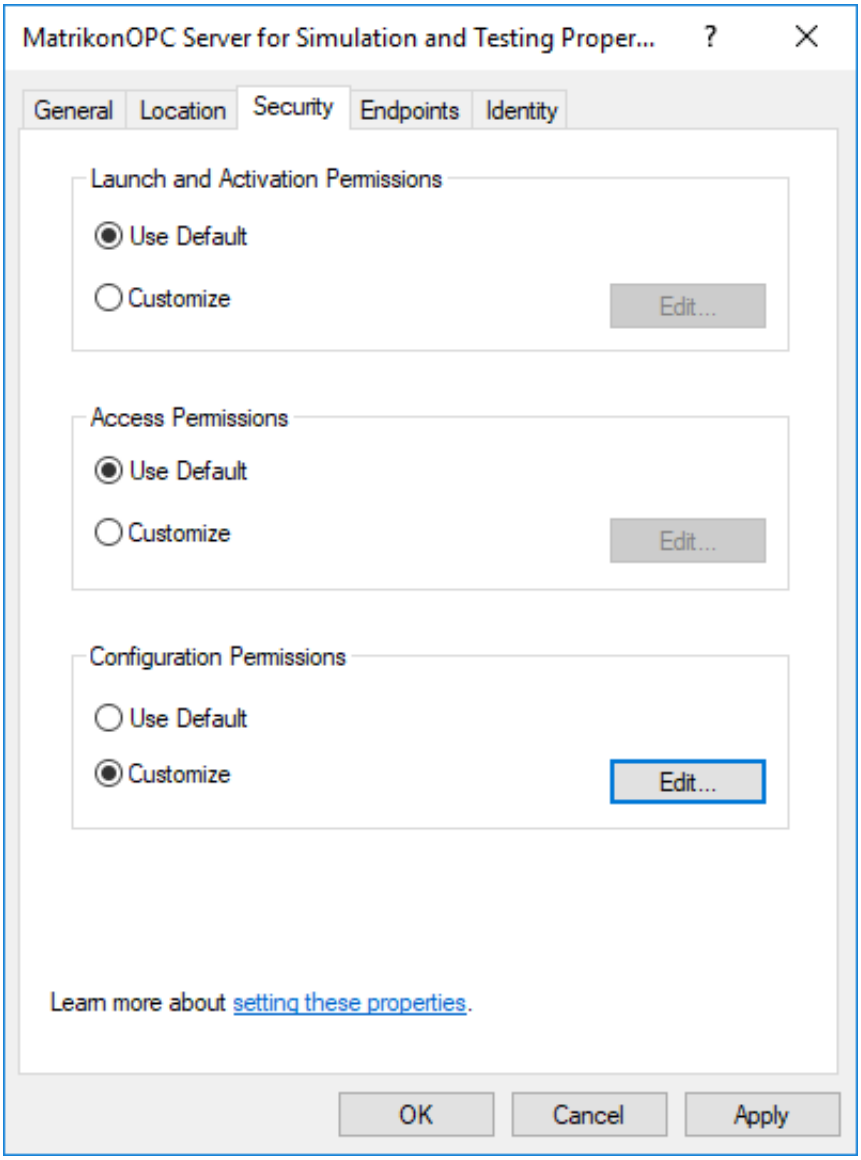


c On the Security Tab, we recommend that you:

- Enable Use default access permissions, which means users/groups shown under Default Security tab in the DCOM configuration utility will have access to connect to this OPC Server.
- Enable Use default launch permissions.
- The same rules apply about using custom launch permissions here as they do for custom access permissions. If you choose to use the custom

permissions to override the defaults, specify which users/groups to which you want to grant permission.

The dialog looks like this:



d On Identity Tab, you specify under what user account you want the OPC Server to run.

This is probably one of the most important settings for the OPC Server, depending on how you will be using your system.

You do not need to configure anything on the Endpoints tab.

Running and Shutting Down the G2-OPC Client Bridge

To run the G2-OPC Client Bridge:

➔ Select "G2 OPC Client Bridge" in the Windows start menu.

To shut down the bridge:

➔ Right-click on the bridge's icon in the notification tray and choose Shutdown.

During shutdown, the bridge process closes all connections to OPC Servers and releases any allocated resources.

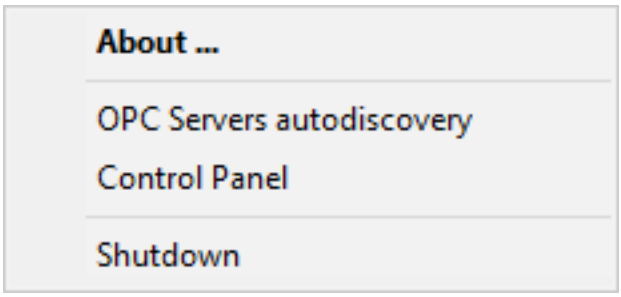
Configuring the G2-OPC Client Bridge at Runtime

The G2-OPC Client Bridge allows you to configure a number of parameters related to logging, filtering, and server status. You can configure these parameters at runtime through a dialog. You can also browse for existing OPC servers at runtime.

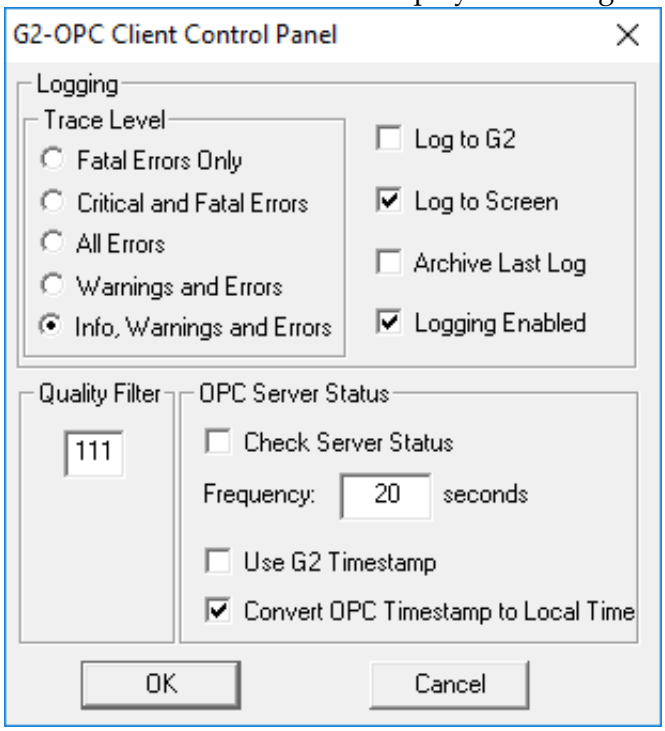
You can also configure these parameters at startup by editing the configuration file, as described in Appendix , Configuration File.

To configure the G2-OPC Client Bridge at runtime:

1 Right click the tray icon for the G2-OPC Client Bridge to display its menu:



2 Choose Control Panel to display this dialog:

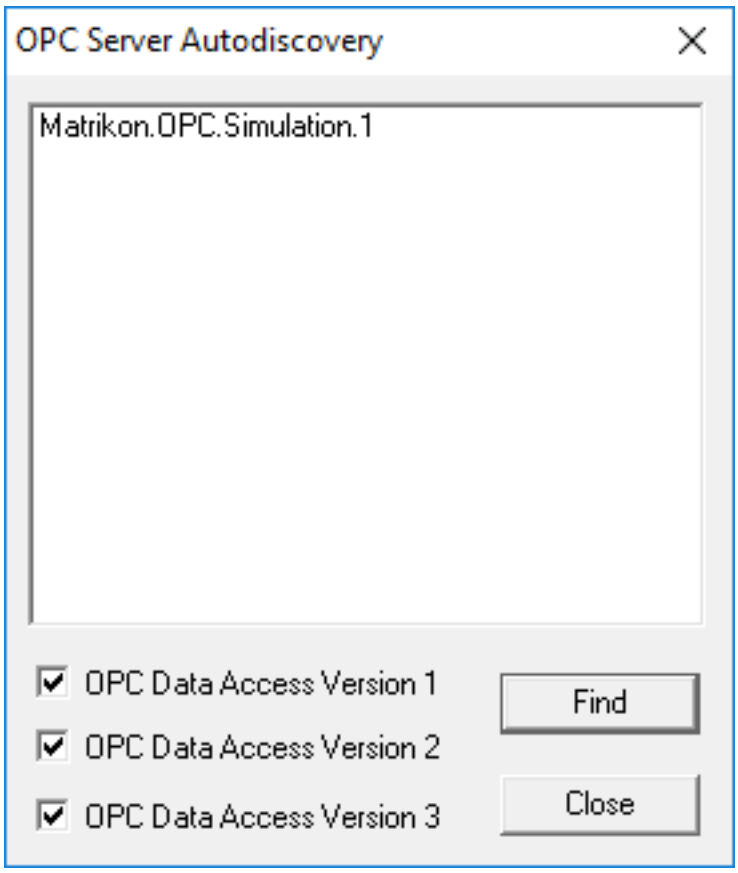


Browsing OPC Servers at Runtime

The auto-discovery dialog of the G2 OPC Client bridges generates a list of OPC DA servers that have been recorded in the registry of your computer.

To browse for an existing OPC server at runtime:

1. Right click the tray icon to display its menu.
2. Choose OPC Servers autodiscovery to display the autodiscovery dialog:



3. Check the versions of the servers in which you are interested.
4. Click the Find button.
5. Click the Close button to close the dialog.

3

Configuring the G2-OPC Client Bridge

Describes how to configure G2-OPC Client Bridge and establish a connection.

Introduction

G2 connects to the G2-OPC Client Bridge through a gsi-opc-interface object, which references an instance of the gsi-opc-interface class contained in the `launch.kb`. Attributes of the instance of gsi-opc-interface specify parameters that make the connection between G2 and the G2-OPC Client Bridge.

This chapter describes the attributes of a gsi-opc-interface object, their definitions, and strategies for optimizing their settings.

Attributes of gsi-opc-interface

The interface object collects all the necessary data to initiate a connection. It contains information required by G2, including the network and the remote system.

PI-INTERFACE, a gsi-opc-interface	
Notes	OK
Item configuration	none
Names	PI-INTERFACE
Identifying attributes	item-id, access-path
Interface warning message level	default to warning message level
Disable interleaving of large messages	no
Interface timeout period	use default
Interface initialization timeout period	unlimited
GSI connection configuration	tcp-ip host "localhost" port-number 22041
External system has a scheduler	yes
Poll external system for data	no
Grouping specification	no grouping
Remote process initialization string	"IntegrationObjects.OPC.PI Group1 1000 0.5 A"
GSI application name	default
GSI interface status	2
Interval to poll external system	use default

Attribute	Description
names	The name of the interface object, which must be unique. Since this object represents the connection to a particular the G2-OPC Client Bridge bridge process, you can choose a name that associates it with the OPC Server.
Allowable values:	Any symbol
Default value:	none
identifying-attributes	Uniquely identifies the variable objects to the OPC Server.
Allowable values:	A list of symbols
Default value:	itemid, accesspath
Notes:	The value of identifying-attributes should always be: itemid, accesspath which are attributes of gsi-opc-item objects.
interface-warning-message-level	Sets the severity level for error and warning messages that G2 provides for the interface object.

Attribute	Description
Allowable values:	Allowable values: 0: No warning or error messages. 1: Serious error messages only. 2: All error messages. 3: All error and warning messages. Level 0 is the lowest severity level and provides the least error information. Increasing the warning message level causes G2 to provide more information about errors and failures that are otherwise only detectable through the value of the gsi-interface-status attribute. Messages are posted to the Operator Logbook by default. For example, when the warning message level is at 0 or 1, a failure to connect to a bridge causes the gsi-interface-status to change to -2 (Error), but no information is made available about why the failure occurred.
Default value:	default to warning message level: The interface-warning-message-level takes on the value of the warning-message-level attribute in the Debugging Parameters system table.
disable-interleaving-of-large-messages	Controls whether G2 Gateway interleaves, or changes the transmission order of message packets.
Allowable values:	yes, no
Default value:	no, which means: • G2 transmits messages in packets. A large message occupies several packets. A small message occupies a single packet. • When more than one message requires transmission across an interface, G2 interleaves the packets that constitute the messages.

Attribute	Description
interface-timeout-period	The length of time G2 will wait for a response from the bridge before logging an error. A reasonable setting depends on how busy your network is and other factors. A good starting value is between 10 and 20 seconds. The minimum setting is 1 second.
Allowable values:	*integer* seconds
Default value:	use default, which uses a default of 1 second
Notes:	See interface-initialization-timeout-period.
interface-initialization-timeout-period	Specifies how long G2 waits to initialize a connection using Gensym (ICP) protocols. The following timeout intervals apply to GSI interfaces: • Establish a connection. • Initialize the connection. • Wait for a response. This attribute applies to the second interval. The interface-timeout-period attribute specifies the timeout period for the first and third intervals.
Allowable values:	An integer specifying some number of seconds. unlimited: The initialization interval never times out. use default: The interface-initialization-timeout-period is the same as the interface-timeout-period.
Default value:	unlimited

Attribute	Description
gsi-connection-configuration	A G2 expression that describes the network connection between G2 and the G2-OPC Client Bridge process. The expression specifies the type of network and the network address of the bridge process.
Allowable values:	tcp-ip host "*host*" port-number *port-number* where: "*host*" is the name of the machine that runs the bridge process. Note that the host name is enclosed in double quotation and may be the IP address of the machine. *port-number* is the TCP/IP port number of the bridge. It is set either from the command line that started the bridge process or in the code for the function getgsiport(). The default port number is 22041. You can change it to any number from 3001 to 29.999 that you are not using for another process on that machine.
Default value:	tcp-ip host "localhost" port-number 22041
external-system-has-a-scheduler	Determines whether scheduling for getting new data values is the responsibility of the OPC Server or G2.
Allowable values:	yes: G2 assumes that the G2 Gateway user code handles the return of data to G2, without explicit requests from G2. no: G2 Gateway continuously reads a queue of requests for data from G2.
Default value:	yes
Notes:	The value of this attribute depends on the communication mode between the bridge and the OPC Server. If they communicate in asynchronous mode, then the external-system-has-a-scheduler attribute should be set to yes; otherwise, it should be set to no.

Attribute	Description
poll-external-system-for-data	Determines whether the G2 Gateway bridge receives unsolicited data from the external system by executing the callback function gsi_g2_poll() every cycle.
Allowable values:	yes: The G2 Gateway bridge calls gsi_g2_poll() every cycle. no: The G2 Gateway bridge does not call gsi_g2_poll().
Default value:	no
Notes:	This attribute should always be set to no.
grouping-specification	(Optional) Enables you to group requests for data service using one or more of the identifying attributes of a variable.
Allowable values:	no grouping
Default value:	no grouping
Notes:	This attribute should always be set to no grouping.
remote-process-initialization-string	A command string that G2 sends to the bridge whenever the bridge is started to select operating options. The string consists of the name of OPC Server and the group setting.

Attribute	Description
Allowable values:	The syntax of the remote process initialization string is as follows: "*server-name group-name update-rate deadband mode*" where: *server-name*: The name of the OPC Server. *group-name*: The OPC group name. *update-rate*: The rate at which to update the group's items, in milliseconds. *deadband*: A float value from 0.0 to 100.0. If the OPC Server does not support this deadband, the default value is 0.0. *mode*: The mode of the communication between the bridge and the OPC Server. The options are: • S: Synchronous mode • A: Asynchronous mode These parameters must be escaped with only one space character. For example: "Integ.PI.IOPC.1 group1 5000 0.5 A"

Attribute	Description
Allowable values: (continued)	To connect the G2-OPC Client Bridge to a remote OPC Server, you can specify the node name in the remote process initialization string. For example, if the OPC Server named IntegrationObjects.OPC.PI is running on MachineA, and its IP address is 65.108.62.138, the remote process initialization string can be one of the following: "MachineA:IntegrationObjects.OPC.PI Group1 1000 0.0 A" or "\\MachineA:IntegrationObjects.OPC.PI Group1 1000 0.0 A" or "65.108.62.138:IntegrationObjects.OPC.PI Group1 1000 0.0 A" **Note**: The OPC server should be registered in the machine on which the bridge is running.
Default value:	none

Attribute	Description
Notes:	If the remote-process-initialization-string is empty, only remote procedures can be used for displaying the available OPC Servers. If the remote-process-initialization-string contains only the name of the OPC Server, the connection will not allow OPC items to be managed. When the *update-rate*, *dead-band*, and *mode* are not specified, the *mode* is asynchronous and the *update-rate* is provided by the OPC Server. If the name of the OPC server contains space characters, the name must be enclosed in quotes("). For example: "@"Integration Objects.PI.OPC.1@" Group1 1000 0.5 A" Also, if the OPC Server named Integration Objects.OPC.PI is running on MachineA, and its IP address is 65.108.62.138, the remote process initialization string can be one of the following: "@"MachineA:Integration Objects.PI.OPC.1@" Group1 1000 0.5 A" or "@"65.108.62.138:Integration Objects.PI.OPC.1@" Group1 1000 0.5 A"
gsi-interface-status	The current connection status.

Attribute	Description
Allowable values:	2 (OK): The connection between the G2 process and the bridge process is successful and being maintained. 1 (Initializing): The OPC system is initializing. When G2 receives this code, it suspends sending messages to the bridge process until it receives an OK code. 0 (Waiting): The interface is either disabled or inactive. -1 (Timeout): The G2 process has not heard from the bridge process within the interface-timeout-period specified for the interface object. The connection has timed out. This code may also indicate that a communication overload has occurred. An alarm condition is not necessary, since the bridge status usually returns to 2 without intervention. -2 (Error): An error condition occurred. The connection between G2 and the G2-OPC Client Bridge bridge has been interrupted.
Default value:	none
Notes:	If the bridge cannot establish a connection to the OPC server, the GSI interface status is automatically set to (–2). This may occur when a wrong OPC server name is supplied.
interval-to-poll-external-system	Controls the polling interval. The value of this attribute is not supported when poll-external-system-for-data is set to no.
Allowable values:	A time interval.
Default value:	none

Connecting G2 to the Bridge Process

To connect G2 to an OPC Server you must first create and configure a gsi-opc-interface object. The G2-OPC Client Bridge process connects to the OPC server, using the remote-process-initialization-string. The connection is established automatically when you perform the following steps.

To connect to the bridge:

1. Start G2.
2. Enable the gsi-opc-interface object corresponding to the G2-OPC Client Bridge process.
3. Edit the interface object's gsi-connection-configuration attribute to specify the settings required for the connection, described in the attributes table in Attributes of gsi-opc-interface.

4

Accessing OPC Server Data from G2

Describes how to access OPC Server data from G2.

Introduction

The G2-OPC Client Bridge provides two facilities for accessing OPC Server data from your G2 application:

- Subclasses of gsi-opc-item that enable the representation of current values of OPC Server variables in G2.
- RPCs (Remote Procedure Calls) for manipulating OPC Server data and browsing available data items in the OPC Server.

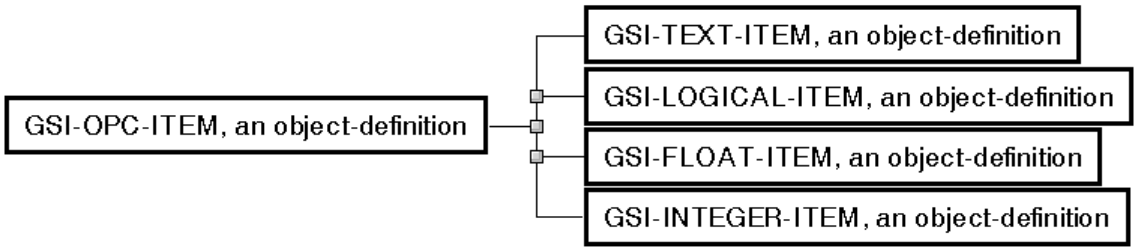
Accessing Data from the OPC Server

To access data located in the OPC Server, you create instances of gsi-opc-item variable objects in the G2 application. Each gsi-opc-item in your application represents a data source in the OPC Server. The value of the item is the last available value of the corresponding data point in the OPC Server.

Since the OPC item are always integer, float, text, or logical, objects in the application will always be instances of one of the following subclasses:

- gsi-float-item: used for floating point quantities.
- gsi-integer-item: used for integer point quantities.
- gsi-text-item: used for string point.
- gsi-logical-item: used for logical (Boolean) point.

Here is the class hierarchy of gsi-opc-item:



Creating a gsi-opc-item Variable

You can obtain any OPC item (Real, Integer, Boolean, or Text) by creating an instance of one of the gsi-opc-item subclasses. The attributes of the gsi-opc-item instance that you must configure are:

Attribute	Description
direct-superior-classes	The direct superior classes of the OPC item. Each OPC item must have gsi-data-service, gsi-message-service, and one of the gsi-opc-item variable subclasses as the superior classes.
class-specific-attributes	You must define the itemid and accesspath attributes to provide a unique identifier for each OPC item. The value of these attributes distinguishes the OPC item from all other GSI variables.

The gsi-opc-item instance is an object definition in the G2 KB, which represents a data point in the OPC Server.

Configuring Variables to Get Data

To configure a variable to get data, create a class definition that inherits from a gsi-opc-item subclass and configure its class-specific attributes. For example, to create an OPC float item with 24P618RC.PV as the itemid to read data from the OPC Server using the gsi-opc-interface object named phd, create an instance of gsi-float-item, set the value of the gsi-interface-name attribute to phd, and set the itemid attribute to "24P618RC.PV".

This table shows how to configure the variable:

24P618RC.PV, a gsi-float-item	
Options	do not forward chain, breadth first backward chain
Notes	GSI-FLOAT-ITEM-XXX-2: OK
Item configuration	none
Names	24P618RC.PV
Tracing and breakpoints	default
GSI interface name	phd
Data server for messages	gsi-data-server
Data type	float
Initial value	none
Last recorded value	no value
History keeping spec	do not keep history
Validity interval	indefinite
Formula	none
Simulation details	no simulation formula yet
Initial value for simulation	default
Data server	GSI data server
Default update interval	none
GSI variable status	0
Item-id	"24P618RC.PV"
Access path	""

Note When using asynchronous communication mode, you should set the default-update-interval to a high value, such as one week, so that the GSI variable is registered when you restart the G2 knowledge base. Another solution is to call specific G2 procedures when starting the KB, as described in Appendix B.

Reading Values from the OPC Server

To read the data, the bridge can work in asynchronous or synchronous mode.

To configure the bridge to work in asynchronous mode:

➔ In the remote-process-initialization-string, set the value of the mode argument to A.

For example, the following remote initialization string receives data with an update interval of 3 seconds:

```
"MyServerName MyGroupName 3000 2.5 A"
```

Note The external-system-has-a-scheduler attribute should be set to yes.

To configure the bridge to work in synchronous mode:

➔ In the remote-process-initialization-string, set the value of the mode argument to S.

For example:

```
"MyServerName MyGroupName 3000 2.5 S"
```

Note The external-system-has-a-scheduler attribute should be set to no.

Data Type Conversions

When the G2-OPC Client Bridge first sees an OPC variable, it tells the server to monitor the specified item. It also tells the server in which data format it expects to receive the item's value. According to the OPC Data Access specification, the server is responsible for either converting the data to the requested format or informing the client that is not possible.

The data type that will be requested depends upon the type of the OPC-item variable. The following table shows this relationship:

Variable Type	Requested Data Type
gsi-integer-item	VT_I4
gsi-float-item	VT_R8
gsi-logical-item	VT_BOOL
gsi-text-item	VT_BSTR

If the server cannot perform the conversion, the GSI variable status is typically set to 203. However, this depends upon the server. In some cases, the GSI variable status will be set to 300.

Combining Asynchronous and Synchronous Modes

To perform synchronous reading of a variable in asynchronous mode, you need to use a display item, such as a readout table, to request the value of the GSI Variable. A readout table requests the value when the validity interval of the GSI variable has expired. You can also use a rule.

The `read-modes.kb` file shows three cases:

- The first shows how to configure the KB for asynchronous mode only.
- The second shows how to configure the KB for synchronous mode only.
- The third shows how to use both modes concurrently.

For further information, see "Returning Solicited Data to G2" in Chapter 3, "Preparing the Bridge User Code" in the *G2 Gateway Bridge Developer's Guide*.

Filtering Data Sent to G2 using the GSI Variable Status

In some cases when the quality is bad, the value returned by the OPC Server can corrupt the data history. You can use a quality filter to filters return values, based on the GSI variable status.

For a list of GSI variable status values, see Appendix , Gsi-Opc-Item Status Values.

For each quality type (Good, Uncertain, Bad), the filter defines three groups. The first contains the list of qualities for which the value is always updated. The second group includes a list of qualities that can be enabled or disabled on demand. The third group includes the list of qualities that do not cause updating. Further, each quality type has two associated levels.

The following table describes when the GSI variable value is updated, based on the quality type, group, and level:

Quality		GSI Status	Level 0	Level 1
Good	Group 1	0	**Updated**	**Updated**
	Group 2	506	Not Updated	**Updated**
	Group 3	555	Not Updated	Not Updated

Quality		GSI Status	Level 0	Level 1
Bad	Group 1	300, 301, 302, 303, 304,305, 306, 307	Not Updated	**Updated**
	Group 2	333	Not Updated	Not Updated
Uncertain	Group 1	404, 405	**Updated**	**Updated**
	Group 2	400, 401,406	Not Updated	**Updated**
	Group 3	444	Not Updated	Not Updated

The quality filter is composed of three digits, as follows:

This digit...	Corresponds to the filter level (0 or 1) applied to this quality...
First	Good
Second	Bad
Third	Uncertain

The GSI variables that will be updated are those whose GSI status values for each quality (Good, Bad, Uncertain) and each group (Group 1, Group 2, Group 3) show **Updated** in the specified Level column (Level 0 or Level 1) in the table above.

All combinations of the three digits are possible. Examples include: 100, 101, 111, 001, 011, 010.

The default filter is 111, which updates GSI variables with a status value of 0, 506, 300, 301, 302, 303, 304,305, 306, 307, 404, 405, 400, 401, and 406.

For example, for a quality filter of 100:

This digit...	Corresponds to the filter level applied to this quality...	Which means...
1	Good	Only GSI variables with a status value of 0 (Group 1) or 506 (Group 2) will be updated.
0	Bad	No updates will occur for any bad quality.
0	Uncertain	Only GSI variables with a status value of 404 of 405 (Group 1) will be updated.

For a quality filter of 001:

This digit...	Corresponds to the filter level applied to this quality...	Which means...
0	Good	Only GSI variables with a status value of 0 (Group1) will be updated.
0	Bad	No updates will occur for any bad quality.
1	Uncertain	Only GSI variables with a status value of 404, 405 (Group 1) and 400, 401, or 406 (Group 2) will be updated.

Writing Values to the OPC Server

It is also possible to write data to the OPC Server. In synchronous mode, the data is written directly to the device and the response is returned immediately. In asynchronous mode, the OPC Server writes the data to a logical queue and the response is returned through a callback.

Remote Procedure Calls

The G2-OPC Client Bridge provides remote procedure calls for:

- Managing OPC Servers.
- Managing OPC Items.
- Browsing OPC Servers.
- Logging.

Managing OPC Servers

rpc-opc-getserverlist

Returns the list of available OPC Servers.

Synopsis

```
rpc-opc-getserverlist
    ( )
    -> ids: class text-list
```

Return Value	Description
ids	The list of program IDs of OPC Servers registered in the machine where the G2-OPC Client Bridge process is running.

rpc-opc-getserverstats

Requests the current OPC Server status.

Synopsis

rpc-opc-getserverstats
( )
-> (*current-time*: float, *start-time*: float, *group-count*: integer, *band-width*: integer, *version*: text, *vendor*: text, *error-code*: integer)

Return Values

Return Value	Description
current-time	The current time of OPC Server in UNIX time.
start-time	The time when the process started in UNIX time.
group-count	The total number of groups managed by the server.
band-width	The approximate percent bandwidth currently in use by server.
version	The OPC Server version.
vendor	Vendor information for the OPC Server.
error-code	0: Success. -1: Failed to get OPC Server state.

rpc-opc-getactualupdaterate

Returns the update rate supported by the OPC Server. The requested update rate of the client may differ from that of the server. In all cases, the server should respond with an update rate that is as close as possible to that requested.

Synopsis

rpc-opc-getactualupdaterate
( )
-> (*update-rate*: integer, *error-code*: integer)

Return Value	Description
update-rate	The value of update rate in milliseconds.
error-code	0: Success. -1: Failed to get update rate.

Example

Suppose the update rate supplied by the server is 250 milliseconds. The value of UpdateRate is 250.

```
UpdateRate, ErrorCode = call rpc-opc-getactualupdaterate()
```

g2-opc-set-groupstate

Allows the client to manage the state of the group.

Synopsis

g2-opc-set-groupstate
(*update-rate*: integer, *dead-band*: float, *active-state*: truth-value)
-> *error-code*: integer, *description*: text

Argument	Description
update-rate	The new update rates value in milliseconds.
dead-band	The percent change in an item value that will cause an exception report of that value to a client.
active-state	The new active state of the group.

Return Value	Description
error-code	0: Success. 1: The server does not support the requested data rate but will use the closest available rate. -1: Invalid group handle. -2: The operation failed for unknown reasons. -3: Not enough memory. -4: An argument to the RPC was invalid. -5: Invalid argument type. -6: Invalid number of arguments.
description	A descriptive status message.

Example

To stop asynchronous call updates:

```
error-code: integer;
description: text

error-code, description = g2-opc-set-groupstate(1000, 0.0 ,false);
```

g2-opc-set-qualityfilterlevel

Sets the filtering level of values returned to G2.

Synopsis

```
g2-opc-set-qualityfilterlevel
    (quality-filter-level: integer)
    -> error-code: integer
```

Argument	Description
quality-filter-level	The filter level to be applied to the GSI variable status.

Return Value	Description
error-code	0: Success. 1: Invalid *quality-filter-level* value. -1: Failed to set the *quality-filter-level* value. -2: Invalid number of arguments. **Note**: When *error-code* differs from 0, the old value is kept.

Example

If you to update only those GSI variables with the status of 0, 506, 404, or 405, you would set the quality filter level to 100, as follows:

- For a Good quality, to update only 0 and 506, you set the filter level to 1.
- For a Bad quality, you set the filter level to 0 for no updates.
- For an Uncertain quality, to update only 404 and 405, you set the filter level to 0.

For more information on how to specify the quality filter, see Filtering Data Sent to G2 using the GSI Variable Status.

The RPC call looks like this:

```
error-code: integer;

error-code = call g2-opc-set-qualityfilterlevel(100)
```

Managing OPC Items

rpc-opc-triggerdeviceread

Allows G2 to refresh on demand all data items directly from the device.

Synopsis

```
rpc-opc-triggerdeviceread
    ( )
    -> error-code: integer
```

Return Value	Description
error-code	0: Success. -1: Failed to get refresh item data.

rpc-opc-writevqt

Writes the value, quality and timestamp of the specified item-id. This RPC has similar functionality to the G2 set action except it also writes quality and timestamp. It is not necessary to define a group to use this RPC. This RPC is specific to OPC Data Access 3.0 compliant server.

Remote Process Initialization String containing only the OPC server name allows to call this RPC.

Example:

RPIS= “MyOPCServerName”

Synopsis

rpc-opc-writevqt
(*item-id* : text, *input-value*: value, *input-timestamp*: text, *input-quality*: integer)
-> (*error-code*: integer, *description*: text)

Argument	Description
item-id	The item ID.
input-value	The desired value to be written to the OPC server.
input-timestamp	Time stamp for this item’s value. The time is provided with millisecond precision. The format is “YYYY/MM/DD hh:mm:ss.mmm”. For example: "2004/06/10 14:32:12.000"
input-quality	The quality of this item. Possible quality values are: 0: GOOD, 300: BAD 400: Uncertain.

Return Value	Description
error-code	0 : Success. -1: Failed. 1 : Partially succeeded.
description	A description of the error message.

Browsing OPC Servers

When the OPC Server allows browsing, the client can browse the available data items in the server to obtain the list of the valid definitions for an itemID. The browse position is initially set to the root of the address space. The client can optionally choose a starting point within a hierarchical space by calling rpc-opc-changebrowserposition. For further information, see the "OPC Common Definition and Interfaces" documentation.

rpc-opc-setbrowserfilter

Sets the filter applied to the itemid when returning the item list, using rpc-opc-getbrowserleaves.

Synopsis

rpc-opc-setbrowserfilter
(*filter*: text)
-> *error-code*: integer

Argument	Description
filter	The text of the filter to apply. The default value is "*".

Return Value	Description
error-code	0: Success. -1: Failed to set the value of the filter.

Example

To get all available items in the server, you can use "*" as the filter argument, the default:

```
ErrorCode = call rpc-opc-setbrowserfilter("*")
```

rpc-opc-setbrowserdatatype

Sets the filter by the available data types to return the item list, using rpc-opc-getbrowserleaves.

Synopsis

rpc-opc-setbrowserdatatype
(*type*: integer)
-> *error-code*: integer

Argument	Description
type	The type of data to apply as a filter. The options are: 0: No filter 1: Short integer 2: Long integer 3: Single-precision float 4: Double-precision float 5: Text 6: Truth-value The default value is 0.

Return Value	Description
error-code	0: Success. -1: Failed to set the value of the filter.

Example

To get all available Short Integer values in the server, you can use 1 as the filter argument:

```
ErrorCode = call rpc-opc-setbrowserdatatype (1)
```

rpc-opc-setbrowserrights

Sets the access rights to the server.

Synopsis

rpc-opc-setbrowserrights
(*right*: integer)
-> *error-code*: integer

Argument	Description
right	The access right. The options are: 0: No filter 1: The client can read the data item's value 2: The client can edit the data item's value The default value is 0.

Return Value	Description
error-code	0: Success. -1: Failed to set the value of the filter.

rpc-opc-getbrowserleaves

Provides a way to move "up" or "down" or "to" in a hierarchical space.

Synopsis

rpc-opc-getbrowserleaves
( )
-> *leaves*: class text-list

Return Value	Description
leaves	The list of available leaves at the current position.

rpc-opc-getbrowserbranches

Returns a list of available branches at the current position.

Synopsis

rpc-opc-getbrowserbranches
( )
-> *branch-list*: class text list

Return Value	Description
branch-list	A list containing the available branch at current position.

rpc-opc-changebrowserposition

Provides a way to move “up” or “down” or “to” in a hierarchical space.

Synopsis

rpc-opc-changebrowserposition
(*direction-code*: integer, *branch-name*: text)
-> *error-code*: integer

Argument	Description
direction-code	The OPC-defined code for the direction in which to move. The options are: • 1 (OPC_BROWSE_UP) — Move up one level. • 2 (OPC_BROWSE_DOWN) — Move down one level to the branch or leaf specified in the second parameter. • 3 (OPC_BROWSE_TO) — Move to an absolute location.
branch-name	When moving up, this argument is ignored and should be set to "". When moving down, this argument specifies the name of the branch to move into. When moving "to", this argument specifies the fully qualified destination or "" to move to the root.

Return Value	Description
error-code	0: Success. -1: Failed. 1: An argument to the function was invalid.

rpc-opc-getitemid

Converts a leaf name to a fully qualified item ID.

Synopsis

rpc-opc-getitemid
(*name*: text)
-> (*item-id*: text)

Argument	Description
name	The name of a branch or leaf at the current level.

Return Value	Description
item-id	The item ID.

Example

To obtain the item ID for the item named PV, call rpc-opc-getitemid with the name PV. Item contains the value "125P425RC.PV", which represent the item ID of the OPC item.

```
ItemId = call rpc-opc-getitemid ("PV")
```

rpc-opc-browse

Browses a single branch of the address space and returns a text list containing the list of branches or leaves depending on the browse filter. This RPC is specific to OPC Data Access 3.0 servers.

Synopsis

rpc-opc-browse
(*item-id*: text; *browse-type*: integer; *browse-filter*: text)
-> *items-id:* class text-list, *error-code*: integer, *error-desc*: text

Argument	Description
item-id	The name of a branch at the current level.
browse-type	The subset of browse elements to return, as an integer: 1: A branch that has children, or possibly has children, that is also an item. 2: A branch that has children, or possibly has children. 3: An item that is not a branch.
browse-filter	The text of the filter to apply. The default value is "".

Return Value	Description
item-id	The item ID.
error-code	0: Success. -1: Failed. 1: Partially succeeded.
description	A descriptive status message.

rpc-opc-browsepart

Browses a single branch of the address space and returns a text list containing the list of branches or leaves depending on the browse filter. This RPC is specific to OPC Data Access 3.0 servers.

Synopsis

rpc-opc-browse
(*item-id*: text, *continuation-point*: text, *max-elements-returned*: integer; *browse-type*: integer; *browse-filter*: text)
-> *items-ids*: class text-list, *continuation-point*: text, *more-elements*: truth-value, *error-code*: integer , *error-desc*: text

Argument	Description
item-id	The name of a branch at the current level.
continuation-point	If this is a secondary call to rpc-opc-browse, the previous call might have returned a continuation point from which the browse can restart. Clients must pass a null string in the initial call to rpc-opc-browse. This is an opaque value, which the server creates. A continuation point is returned if a server supports continuation points and the reply is larger than *max-elements-returned*. The continuation point allows the client to resume the browse from the previous completion point.

Argument	Description
max-elements-returned	The maximum number of elements to return. If the server supports continuation points, then the server can return a continuation point at a value less than *max-elements-returned*. If the server does not support continuation points, and more than the maximum are available, then the server returns the maximum number of elements and sets the pbMoreElements parameter to true. If *max-elements-returned* is 0, then there is no client-side restriction on the number of returned elements.
browse-type	The subset of browse elements to return, as an integer: 1: A branch that has children, or possibly has children, that is also an item. 2: A branch that has children, or possibly has children. 3: An item that is not a branch.
browse-filter	The text of the filter to apply. The default value is "".

Return Value	Description
item-ids	A list containing the available branch and leaves at the current position.
continuation-point	See the description of the *continuation-point* argument.
more-elements	true if the server does not support continuation points and there are more elements than *max-elements-returned*.
error-code	0: Success. -1: Failed. 1: Partially succeeded.
description	A descriptive status message.

rpc-opc-getproperties

Returns a list of the Item properties associated with an item.

Synopsis

rpc-opc-getproperties
(*item-id*: text)
-> *att-count*: integer, *property-id*: class value-list,
property-desc: class text-list, *property-type*: class text-list,
property-val: class text-list, *error-code* : integer, *error-description* : text

Argument	Description
item-id	The name of a branch for which the caller wants to read properties.

Return Value	Description
att-count	The number of properties returned.
property-id	The properties ID.
property-desc	The property description.
property-type	The property data type.
property-val	The value of the property.
error-code	Error code.
description	A descriptive status message.

Logging

The following RPCs allows the configuration of the bridge for logging purposes.

g2-opc-set-tracelevel

Sets the level of message tracing.

Synopsis

g2-opc-set-tracelevel
(*trace-level*: integer)
-> *error-code*: integer

Argument	Description
trace-level	The trace level. The options are: 0: Log fatal error messages; whenever this error occurs, the context is shutdown. 1: Log the critical error messages. 2: Log error messages. 3: Log warning messages. 4: Log information messages. The default value is trace level 0.

Return Value	Description
error-code	0: Success. -1: Failed to set the trace level.

Example

To set the trace level to its highest value:

```
ErrorCode = call g2-opc-set-tracelevel(4)
```

g2-opc-tracestart

Starts the trace log.

Synopsis

g2-opc-tracestart
( )
-> *error-code*: integer

Return Value	Description
error-code	0: Success. -1: Failed to start trace logging.

Example

To start trace logging:

```
ErrorCode = call g2-opc-tracestart ()
```

g2-opc-tracestop

Stops trace logging.

Synopsis

g2-opc-tracestop
( )
-> *error-code*: integer

Return Value	Description
error-code	0: Success. -1: Failed to stop trace logging.

Example

To stop trace logging:

```
ErrorCode = call g2-opc-tracestop ()
```

g2-opc-trace-to-g2

Displays log messages in G2.

Synopsis

g2-opc-trace-to-g2
(*display-in-g2*: truth-value)
-> *error-code*: integer

Argument	Description
display-in-g2	true: Display log messages in G2. false: Stop displaying log messages in G2. The default value is false.

Return Value	Description
error-code	0: Success. -1: Failed to display in G2.

g2-opc-trace-to-stdout

Displays log message in the G2-OPC Client Bridge console.

Synopsis

g2-opc-trace-to-stdout
(*display-in-console*: truth-value)
-> *error-code*: integer

Argument	Description
display-in-console	true: Display log messages in the console. false: Stop displaying log messages in the console. The default value is false.

Return Value	Description
error-code	0: Success. -1: Failed to display in the console.

1

Configuration File

Describes the log file settings you can edit in the configuration file.

Introduction

The G2-OPC Client Bridge includes a configuration file named `ConfigFile.ini`, which includes a number of parameters related to logging, filtering, and server status. These parameters all have default settings, which you can change at startup by editing the configuration file.

The `ConfigFile.ini` file is a text file that is divided into sections with the section name in brackets. Following each section are parameter/value lines. Note that the section header is required and each key must appear under its relevant section.

You can also change the configuration file at runtime, as described in Configuring the G2-OPC Client Bridge at Runtime.

To change the configuration file:

1 Open `ConfigFile.ini` in a text editor.

2 Edit any of the parameters listed in the following table:

Parameter	Description	Default Value
Log Settings		
`LogFileName`	Log file name, if not specified the default value is used.	`LogEvent.LOG`
`LogFileMaxSize`	The maximum log file size, in bytes. Once this size is reached during run-time, the log file is overwritten.	`1048576*2 ~ 2 Mo`
`LogLevel`	The log level. Possible values are: 0: Only fatal error messages are logged. 1: All critical error messages are logged. 2: All errors are logged. 3: All warnings are logged. 4: All information is logged.	`0`
`LogToG2`	`TRUE`: Messages appear on G2 Message Board as they are logged. `FALSE`: Messages are only logged to file.	`FALSE`
`ArchiveLastLog`	`TRUE` : Old file is copied to an intermediate file with incremental extension, before being overwritten. `FALSE` : Any pre-existing log file is erased and overwritten at startup.	`FALSE`
`LogToScreen`	`TRUE` : Messages appear on console screen as they are logged. `FALSE` : Messages are only logged to file.	`FALSE`

Parameter	Description	Default Value
Quality		
QualityFilter	The filter that will be applied to the GSI variable's status value. See Filtering Data Sent to G2 using the GSI Variable Status.	111
Server Status		
CheckStatus	Enables/disables OPC server status check.	TRUE
CheckFrequency	The frequency at which the bridge checks the OPC server status in seconds.	5
UseG2TimeStamp	FALSE: The bridge uses the timestamp provided by the OPC Server. TRUE: The bridge ignores the OPC Server timestamp and uses G2 timestamp.	FALSE
InitSecurity	FALSE: User should apply all required security settings to enable the OPC Server to access the bridge to send data updates. In fact, all security settings you have done on the server have to be replicated for the OPC Bridge application. TRUE: Automatically, initialization of the bridge security allows the client to have access to it.	TRUE

3 Save the file for the log settings to take effect.

Here is a sample configuration file that you might create to specify a different configuration:

```
[LogSetting]
LogLevel=4
LogToScreen=FALSE
ArchiveLastLog=TRUE
LogToG2=FALSE
LogFileName=LogEvent.LOG
LogFileMaxSize=2097152
[Quality]
QualityFilter=111
[ServerStatus]
CheckStatus=TRUE
CheckFrequency=5
UseG2TimeStamp=FALSE
InitSecurity=TRUE
```

1

Configuring GSI Variables in Asynchronous Mode

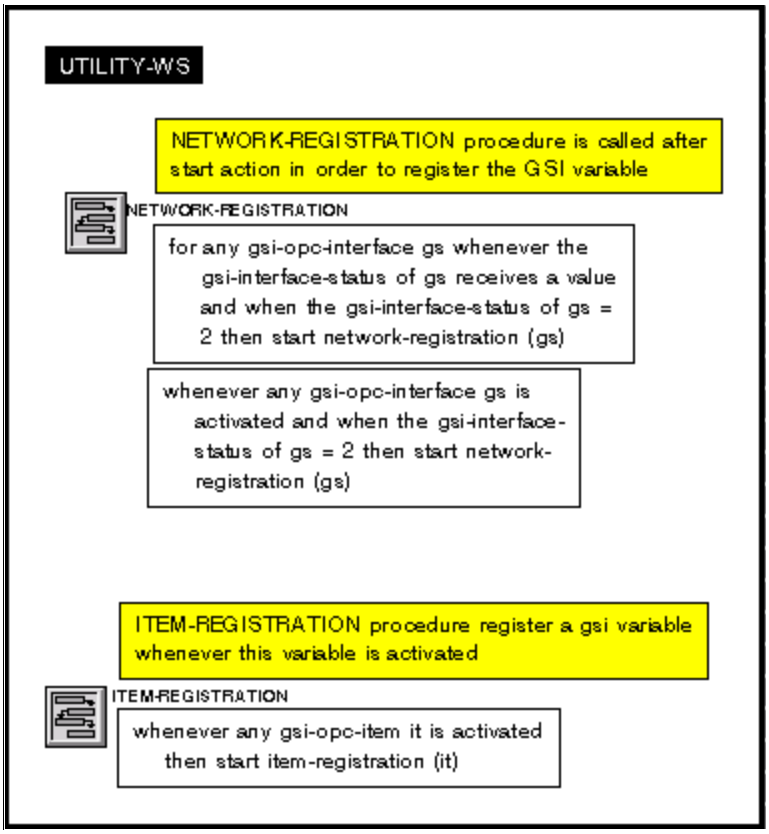
Describes how to configure gsi-variables in asynchronous mode.

Introduction

The following figure shows how to set gsi-variables in asynchronous communication mode. The two procedures network-registration and item-registration register the gsi-variables after starting communication between the bridge and OPC Servers.

Note In synchronous communication mode, the utility-ws workspace should be deleted or disabled.

Here is the utility-ws workspace:



Here is the network-registration procedure:

NETWORK-REGISTRATION, a procedure	
Notes	OK
Authors	haifa (23 May 2002 12:51 p.m.)
Change log	0 entries
Item configuration	none
Tracing and breakpoints	default
Class of procedure invocation	none
Default procedure priority	6
Uninterrupted procedure execution limit	use default

```
network-registration(icp-item : class gsi-interface )
gsi-var: class gsi-opc-item;
handle: integer;
gsi-var-int: symbol;

begin

  if icp-item  has a name then
   for gsi-var = each gsi-opc-item do
        if the gsi-interface-name of gsi-var exists and  the  gsi-interface-
          name of gsi-var /= the symbol none then
              begin
                  gsi-var-int = the gsi-interface-name of gsi-var;
                  if gsi-var-int   = the name of icp-item then
                  handle = call g2-register-on-network (gsi-var, icp-item);
             end;
  end;
end
```

Here is the item-registration procedure:

ITEM-REGISTRATION, a procedure	
Notes	OK
Authors	haifa (23 May 2002 12:41 p.m.)
Change log	0 entries
Item configuration	none
Tracing and breakpoints	default
Class of procedure invocation	none
Default procedure priority	6
Uninterrupted procedure execution limit	use default

```
item-registration(I: class item )
handle: integer;
name-int: symbol;

begin
    if the gsi-interface-name of I exists and  the  gsi-interface-name of I /=
     the symbol none then
        begin

          name-int = the gsi-interface-name of I;
          if there exists a gsi-interface gs named by name-int and the
            gsi-interface-status of gs =2 then
              handle = call g2-register-on-network (I, gs);
        end;

end
```

1

Gsi-Opc-Item Status Values

Describes status values for gsi-opc-item when connection takes place and errors that can occur.

Introduction

The following tables describe the possible values for the gsi-status attribute of the GSI variable when connection takes place between G2 and the bridge process.

General OPC Status

The following status values can occur when the GSI interface object makes a connection:

Value	OPC Error	Description
0	OPC_GOOD_NONSPEC	The value is good. There is no special condition.
50	GSI_UNSUPPORTED_ TYPE	The data type of the variable is not supported.
51	GSI_UNREGISTERED	The variable is not correctly registered with the bridge.
52	GSI_DUPLICATED_ITEM	The item is registered more than once with the same item id and access path.
506	OPC_GOOD_LOCAL_ OVERRIDE	There is some server specific problem with the configuration.
555	OPC_GOOD_INVALID_ QUALITY	Receive an invalid good quality “Not used by OPC”

The following status values indicate problems that occur when the GSI variable is registered:

Value	OPC Error	Description
100	OPC_NO_GROUP_DEFINED	No valid group defined for the item.
101	G2OPC_NO_SERVER_DEFINED	
102	G2OPC_DEF_ATTR_BAD	
103	G2OPC_CANT_ADD_TO_PUB	
105	G2OPC_QI_FAILED	
106	G2OPC_MEM_ALLOC_FAILED	
109	G2OPC_READ_FAILURE	
110	G2OPC_UNEXPECTED	

Value	OPC Error	Description
111	OPC_MAX_ITEM_REACHED	The maximum number of items is reached.
199	OPC_E_BADRIGHTS	The item access rights do not allow the read/write operation.
200	OPC_E_INVALIDITEMID	The passed item id is invalid.
201	OPC_E_UNKNOWNITEMID	The item is no longer available in the server address space.
202	G2OPC_BAD_ACCESSPATH	
203	OPC_E_BADTYPE	The item data type is bad.
205	G2OPC_WRITE_FAILURE	
206	OPC_ADD_ITEM_FAIL	Unable to add item, no specific reason is known.

Bad Quality Status

The following status values indicate that the value of the GSI variable is bad:

Value	Status Value	Description
300	OPC_BAD_NONSPEC	The value is bad but no specific reason is known.
301	OPC_BAD_CONFIG_ERROR	There is some server specific problem with the configuration.
302	OPC_BAD_NOT_CONNECTED	The input is required to be logically connected to something but is not.
303	OPC_BAD_DEVICE_FAILURE	A device failure has been detected.
304	OPC_BAD_SENSOR_FAILURE	A sensor failure had been detected.

Value	Status Value	Description
305	OPC_BAD_LAST_KNOWN_VAL	Communications have failed. However, the last known value is available.
306	OPC_BAD_COMM_FAILURE	Communications have failed. There is no last known available value.
307	OPC_BAD_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.
308	OPC_BAD_NOT_INITIALIZED	Attempt to read a value before the OPC Server has had a chance to initialize it.
333	OPC_BAD_INVALID_QUALITY	Receive an invalid bad quality “Not used by OPC”

Uncertain Quality Status

The following status values indicate that the value of the GSI variable is uncertain:

Value	Status Value	Description
400	OPC_UNC_NONSPEC	There is no specific reason why the value is uncertain.
401	OPC_UNC_LAST_USABLE_VAL	Whatever was writing this value has stopped doing so. The returned value should be regarded as ‘stale’.

Value	Status Value	Description
404	OPC_UNC_SENSOR_NOT_ ACCUR	Either the value has 'pegged' at one of the sensor limits (in which case the limit field should be set to 1 or 2) or the sensor is otherwise known to be out of calibration via some form of internal diagnostics
405	OPC_UNC_EGU_EXCEEDED	The returned value is outside the limits defined for this parameter.
406	OPC_UNC_SUB_NORMAL	The value is derived from multiple sources and has less than the required number of Good sources.
444	OPC_UNC_INVALID_QUALITY	Receive an invalid uncertain quality "Not used by OPC"

Invalid Quality

The following status values indicate that the value of the GSI variable is invalid:

Value	Status Value	Description
666	OPC_INVALID_QUALITY	The value quality of the item is invalid.

Index