

Integration Objects'

Solution for OPC DA-G2 Connectivity

G2 OPC Link
Version 3.2 Rev 1

User Guide

Compatibility

OPC DA 1.0a
OPC DA 2.00
OPC DA 2.05a
OPC DA 3.0

G2 OPC Link User Guide Version 3.2 Rev 1
January 2012

Copyright © 2005 – 2012 Integration Objects. All rights reserved.

No part of this document may be reproduced, stored in a retrieval system, translated, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Integration Objects.

Windows® and Windows NT® are registered trademarks of Microsoft Corporation.

Gensym, G2 and GSI are registered trademarks of Gensym Corporation.

TABLE OF CONTENTS

PREFACE	2
About this User Guide	2
Target Audience	2
Customer Support Service	2
INTRODUCTION	3
1. Overview	3
2. System Architecture	3
3. G2 Compatibility	4
4. Using Previous Knowledge Bases	4
5. Distribution Materials	4
6. System Requirements	4
GETTING STARTED	5
1. Installing G2 OPC Link	5
2. Starting Up G2 OPC Link Application	5
3. Stopping G2 OPC Link Application	6
4. Configuration G2 OPC Link at Runtime	7
5. Browsing OPC Servers	7
6. Removing G2 OPC Link Application	8
configURING G2 OPC LINK KNOWLEDGE BASE FILES	9
1. Introduction	9
2. Attributes of GSI-OPC	9
3. Connecting G2 To The Bridge	19
ACCESSING REAL TIME DATA OF OPC SERVERS FROM G2	20
1. Introduction	20
2. Accessing Data From The OPC Server	20
3. Creating a gsi-opc-item Variable	21
4. Configuring Variables to Get Data	21

5.	Reading Variables from the OPC Server.....	22
6.	Combining Asynchronous and Synchronous Mode.....	23
7.	Filtering Data Sent to G2 using the GSI Variable Status.....	23
8.	Writing Values to the OPC Server.....	25
9.	Remote Procedure Calls.....	25
9.1.	Managing OPC Servers.....	26
9.2.	Managing OPC Items.....	29
9.3.	Browsing OPC Servers.....	30
9.4.	Logging.....	36
	APPENDIX A: CONFIGURATION FILE.....	39
	APPENDIX B: CONFIGURING GSI VARIABLES IN ASYNCHRONOUS MODE.....	41
	APPENDIX C: Gsi-Opc-Item Status Values.....	45
1.	General OPC Status.....	45
2.	Bad Quality Status.....	46
3.	Uncertain Quality Status.....	46
4.	Invalid Quality.....	47
	APPENDIX D: DCOM CONFIGURATION FOR WINDOWS XP.....	48
1.	Setting Up Client DCOM Configuration.....	48
2.	Registering OPC Servers.....	52
3.	Configuring The Server For DCOM On Windows XP.....	54
4.	Configuring DCOM Settings For Your OPC Server.....	55

PREFACE

About this User Guide

This guide:

- Provides information on the G2 OPC Link features,
- Describes the system requirements,
- Explains how to install G2 OPC Link,
- Explains how to establish connections to multiple OPC Servers simultaneously and communicate with them through the G2 OPC Link bridge process.

Target Audience

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

Customer Support Service

Offices	Email
Houston, USA: +1 713 609 9208	Support: customerservice@integrationobjects.com
Genova, Italy: +39 34 75 83 93 47	Sales: sales@integrationobjects.com
Tunis, Tunisia: +216 71 861 803	Online: www.integrationobjects.com

INTRODUCTION

1. Overview

G2 OPC Link 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.

G2 OPC Link 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, G2 OPC Link provides remote procedure calls (RPCs) for managing and browsing an OPC Server. This data can be used for trend analysis and sophisticated inference by the G2 inference engine.

2. System Architecture

To retrieve process data from OPC Server(s) in real time, you can use G2 OPC Link in local and distributed configurations:

- In local configurations, G2 OPC Link 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, G2 OPC Link 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.

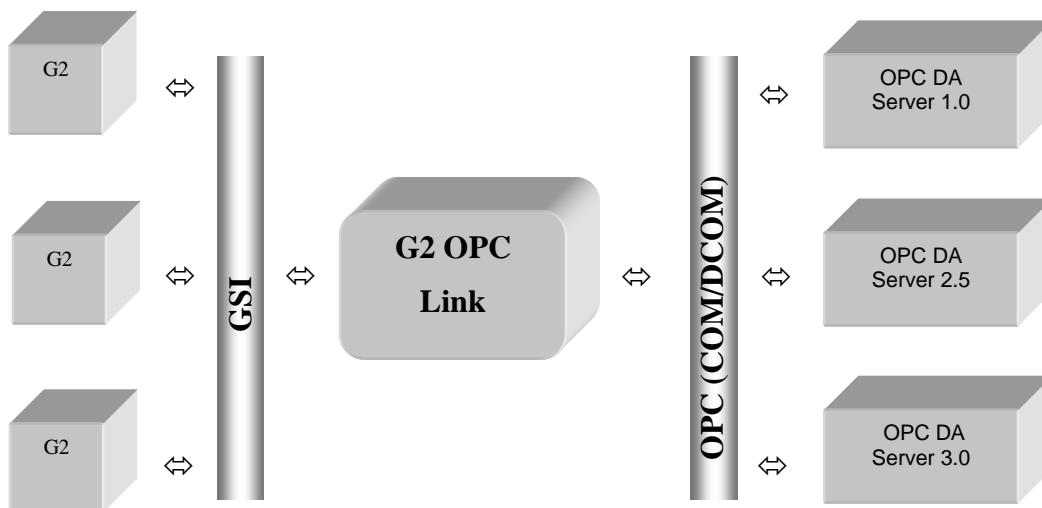


Figure 1: G2 OPC Link System Architecture

3. G2 Compatibility

The G2 OPC Link is compatible with G2 Version 6.0 Rev 0 or later on Windows platforms only.

4. Using Previous Knowledge Bases

A knowledge base built to use previous versions of G2 OPC Link will not be fully compatible with this version. You must replace the G2 OPC Link executable, G2opc.exe, and knowledge base, launch.kb, with files provided with G2 version 8.2 Rev 4.

5. Distribution Materials

The following kbs are supplied with the G2 OPC Link:

- browse.kb: contains browsing procedures
- read-modes.kb: contains asynchronous and synchronous configurations
- group-state.kb: contains a procedure that changes group update rate, deadband, and sets its status to active

6. System Requirements

- OPC Servers compliant with OPC Data Access Server Version 1.0 through 3.0 and all intermediate versions.
- Windows NT 4.0 (Service Pack 6a), Windows 2000, or Windows XP
- A minimum of 128 MB.

GETTING STARTED

This section explains how to install, run and remove G2 OPC Link application.

1. Installing G2 OPC Link

In order to install G2 OPC Link, the user should proceed with the following steps:

- Open the installation CD and double click on the setup exe file. This will launch the installation wizard.
- Follow the installation wizard as it guides you through the different setup steps.
- Click **Finish**.

The installation wizard copies all the necessary files to the target folder, creates a shortcut icon to launch the G2 OPC Link application from the start menu and makes an un-installation entry in the Add/Remove Programs in the Control Panel.

2. Starting Up G2 OPC Link Application

Integration Objects' G2 OPC Link can be manually started from its shortcut under the Programs menu:

Start → Programs → Integration Objects → G2 OPC Link → G2 OPC Link.

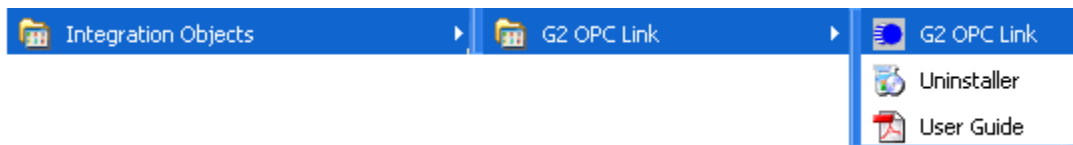


Figure 2: Launching G2 OPC Link

Once launched, the user interface of G2 OPC Link appears, as shown in the figure below:

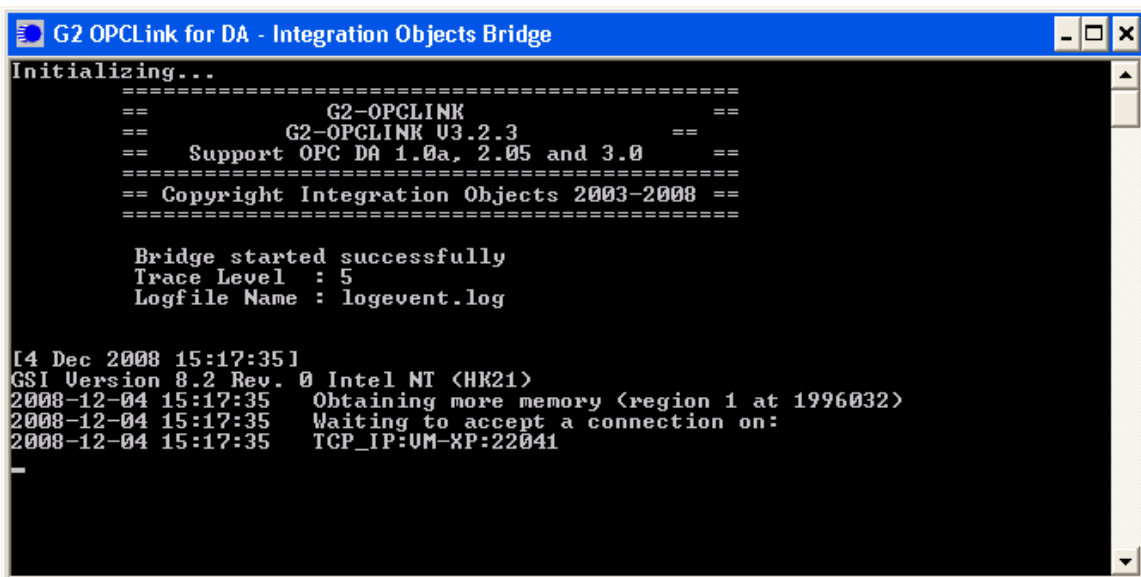


Figure 3: G2 OPC Link User Interface

An icon in the windows system tray will also be displayed (Figure 3).



Figure 4: G2 OPC Link System Tray Icon

Using a double in the system tray icon, a contextual menu will be displayed allowing you to show the control panel and the OPC Server auto-discovery user interfaces, to display the about box or to shut down the application (Figure 5).

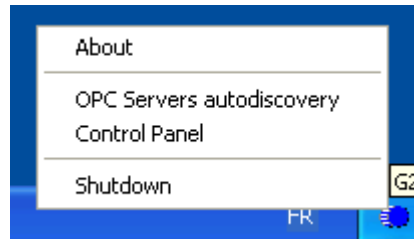


Figure 5: G2 OPC Link System Tray Menu

3. Stopping G2 OPC Link Application

To shutdown the bridge application, close the console window or press Ctrl + C. During shutdown, the bridge process closes all connections to OPC Servers and releases any allocated resources.

4. Configuration G2 OPC Link at Runtime

G2 OPC Link 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 A, "Configuration File" on page 41.

To configure G2 OPC Link at runtime:

- 1 Double-click the tray icon for G2 OPC Link to display its menu:



Figure 6: G2 OPC Link System Tray Menu – Control Panel

- 2 Choose Control Panel to display this dialog:

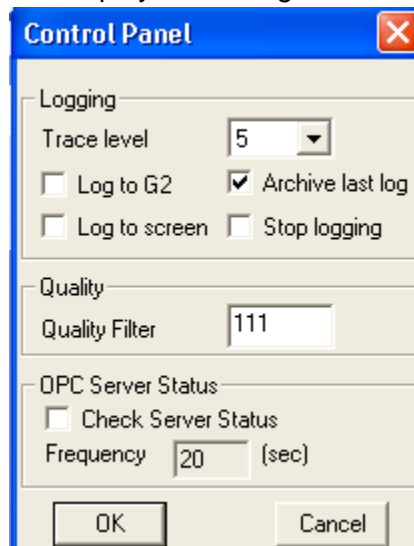


Figure 7: G2 OPC Link Control Panel

5. Browsing OPC Servers

To browse for an existing OPC server at runtime:

1. Double-click the tray icon to display its menu.
2. Choose OPC Servers auto discovery to display this dialog:

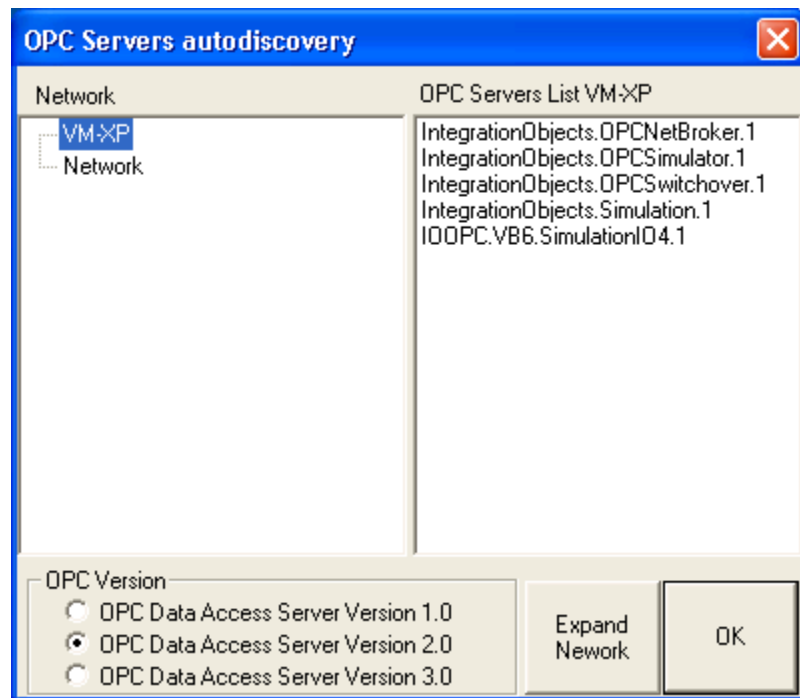


Figure 8: OPC Servers autodiscovery

6. Removing G2 OPC Link Application

To remove the G2 OPC Link from your machine, just click the **Uninstaller** shortcut icon from the start menu, as shown in the figure below.

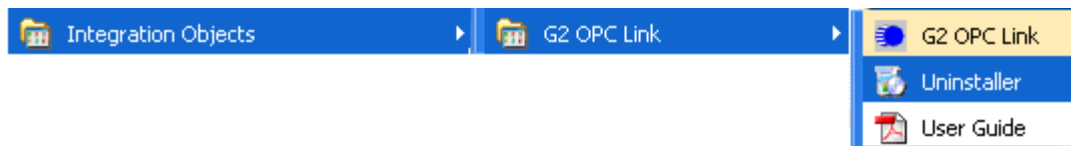


Figure 9: Removing G2 OPC Link

The G2 OPC Link can also be removed manually as follows:

1. Go to the Control Panel.
2. Click **Add/Remove Programs**.
3. In Add/Remove Programs dialog screen select "Integration Objects' G2 OPC Link".
4. Click **Change/Remove** then **OK**.

CONFIGURING G2 OPC LINK KNOWLEDGE BASE FILES

1. Introduction

G2 connects to the G2 OPC Link through a `gsi-opc-interface` object, which references an instance of the `gsi-opc-interface` class contained in the `launch.kb`. The attributes of the `gsi-opc-interface` instance specify parameters that define the connection between G2 and G2 OPC Link. This chapter describes the attributes of a `gsi-opc-interface` object, their definitions, and strategies for optimizing their settings.

2. Attributes of GSI-OPC

This interface object collects all the data necessary 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

Figure 10: gsi-opc-interface

Attribute	Description
names	The name of the interface object, which must be unique. Since this object represents the connection to a particular G2 OPC Link bridge process, you can choose a name that associates it with the OPC Server.
<i>Allowable values:</i>	Any symbol
<i>Default value:</i>	none
identifying- attributes	Uniquely identifies the variable objects to the OPC Server.
<i>Allowable values:</i>	A list of symbols
<i>Default value:</i>	Item-id, access-path
<i>Notes:</i>	The value of identifying-attributes should always be: Item-id, access-path 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
<i>Allowable values:</i>	<p data-bbox="589 289 786 310">Allowable values:</p> <p data-bbox="589 338 964 359">0: No warning or error messages.</p> <p data-bbox="589 386 943 407">1: Serious error messages only.</p> <p data-bbox="589 434 834 455">2: All error messages.</p> <p data-bbox="589 483 980 504">3: All error and warning messages.</p> <p data-bbox="589 531 1325 758">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 <code>gsi-interface-status</code> attribute. Messages are posted to the Operator Logbook by default.</p> <p data-bbox="589 785 1325 926">For example, when the warning message level is at 0 or 1, a failure to connect to a bridge causes the <code>gsi-interface-status</code> to change to -2 (Error), but no information is made available about why the failure occurred.</p>
<i>Default value:</i>	<p data-bbox="589 961 1325 1066">default to warning message level: The <code>interface-warning-message-level</code> takes on the value of the <code>warning-message-level</code> attribute in the Debugging Parameters system table.</p>
disable-interleaving-of-large-messages	<p data-bbox="589 1188 1325 1255">Controls whether G2 Gateway interleaves, or changes the transmission order of message packets</p>
<i>Allowable values:</i>	yes, no
<i>Default value:</i>	<p data-bbox="589 1394 786 1415">no, which means:</p> <ul data-bbox="639 1444 1325 1680" style="list-style-type: none"> <li data-bbox="639 1444 1325 1549">- G2 transmits messages in packets. A large message occupies several packets. A small message occupies a single packet. <li data-bbox="639 1577 1325 1680">- When more than one message requires transmission across an interface, G2 interleaves the packets that constitute the messages.

Attribute	Description
interface-timeout-period	<p>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.</p>
<i>Allowable values:</i>	<i>integer seconds</i>
<i>Default value:</i>	use default, which uses a default of 1 second
<i>Notes:</i>	See interface-initialization-timeout-period.
interface-initialization-timeout-period	<p>Specifies how long G2 waits to initialize a connection using Gensym (ICP) protocols. The following timeout intervals apply to GSI</p> <ul style="list-style-type: none"> - Interfaces: Establish a connection. - Initialize the connection. - Wait for a response. <p>This attribute applies to the second interval. The interface-timeout-period attribute specifies the timeout period for the first and third intervals.</p>
<i>Allowable values:</i>	<p>An integer specifying some number of seconds.</p> <p>Unlimited: The initialization interval never times out.</p> <p>Use default: The interface-initialization-timeoutperiod is the same as the interface-timeoutperiod.</p>
<i>Default value:</i>	<i>unlimited</i>

Attribute	Description
gsi-connection-configuration	<p>A G2 expression that describes the network connection between G2 and the G2 OPC Link bridge process. The expression specifies the type of network (DECnet or TCP/IP) and the network address of the bridge process.</p> <p><i>Allowable values:</i> tcp-ip host "<i>host</i>" port-number <i>port-number</i> where: "<i>host</i>" 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. <i>port-number</i> 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.</p> <p><i>Default value:</i> tcp-ip host "localhost" port-number 22041</p>
external-system-has-a-scheduler	<p>Determines whether scheduling for getting new data values is the responsibility of the OPC Server or G2.</p> <p><i>Allowable values:</i> 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.</p> <p><i>Default value:</i> yes</p> <p><i>Notes:</i> 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.</p>

Attribute	Description
poll-external- system-for-data	<p>Determines whether the G2 Gateway bridge receives unsolicited data from the external system by executing the callback function <code>gsi_g2_poll()</code> every cycle.</p> <p><i>Allowable values:</i> yes: The G2 Gateway bridge calls <code>gsi_g2_poll()</code> every cycle.</p> <p>no: The G2 Gateway bridge does not call <code>gsi_g2_poll()</code>.</p> <p><i>Default value:</i> no</p> <p><i>Notes:</i> This attribute should always be set to no.</p>
grouping- specification	<p>(Optional) Enables you to group requests for data service using one or more of the identifying attributes of a variable.</p> <p><i>Allowable values:</i> no grouping</p> <p><i>Default value:</i> no grouping</p> <p><i>Notes:</i> This attribute should always be set to no grouping.</p>
remote-process- initialization-string	<p>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.</p>

Attribute	Description
<p><i>Allowable values:</i></p>	<p>The syntax of the remote process initialization string is as follows:</p> <p style="text-align: center;"><i>“server-name group-name update-rate deadband mode”</i></p> <p>where:</p> <p><i>server-name:</i> The name of the OPC Server.</p> <p><i>group-name:</i> The OPC group name.</p> <p><i>update-rate:</i> The rate at which to update the group's items, in milliseconds.</p> <p><i>deadband:</i> A float value from 0.0 to 100.0. If the OPC Server does not support this deadband, the default value is 0.0.</p> <p><i>mode:</i> The mode of the communication between the bridge and the OPC Server. The options are:</p> <ul style="list-style-type: none"> • S: Synchronous mode • A: Asynchronous mode <p>These parameters must be escaped with only one space character.</p> <p>For example:</p> <p style="text-align: center;">"Integ.PI.IOPC.1 group1 5000 0.5 A"</p>

Attribute	Description
<p><i>Allowable values:</i> (continued)</p>	<p>To connect G2 OPC Link to a remote OPC Server, you can specify the node name in the remote process initialization string.</p> <p>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:</p> <pre> "MachineA:IntegrationObjects.OPC.PI Group1 1000 0.0 A" or "\MachineA:IntegrationObjects.OP C.PI Group1 1000 0.0 A" or "65.108.62.138:IntegrationObjects.OP C.PI Group1 1000 0.0 A" </pre>
<i>Default value:</i>	<p>Note: The OPC server should be registered in the machine on which the bridge is running.</p> <p>none</p>
<i>Notes:</i>	<p>If the remote-process-initialization-string is empty, only remote procedures can be used for displaying the available OPC Servers.</p> <p>If the remote-process-initialization-string contains only the name of the OPC Server, the connection will not allow OPC items to be managed.</p> <p>When the <i>update-rate</i>, <i>dead-band</i>, and <i>mode</i> are not specified, the <i>mode</i> is asynchronous and the <i>update-rate</i> is provided by the OPC Server.</p> <p>If the name of the OPC server contains space characters, the name must be enclosed in hash-marks (#). For example:</p> <pre>#Integration Objects.PI.OPC.1#</pre>

Attribute	Description
<p>gsi-interface-status</p> <p><i>Allowable values:</i></p> <p>2 (OK): The connection between the G2 process and the bridge process is successful and being maintained.</p> <p>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.</p> <p>0 (Waiting): The interface is either disabled or inactive.</p> <p>-1 (Timeout): The G2 process has not heard from the bridge process within the <code>interfacetimeout-period</code> 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.</p> <p>-2 (Error): An error condition occurred. The connection between G2 and the G2 OPC Link bridge process has been interrupted.</p> <p><i>Default value:</i> none</p> <p><i>Notes:</i> 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.</p>	
<p>interval-to-poll-external-system</p> <p><i>Allowable values:</i> A time interval.</p> <p><i>Default value:</i> none</p>	

3. Connecting G2 To The Bridge

To connect G2 to an OPC Server you must first create and configure a `gsi-opcinterface` object. The G2 OPC Link 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 Link bridge process.
3. Edit the interface object's `gsi-connection-configuration` attribute to specify the settings required for the connection.

ACCESSING REAL TIME DATA OF OPC SERVERS FROM G2

1. Introduction

G2 OPC Link provides two ways to access 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.

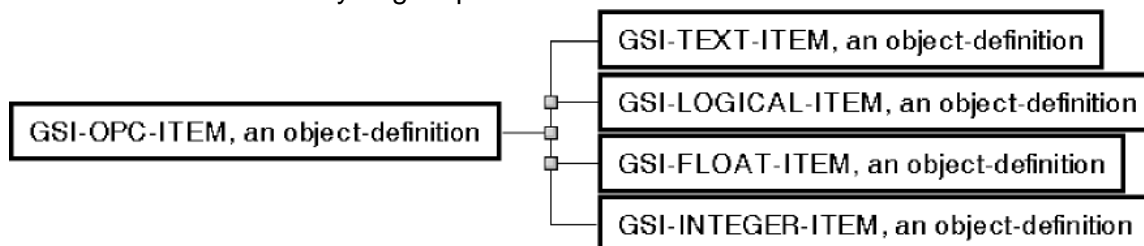
2. Accessing Data From The OPC Server

To access data located in an OPC Server, you create instances of `gsi-opc-item` variable objects in your 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 items are always integer, float, text or logical objects in the application, they will always be instances of one of the following subclasses:

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

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



3. 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-opcitem variable subclasses as the superior classes.
Class-specific-attributes	You must define the item-id and access-path 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.

4. Configuring Variables to Get Data

To configure a variable to get data, create a class definition that inherits from a gsiopc-item subclass and configure its class-specific attributes. For example, to create an OPC float item with 24P618RC.PV as the item-id 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 item-id 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	G SI-FLOAT-ITEM-XXX-2: OK
Item configuration	none
Names	24P618RC.PV
Tracing and breakpoints	default
G SI 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 fomula yet
Initial value for simulation	default
Data server	G SI data server
Default update interval	none
G SI variable status	0
Itemid	"24P618RC.PV"
Access path	""



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.

5. Reading Variables 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"
```



The external-system-has-a-scheduler attribute that 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"
```



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

6. Combining Asynchronous and Synchronous Mode

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.

7. 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 filter return values, based on the GSI variable status. For a list of GSI variable status values, see Appendix C, "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
Bad	Group 1	300, 301, 302, 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
0	Bad	No updates will occur for any bad quality.
0	Uncertain	Only GSI variables with a status value of 404 or 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

8. 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.

9. Remote Procedure Calls

G2 OPC Link provides remote procedure calls for:

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

9.1. Managing OPC Servers

rpc-opc-getserverlist

Returns the list of available OPC Servers.

Synopsis

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

Return Value	Description
<u>ids</u>	The list of program IDs of OPC Servers registered in the machine where the G2 OPC Link 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 Value	Description
<i>Current-time</i>	The current time of OPC Server in UNIX time.
<i>Start-time</i>	The time when the process started in UNIX time.
<i>Group-count</i>	The total number of groups managed by the server.
<i>Band-width</i>	The approximate percent bandwidth currently in use by server.
<i>Version</i>	The OPC Server version.
<i>Vendor</i>	Vendor information for the OPC Server.
<i>Error-code</i>	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
<i>Update-rate</i>	The value of the update rate in milliseconds
<i>Error-code</i>	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: truthvalue)
```

```
-> error-code: integer, description: text
```

Argument	Description
<i>Update-rate</i>	The new update rate value in milliseconds.
<i>Dead-band</i>	The percent change in an item value that will cause an exception report of that value to a client.
<i>Active-state</i>	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
<i>Quality-filter-level</i>	The filter level to be applied to the GSI variable status.

Return Value	Description
<i>Error-code</i>	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 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, set the filter level to 1.
- For a Bad quality, set the filter level to 0 for no updates.
- For an Uncertain quality, to update only 404 and 405, 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 rpc-opc-set-qualityfilterlevel(100)
```

9.2. 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

Argument	Description
<i>Error-code</i>	0: Success. -1: Failed to get refresh item data.

rpc-opc-writevqt

Writes the value, quality and timestamp of the specified item-id. This is functionally similar to the G2 Set action except that Quality and Timestamp may be written. It is not necessary to define a group to use this RPC. This RPC is specific to OPC Data Access 3.0 servers.

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

Example RPIS= “MyOPCServerName”

Synopsis

```
rpc-opc-writevqt (item-id : text, Input-value: value, Input-timestamp: text, Inputquality: integer)
```

-> (error-code: integer, description: Text)

Argument	Description
<i>Item-id</i>	The item ID.
<i>Input-value</i>	The desired value to be written to the OPC server.
<i>Input-timestamp</i>	TimeStamps for this item's value. The time is provided with millisecond precision. Format “YYYY/MM/DD hh:mm:ss.mmm”

	Example "2004/06/10 14:32:12.000"
<i>Input-quality</i>	The quality of this item. Possible quality values: 0: GOOD, 300: BAD 400: Uncertain.

Return Value	Description
Error-code	0 : Success. -1: Failed. 1 : Partially succeeded.
Description	A descriptive error message.

9.3. 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-opcchangebrowserposition`.

For further information, see the "OPC Common Definition and Interfaces" documentation.

rpc-opc-setbrowserfilter

Sets the filter applied to the item-id when returning the item list, using `rpc-opcgetbrowserleaves`.

Synopsis

`rpc-opc-setbrowserfilter (filter: text)`

-> error-code: integer

Argument	Description
<i>Filter</i>	The text of the filter to apply. The default value is "*" .

Return Value	Description
<i>Error-code</i>	0: Success. -1: Failed to set the value of the filter.

Example

To get all available items in the server, 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
<i>Type</i>	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
<i>Error-code</i>	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
<i>Right</i>	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
<i>Error-code</i>	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
<i>Leaves</i>	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
<i>Branch-list</i>	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: truth-value, branch-name: text)

-> error-code: integer

rpc-opc-changebrowserposition (level: integer, branch-name: text)

-> error-code: integer

Argument	Description
<i>Direction</i>	The direction to browse It is: - Up if direction is false. - Down if direction is true.
<i>Level</i>	The level by which to change the current position.
<i>Branch-name</i>	The name of the branch to move into, which would be one of the strings returned by <code>rpc-opc-getitemid</code> .

Return Value	Description
Error-code	0 : Success. -1: Failed. 1 : An argument to the function was invalid. 2 : Not enough memory.

rpc-opc-getitemid

Converts a leaf name to a fully qualified item ID.

Synopsis

`rpc-opc-getitemid (name: text)`

-> `item-id: text`

Argument	Description
<i>Name</i>	The name of a branch or leaf at the current level.

Return Value	Description
<i>Item-id</i>	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.

Item-Id = 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
<i>Item-id</i>	The name of a branch at the current level.
<i>Browse-type</i>	The subset of browse elements to return: 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.
<i>Browse-filter</i>	The text of the filter to apply. The default value is "".
Return Value	Description
<i>Items-id</i>	A list containing the available branches and leaves at current position.
<i>Error-code</i>	0 : Success. -1: Failed. 1 : Partially succeeded.
<i>Description</i>	A descriptive error 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, ContinuationPoint: text, MaxElementsReturned, 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 the 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.
Max-element-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 branches and leaves at the current position.
Continuation-points	See the description of the continuation-point argument.
More-elements	If the Server does not support a Continuation Point then the server will set MoreElements to True if there are more elements than MaxElementsReturned.
Error-code	0 : Success. -1: Failed.

	1 : Partially succeeded.
Description	A descriptive error 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
<i>Item-id</i>	The item-id for which the caller wants to read the list of properties

Return Value	Description
<i>Att-count</i>	The number of returned properties.
<i>Property-id</i>	The properties ID.
<i>Property-desc</i>	The property description.
<i>Property-type</i>	The property data type.
<i>Property-val</i>	The property value.
<i>Error-code</i>	The returned error code
<i>Error-description</i>	A descriptive error message.

9.4. Logging

The following RPCs allow 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
<i>Trace-level</i>	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
<i>Error-code</i>	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
<i>Error-code</i>	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
<i>Error-code</i>	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
<i>Display-in-g2</i>	true: Display log messages in G2. false: Stop displaying log messages in G2. The default value is false.

Return Value	Description
<i>Error-code</i>	0: Success. -1: Failed to display in G2.

g2-opc-trace-to-stdout

Displays log messages in the G2 OPC Link bridge console.

Synopsis

g2-opc-trace-to-stdout (display-in-console: truth-value)

-> error-code: integer

Argument	Description
<i>Display-in-console</i>	true: Display log messages in the console false: Stop displaying log messages in the console.
Return Value	Description
<i>Error-code</i>	0: Success. -1: Failed to display in the console.

APPENDIX A: CONFIGURATION FILE

This appendix describes the log file settings you can edit in the configuration file.

G2 OPC Link 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.

You can also change the configuration file at runtime, as described in “Configuring G2 OPC Link 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
Log Settings		
<i>LogFileName</i>	Log file name, if not specified the default value is used.	LogEvent.LOG
<i>LogFileMaxSize</i>	The maximum log file size, in bytes. Once this size is reached during runtime, the log file is overwritten.	2097152~ 2 Mo
<i>LogLevel</i>	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
<i>LogToG2</i>	TRUE: Messages appear on G2 Message Board as they are logged. FALSE: Messages are only logged to file.	FALSE
<i>ArchiveLastLog</i>	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
<i>LogToScreen</i>	TRUE: Messages appear on console screen as they are logged. FALSE: Messages are only logged to file.	FALSE
Quality		

<i>QualityFilter</i>	The filter that will be applied to the GSI variable's status value.	111
Server Status		
<i>CheckStatus</i>	Enables/disables OPC server status check.	TRUE
<i>UseG2TimeStamp</i>	FALSE: the bridge uses the timestamp provided by the OPC Server TRUE: The bridge ignores the OPC Server timestamp and uses G2 timestamp	FALSE
<i>CheckFrequency</i>	The frequency at which the bridge checks the OPC server status in seconds	5
<i>InitSecurity</i>	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
<i>AsyncMode</i>	Specify the OPC write method to be used when setting new values to GSI OPC ITEMS: 0: OPC synchronous method 1: OPC asynchronous method version 1.0 2: OPC asynchronous method version 2.0	2

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
AsyncMode=2
```

APPENDIX B: CONFIGURING GSI VARIABLES IN ASYNCHRONOUS MODE

This appendix describes how to configure gsi-variables in asynchronous mode. 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.



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

Here is the utility workspace:

The screenshot shows a window titled 'UTILITY' with a sub-header 'UTILITY'. It contains two sections:

- NETWORK-REGISTRATION**: A yellow box at the top states: "NETWORK-REGISTRATION procedure is called after start action in order to register the GSI variable". Below it, a box with a list icon contains two rules:
 - for any gsi-opo-interface gs whenever the gsi-interface-status of gs receives a value and when the gsi-interface-status of gs = 2 then start network-registration (gs)
 - whenever any gsi-opo-interface gs is activated and when the gsi-interface-status of gs = 2 then start network-registration (gs)
- ITEM-REGISTRATION**: A yellow box states: "ITEM-REGISTRATION procedure register a gsi variable whenever this variable is activated". Below it, a box with a list icon contains one rule:
 - whenever any gsi-opo-item it is activated then start item-registration (it)

Here is the network-registration procedure:

NETWORK-REGISTRATION, a procedure	
Notes	OK
Authors	none
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;
end
    
```

Here is the item-registration procedure:

ITEM-REGISTRATION, a procedure	
Notes	OK
Authors	none
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(l: class item )
handle: integer;
name-int: symbol;

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

      name-int = the gsi-interface-name of l;
      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 (l,
        gs);
      end;
    end;
end
  
```

APPENDIX C: GSI-OPC-ITEM STATUS VALUES

This appendix describes status values for gsi-opc-item when a connection takes place and errors that can occur. The following tables describe the possible values for the gsi-status attribute of the GSI variable when a connection takes place between G2 and the bridge process.

1. 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.
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.
203	OPC_E_BADTYPE	The item data type is bad.
206	OPC_ADD_ITEM_FAIL	Unable to add item, no specific reason is known.

2. 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.
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.
333	OPC_BAD_INVALID_QUALITY	Receive an invalid bad quality "Not used by OPC"

3. 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'.
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	Received an invalid uncertain quality "Not used by OPC"

4. 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.

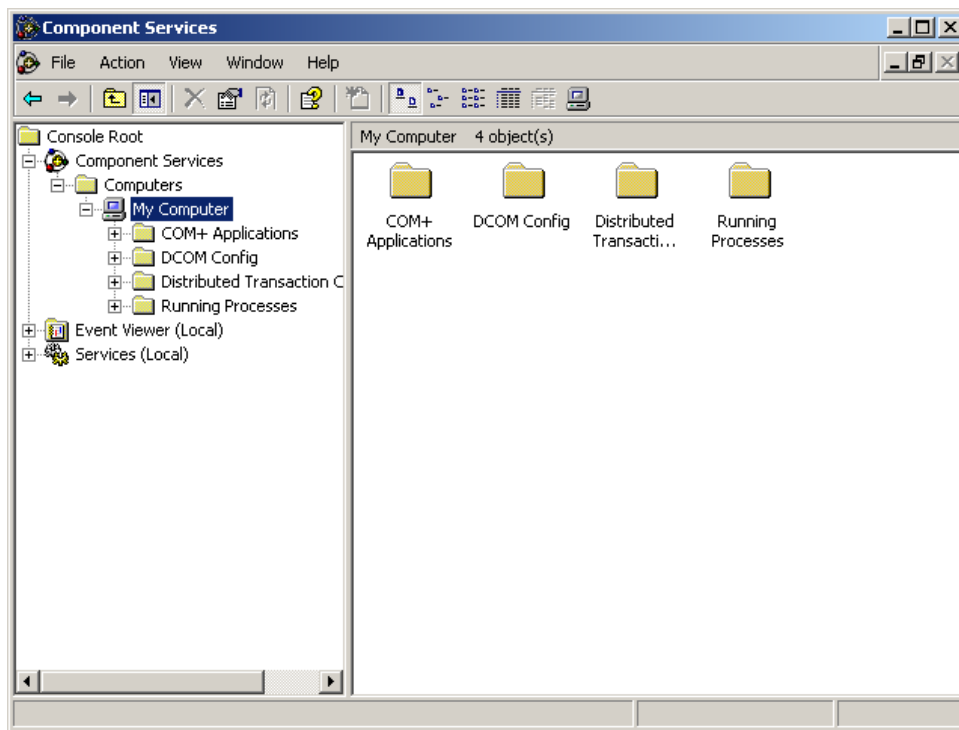
APPENDIX D: DCOM CONFIGURATION FOR WINDOWS XP

This section describes how to configure the client for DCOM on Windows XP.

1. Setting Up Client DCOM Configuration

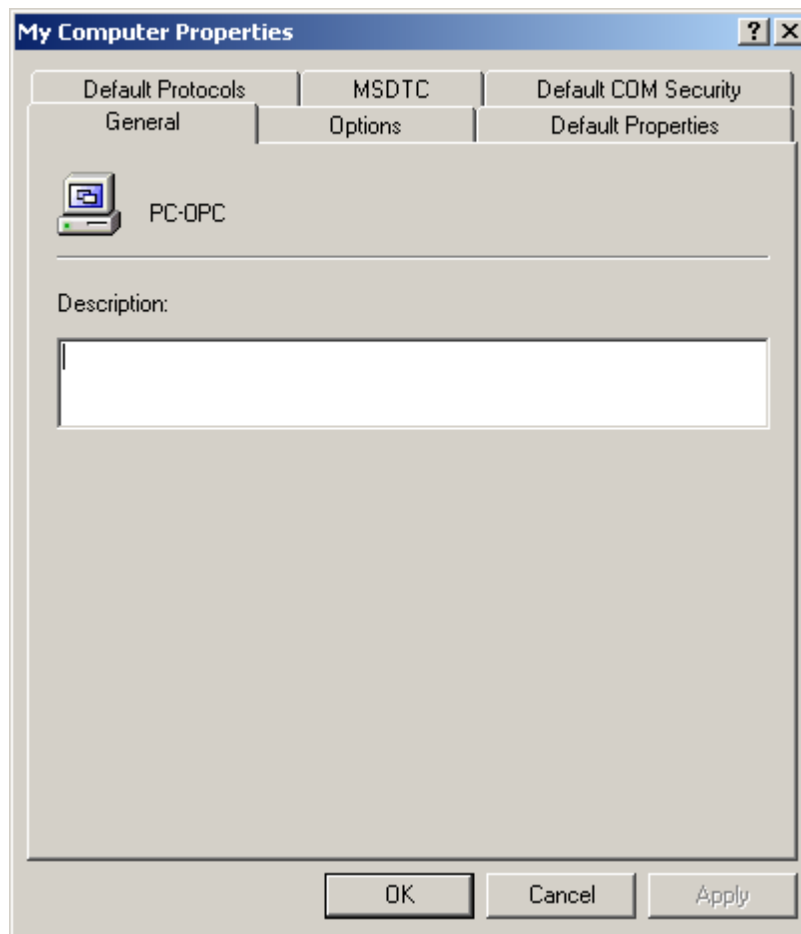
To set up client:

1. Login as Administrator.
2. Choose Component Services from Control Panel > Administrative Tools.



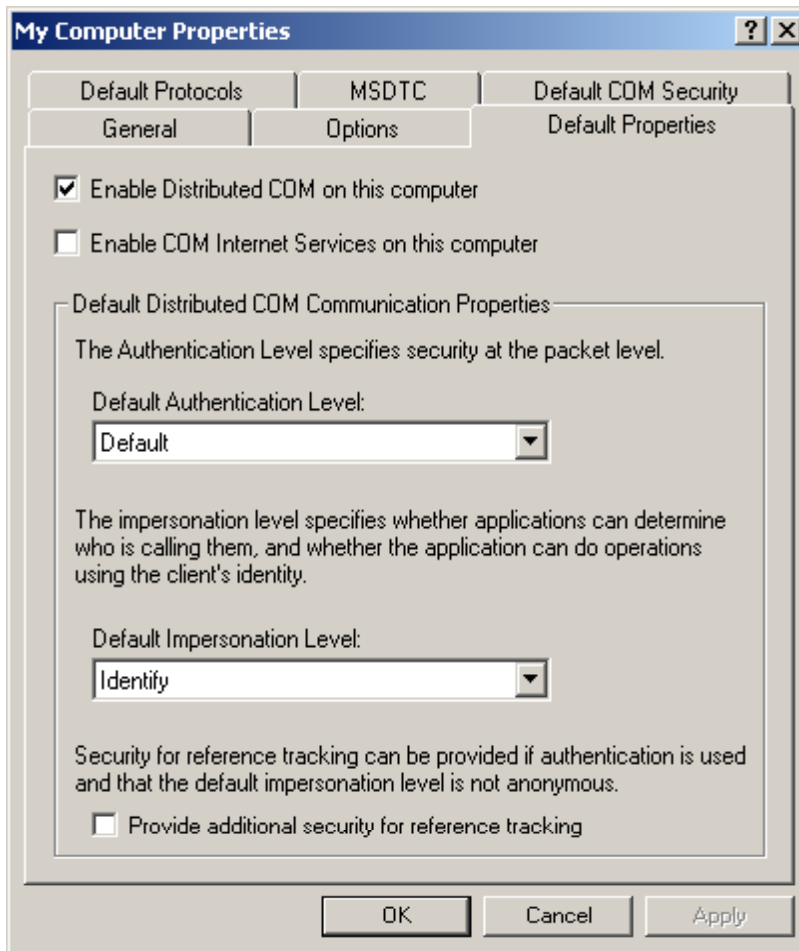
3. From Action menu, select Computer Properties. This section is similar to configuring com with Windows NT.

The Computer Properties configuration utility looks like this:

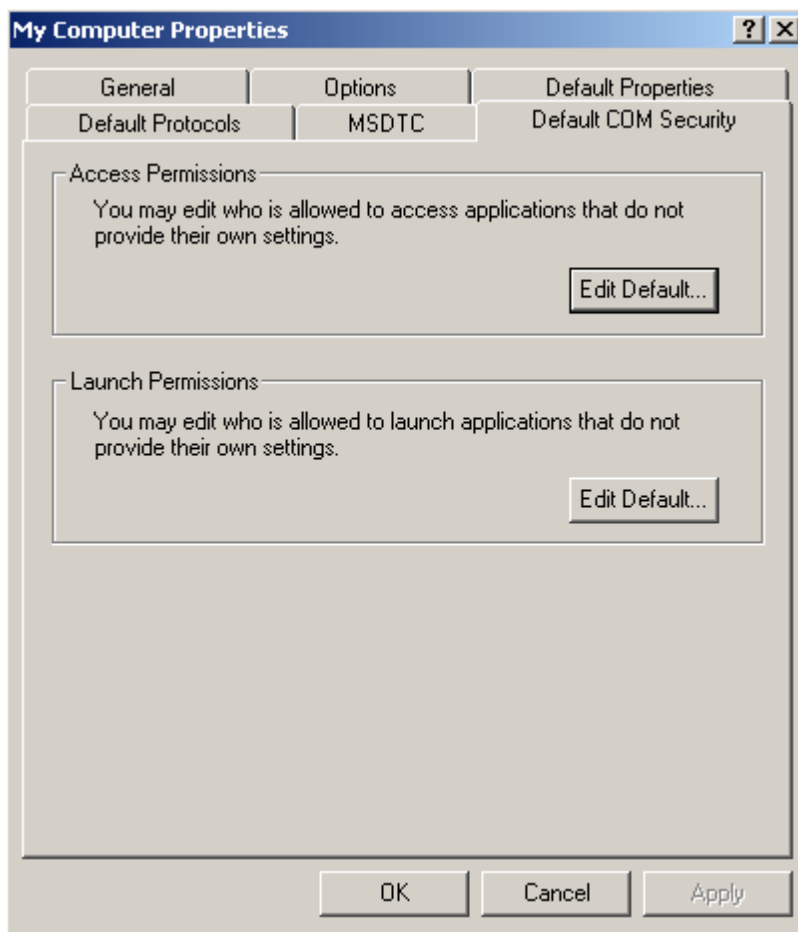


4. Click the **Default Properties** tab and configure the dialog, as follows:
 - a. Ensure that the Enable Distributed COM on this computer is enabled.
 - b. Configure the Default Authentication Level to be Connect.
 - c. Configure the Default Impersonation Level to be Identity.

The Default Properties tab should look like this:

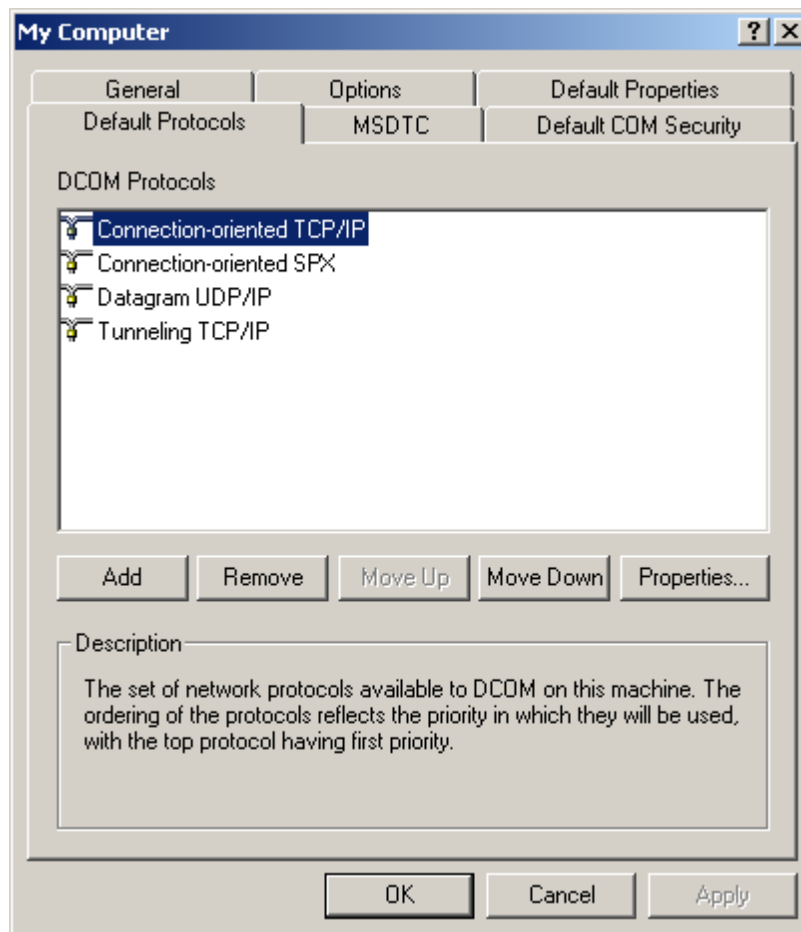


5. Click the Default Security tab:



This is where you configure who has access to G2 OPC Link 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 G2 OPC Link performs subscription-based read operations.
7. Click the **Default Protocols** tab:



This is where you set the network protocols on the client computer to use DCOM.

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

2. Registering OPC Servers

You must now register the OPC Server(s) that G2 OPC Link 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:

1. 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.

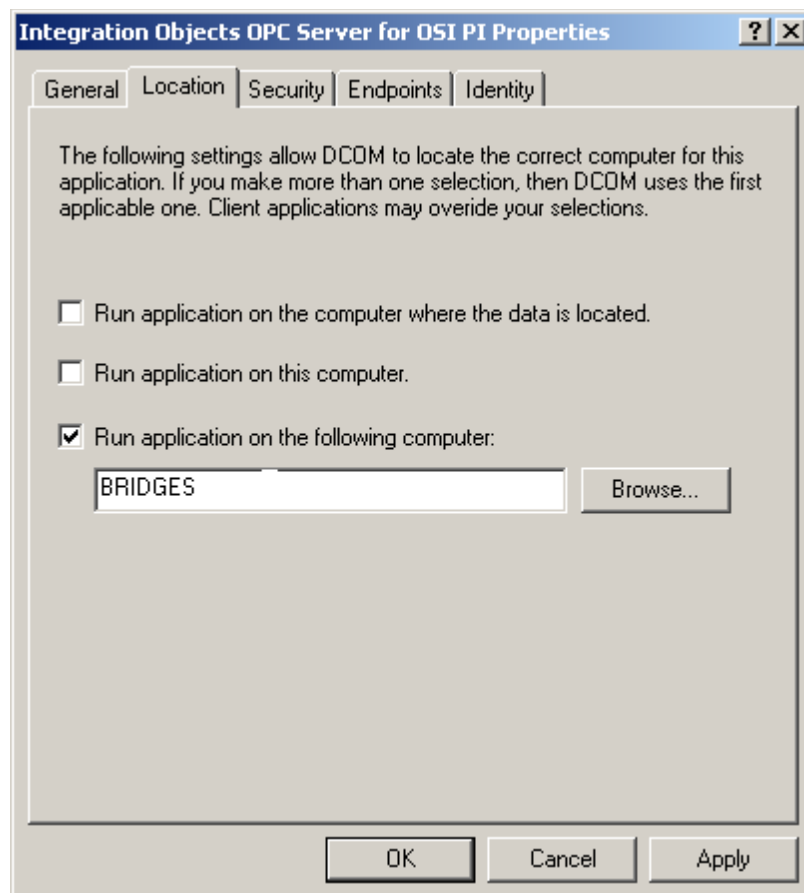


In either case, the desired registry information for the OPC Server must not include the LocalServer32 or InProcServer32 keys, and must include the AppID key and the associated RemoteServerName.

If the server application name does not show up in the DCOM config 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 remotely access "Integration Objects OPC for PI," which is an OPC server for PI, enable the Run application on the following computer option and enter the NetBIOS name of the computer on which to run the application, as shown below:



3. Configuring The Server For DCOM On Windows XP

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

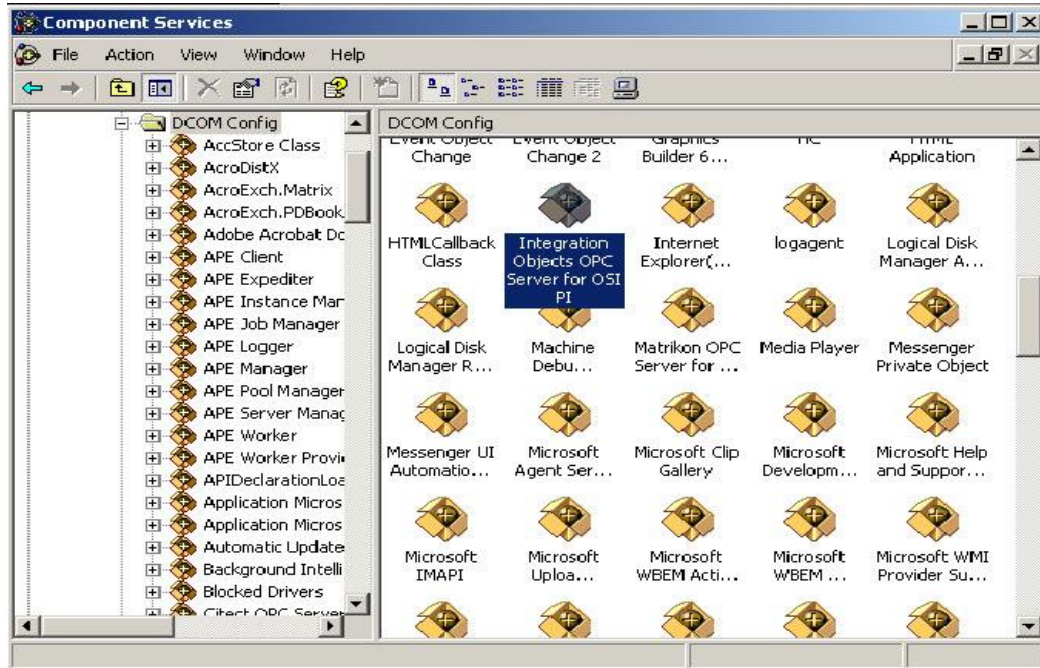
To set up the server:

1. Launch the Component Services Utility on the computer where your target OPC Server is running.
2. Configure the Default Properties tab as you did on the client side.
3. Click the **Default Security** tab.
4. This is where you tell the operating system who you will allow to access OPC
5. 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).
6. Click the **Default Access Permissions** tab.
7. 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 you grant permissions.
8. Click the **Default Security** tab and click Edit Defaults under Default Launch Permissions.
9. On this tab, you define who can actually start your OPC Server on this computer.
10. You add users/groups the same way you did when configuring Access Permissions.
11. On the **Default Security** tab, click **Edit Defaults** under Default Configuration Permissions.
12. If you are setting up DCOM for the first time, we recommend that you do not change these settings.
13. Configure the Default Protocols tab as you did on the client side.

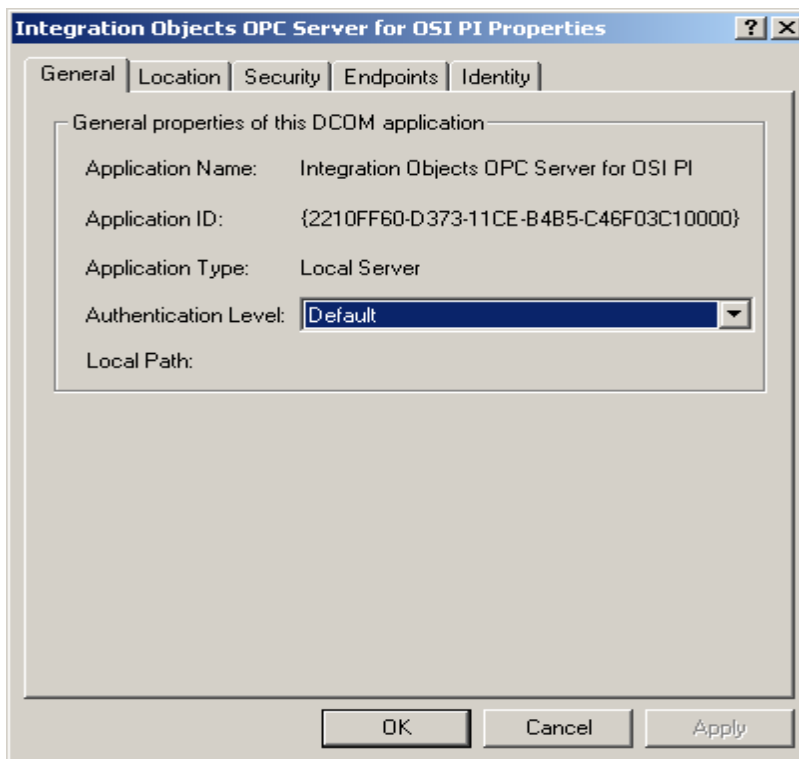
4. Configuring DCOM Settings For Your OPC Server

To configure DCOM settings for your OPC Server:

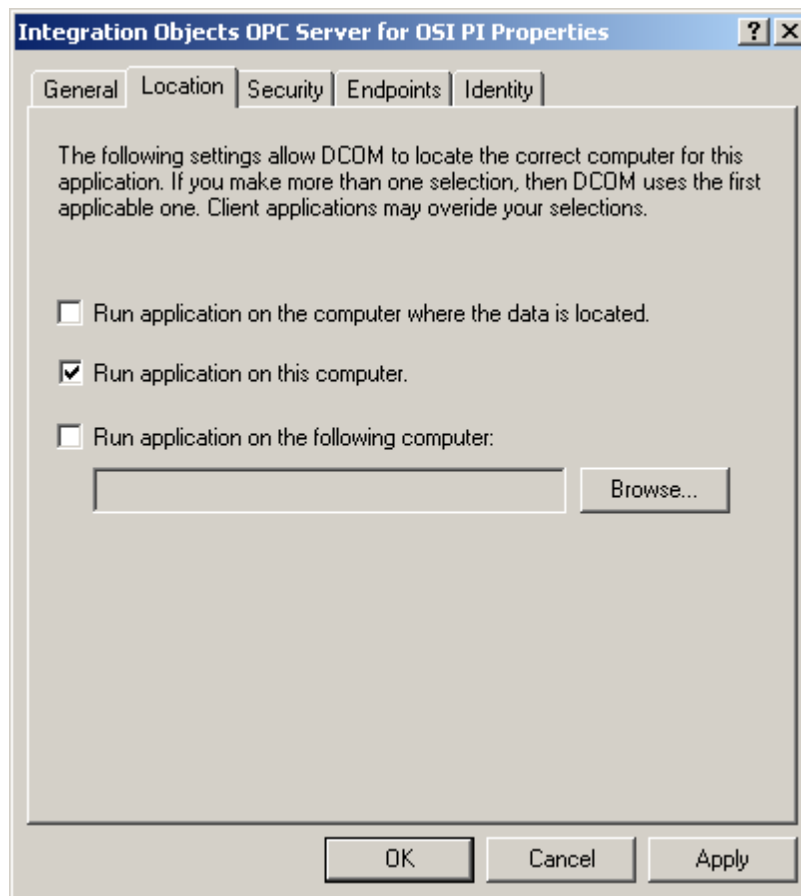
1. Click the **Application** tab in DCOM Configuration dialog and browse until you find the OPC Server of your choice.



2. Select the server and either double-click it or choose Properties to configure server-specific settings, as follows:
 - a. On the General tab, we recommend that you leave the Authentication Level as Default:

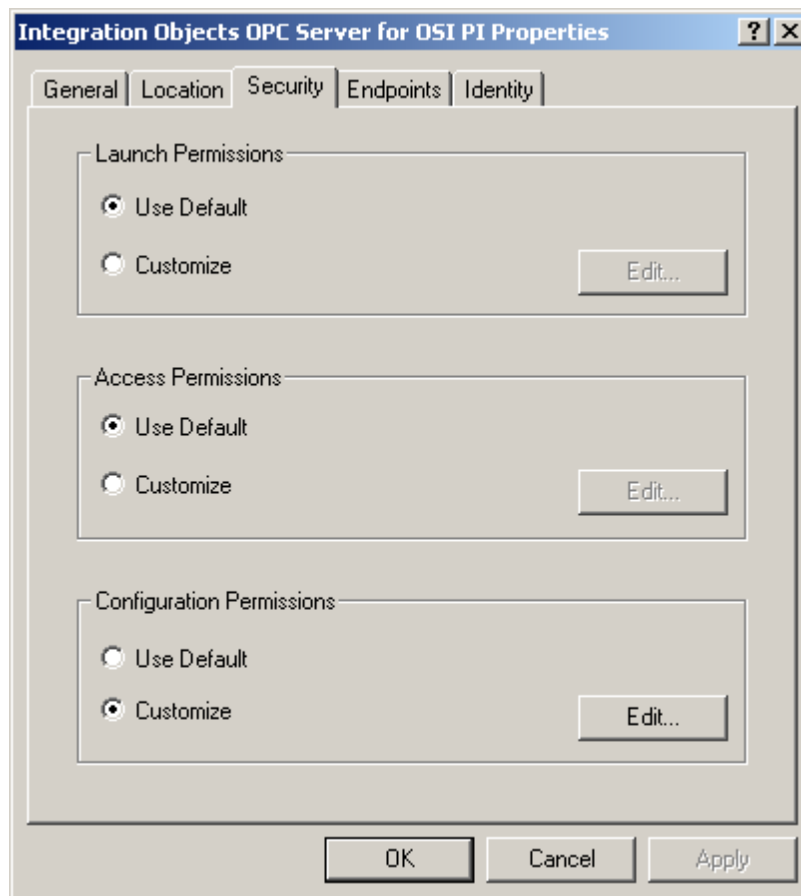


- b. On the Location Tab, ensure that Run application on this computer is the only check box that is checked:



- c. On the Security Tab, we recommend that you:
- i. Enable Use default access permissions, which means that users/groups shown under Default Security tab in the DCOM configuration utility will have access to connect to this OPC Server.
 - ii. Enable Use default launch permissions.
 - iii. The same rules apply to using custom launch permissions and 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:



3. On the Identity Tab, 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.

For additional information on this guide, questions or problems to report, please contact:

Offices

- Houston, USA: +1 713 609 9208
- Genova, Italy: +39 34 75 83 93 47
- Tunis, Tunisia: +216 71 861 803

Email

- Support Services: customerservice@integrationobjects.com
- Sales: sales@integrationobjects.com

To find out how you can benefit from other Integration Objects products and custom-designed solutions, please visit us on the Internet:

Online

- www.integrationobjects.com