

**Siemens
Energy & Automation, Inc.**

SOFTWARE GUIDE

SG390PCDS-1
Rev: 3
October 2005

APACS+TM OPC Device Server Version 1.10 or Higher

TABLE OF CONTENTS

<u>SECTION AND TITLE</u>	<u>PAGE</u>
1.0 Introduction	1-1
1.1 Product Description	1-1
1.2 Product Support	1-3
1.3 Related Literature	1-5
2.0 OPC Device Server Architectures	2-1
2.1 Local OPC Device Server Architectures	2-1
2.1.1 Local OPC Device Server and OPC Client Using Ethernet	2-1
2.1.2 Local OPC Device Server and OPC Client Using MODULBUS/MODULNET	2-1
2.2 Distributed OPC Device Server Architectures	2-4
2.2.1 Distributed OPC Device Server and OPC Client on Dual Workstations	2-4
2.2.2 Distributed OPC Device Server and OPC Client on Shared Tag Server	2-4
2.3 OPC Compliance	2-7
3.0 Installation	3-1
3.1 Platform Requirements	3-1
3.2 Installation Considerations	3-1
3.2.1 Software Licensing	3-2
3.2.2 M-BUS/M-NET Drivers	3-3
4.0 Server Startup and Operation	4-1
4.1 Operation	4-2
4.2 Troubleshooting	4-4
5.0 Device Server Settings	5-1
5.1 APACS+ Control Panel Applet	5-1
5.2 Clients	5-3
5.3 Options	5-3
5.3.1 Logging	5-4
5.3.2 Runtime	5-5
5.3.3 Startup Mode	5-5
5.3.4 Settings	5-6
5.4 Load Browse DB	5-7
5.5 Report	5-8
5.6 Server Tags	5-9
5.7 Help	5-16
5.8 View Log	5-16
5.9 Distributed OPC Device Server to OPC Client Communications (DCOM)	5-17
5.9.1 DCOM Set-Up for OPC Device Server	5-17
5.9.2 Windows XP and 2003 Server DCOM Set-Up.....	5-17
5.9.3 Windows XP and 2003 Server Software Firewall Set-Up.....	5-24
5.9.4 Windows NT and 2000 DCOM Set-Up.....	5-29

6.0 Client Browse Function Settings	6-1
--	-----

Appendices

Appendix A Error Codes	A-1
Appendix B Data Types	B-1
Appendix C System Tags	C-1
Appendix D Data Quality	D-1
Appendix E Program Boundaries and Considerations	E-1

LIST OF FIGURES

<u>FIGURE AND TITLE</u>	<u>PAGE</u>
1-1 Simplified APACS+ System with OPC Device Server	1-2
1-2 Icon for APACS+ Control Panel Applet	1-2
2-1 Local OPC Device Server / Client Using Ethernet	2-2
2-2 Local OPC Device Server / Client Using M-BUS/M-NET	2-3
2-3 Distributed OPC Device Server / Client Using DCOM Between Two Stations	2-5
2-4 Distributed OPC Device Server / Client Using DCOM and Shared Tag Server	2-6
4-1 Icon for APACS+ Control Panel Applet	4-1
4-2 OPC Device Server Control Panel Display	4-1
4-3 Configuration Options Display	4-2
4-4 Example Browse Display	4-3
4-4 Example Log Files	4-5
5-1 Icon for APACS+ Control Panel Applet	5-2
5-2 OPC Device Server Control Panel Display	5-2
5-3 OPC Server Icon in System Tray	5-2
5-4 Connected Clients Dialog Box	5-3
5-5 Configuration Options Dialog Box	5-4
5-6 Save Configuration Dialog Box	5-6
5-7 Load Configuration Dialog Box	5-6
5-8 APACS+ Browse DB Tag Loader Dialog Box	5-8
5-9 Save TIC Report Dialog Box	5-9
5-10 Device Server Tags Dialog Box	5-10
5-11 Communications Settings Dialog Box	5-11
5-12 DCOM Configuration, Applications Tab	5-29
5-13 DCOM Configuration, Default Properties Tab	5-30
5-14 DCOM Configuration, Defaults Security Tab	5-30
5-15 DCOM Configuration, Default Protocols Tab	5-31
5-16 OPC Device Server Properties, General Tab	5-31
5-17 OPC Device Server Properties, Location Tab	5-32

5-18	OPC Device Server Properties, Security Tab	5-32
5-19	OPC Device Server Properties, Identity Tab	5-33
5-20	OPC Device Server Properties, Endpoints Tab	5-33

LIST OF FIGURES (Continued)

<u>FIGURE AND TITLE</u>	<u>PAGE</u>
6-1 Example Client Browse Function	6-1
6-2 Navigating the Browse Database	6-2
6-3 Example of Browse Filtering	6-2

LIST OF TABLES

<u>TABLE AND TITLE</u>	<u>PAGE</u>
1-1 TIC Contact Information	1-4
A-1 Runtime Error Codes	A-1
A-2 Extended Error Codes	A-4
B-1 APACS+ to OPC Data Types	B-1
B-2 OPC Canonical Types to OPC Requested Types	B-2
C-1 Basic System Tags	C-1
C-2 APACS+ System Tags	C-2
C-3 APACS+ System Tags Per Controller	C-9
C-4 OPC System Tags	C-12
D-1 Data Quality	D-1

SIGNIFICANT CHANGES FOR REVISION 3

SECTION AND DESCRIPTION

- 1.1 Product Support** - This section has been updated to reflect the new Technical Support Center information.
- 5.0 Device Server Settings** - This section has been updated with new text and graphics to detail the new DCOM and Firewall setup for Windows XP and 2003 Server.

Siemens Energy & Automation, Inc. assumes no liability for errors or omissions in this document or for the application and use of information included in this document. The information herein is subject to change without notice.

*APACS+ and ProcessSuite are trademarks of Siemens Energy & Automation, Inc.
All other trademarks are the property of the respective owners.*

© Copyright 2005 Siemens Energy & Automation, Inc. All rights reserved.

1.0 Introduction

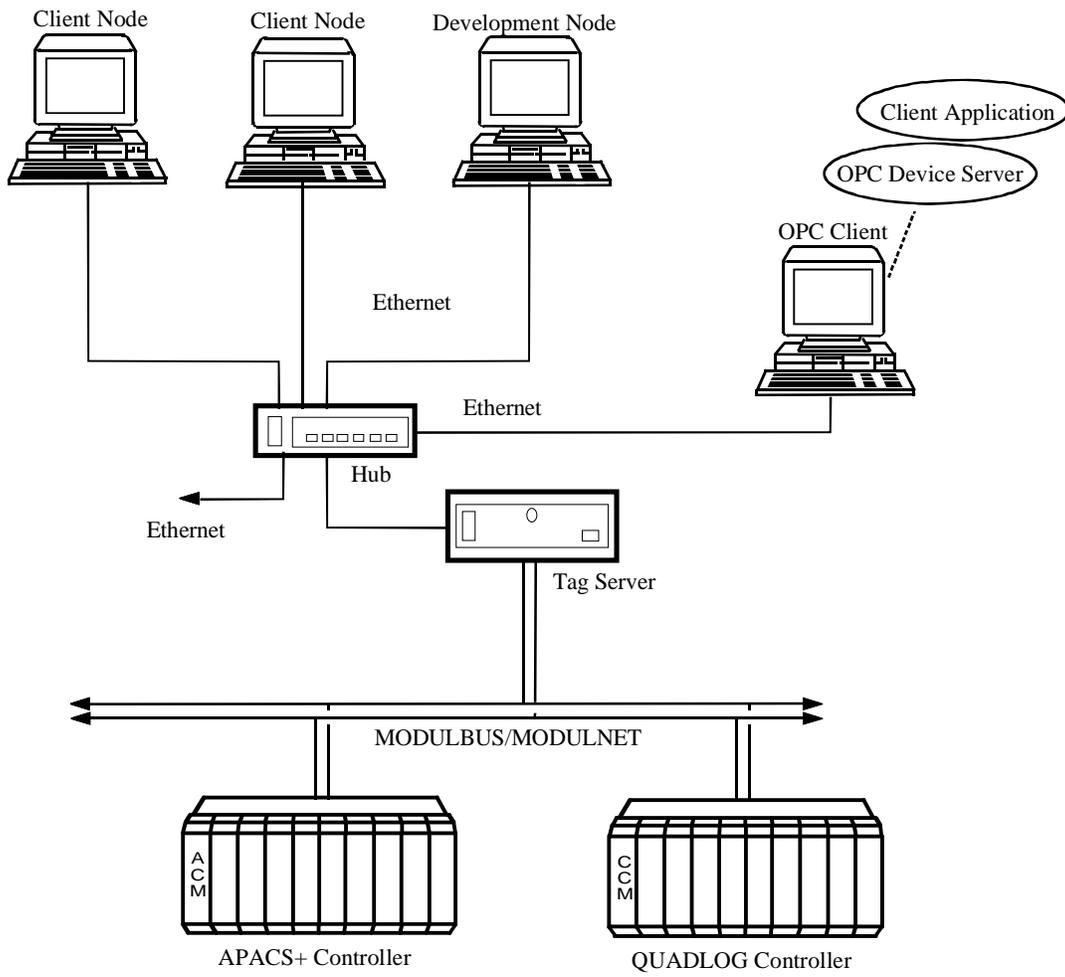
This Guide provides installation, setup, and start-up information for the APACS+™ OPC Device Server. The OPC Device Server provides an OPC-compliant application with access to an APACS+ system's controller(s). Examples of third party applications would be clients that handle automatic loop tuning or perform process analysis.

This document is divided into the following major sections:

- Section 1, Introduction - Summarizes the information presented in this document. In addition, it presents a description of the product, provides information about obtaining product support and supplies a list of related literature.
- Section 2, OPC Device Server Architecture – Details the product's architecture for Ethernet and MODULBUS implementations, and provides information regarding OPC compliance.
- Section 3, Installation - Describes installation of the product. This includes general hardware requirements, prerequisite software, licensing requirements, and OPC Device Server software installation.
- Section 4, Server Startup and Operation - Describes the proper sequence and procedures in which to start the server in relation to front-end clients and back-end process I/O systems.
- Section 5, Device Server Settings – Provides information about the OPC Device Server's initial setup for Control Panel Applet, clients, options, browse, server tags, and reports.
- Section 6, Client Browse Function Settings – Furnishes information about initial Browse function setup using an example browse client.

1.1 Product Description

The OPC Device Server software provides a third party OPC-compliant application with read and write access to an APACS+ Process Automation System's controllers as illustrated in Figure 1-1. This includes APACS+ Advanced Controllers and/or QUADLOG® Safety Controllers. Since OPC has become a widely used process industry standard, it extends the flexibility of an APACS+ system by providing an easier path for accommodating third party applications. Examples of applications would be clients that perform automatic loop tuning or perform process analysis. The physical connection from the device server to a controller can be implemented two basic ways. One way is to use a system's Ethernet network. Another way is to connect to a system's MODULBUS/MODULNET. The OPC Device Server requires minimal setup after installation. The OPC Device Server runs as a Windows™ NT service and user interface access to it is provided via the APACS+ Control Panel Applet. The icon for starting this applet is shown in Figure 1-2. The applet provides a graphical interface to facilitate server setup, tag browsing, monitoring, and status reports. These functions are covered in this document. The OPC Device Server and associated third party OPC client application(s) can be installed on the same workstation or can be installed on separate workstations.



SG13631.DS4

Figure 1-1 Simplified APACS+ System with OPC Device Server

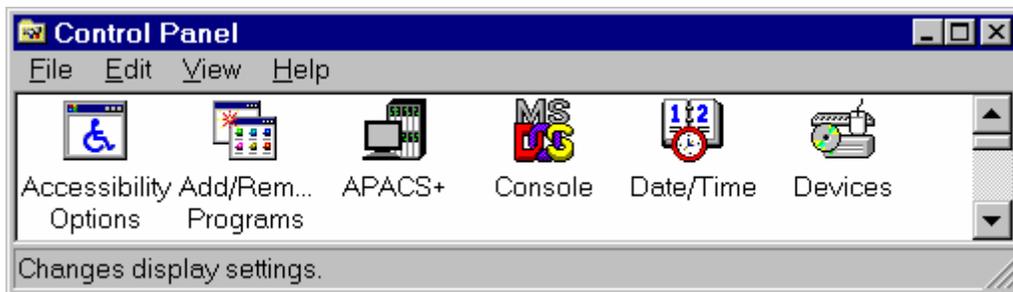


Figure 1-2 Icon for APACS+ Control Panel Applet

1.2 Product Support

Our Technical Support Centers (TSC) offer a variety of technical support services that are designed to assist you with Siemens products and systems. Our support engineers have experience with troubleshooting, development, system startup, and system test. They will help you to solve your issues in an efficient and professional manner.

Customers in North America can contact Siemens Technical Support Center at 1-800-333-7421, on the web at: <http://support.automation.siemens.com>, or by e-mail: techsupport.sea@siemens.com

Customers outside North America can contact their local Siemens subsidiary; addresses and telephone numbers are listed on the Internet at the web site: <http://support.automation.siemens.com>.

When contacting Siemens, customers will be asked to provide site-contact information (name, address, and phone number), the product involved and detailed information regarding the nature of the issue.

Product documentation is now located in the Library forum of the Process Automation User Connection at: <http://sitiescape.sea.siemens.com/>. The Process Automation User Connection is a secure site. Registration is open to all verified users of Siemens process automation systems. If you are not already, and would like to become a member, please visit our Process Automation User Connection web page at: <http://www.sea.siemens.com/process/support/papauc.html>

Contained within the Process Automation User Connection is the APACS+/QUADLOG Secure Site at: <http://sitiescape.sea.siemens.com/forum/aca-1/dispatch.cgi/f.apacsquadlo> forum. This site is only open to customers with an active service agreement. It contains all service manuals, service memos, service notes, configuration manuals, etc. for the APACS+ and QUADLOG family of products. If you are experiencing technical difficulties with the site, please contact SiteScape technical support at: toll free 1-877-234-1122 (US) or 1-513-336-1474.

A&D Technical Support

Worldwide, available 24 hours a day:



United States: Johnson City, TN	Worldwide: Nürnberg	Asia / Australia: Beijing
<p>Technical Support and Authorization Local time: Monday to Friday 8:00 AM to 5:00 PM Telephone:+1 (423) 262 2522 or +1 (800) 333-7421 (USA only) Fax:+1 (423) 262 2289 Mail to: techsupport.sea@siemens.com GMT: -5:00</p>	<p>Technical Support 24 hours a day, 365 days a year Phone:+49 (180) 5050-222 Fax:+49 (180) 5050-223 E-Mail: ad.support@siemens.com GMT:+1:00</p> <hr/> <p>Authorization Local time: Monday to Friday 8:00 AM to 5:00 PM Phone: +49 (180) 5050-222 Fax: +49 (180) 5050-223 Mail to: ad.support@siemens.com GMT: +1:00</p>	<p>Technical Support and Authorization Local time: Monday to Friday 8:00 AM to 5:00 PM Phone:+86 10 64 75 75 75 Fax:+86 10 64 74 74 74 Mail to:ad.support.asia@siemens.com GMT:+8:00</p>
<p>Automation and Drives Service and Support International http://www.siemens.com/automation/service&support</p>		
<p>The languages of the SIMATIC Hotlines and the authorization hotline are generally German and English.</p>		

1.3 Related Literature

The following documents are available to complement the installation, configuration, and maintenance of the OPC Device Server. Generally, all needed documentation is supplied with your system. It may be supplied in electronic form in the Adobe™ portable document format (PDF), printed form, or both. Refer to these documents as needed or as called for in text. Documents supplied by Siemens in electronic form are provided on a CD supplied with your system. Pre-printed copies may also be purchased.

- *APACS+ MBI/MNI Interface Card & M-BUS/M-NET Driver Software Installation and Service Instruction* (document number SD39MBIMNI-2)
- *APACS+ Database Automation* (document number CG39DA-1)
- *NIM32 (Network Interface Manager) Software* (document number SG39-8)
- *OPC Client Application Documentation*
- *Rack-mounted Network Interface (RNI) Installation and Service Instruction* (document number SD39RNI-2)
- *Rack-mounted Industrial Server (RIS) Installation and Service Instruction* (document number SD39RIS-1)
- *Runtime Application Programming Interface (API) Version 4.00 or Higher* (document number SG39RTAPI-2)
- *Windows NT Documentation*

#

2.0 OPC Device Server Architectures

This section outlines the prominent architectures of the OPC Device Server. Some architectures feature the OPC Device Server and one or more OPC client applications residing within the same workstation. Other architectures feature distributed layouts, based on DCOM (Distributed Common Object Model) technology, whereas the OPC Device Server and one or more OPC client applications reside on different workstations. Since there are many ways to layout a distributed architecture, this document features those supported by Siemens.

2.1 Local OPC Device Server Architectures

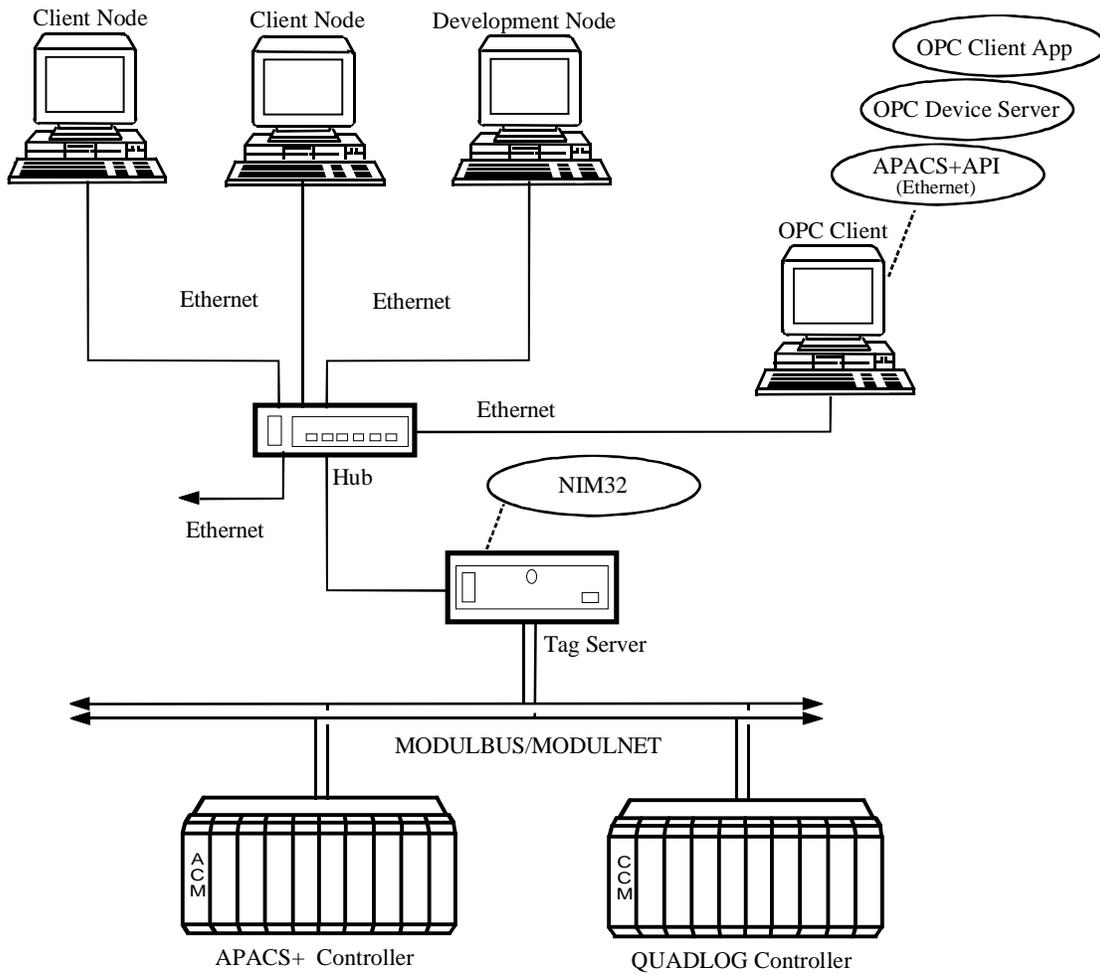
2.1.1 Local OPC Device Server and OPC Client Using Ethernet

This section presents a local OPC Device Server and an OPC client application residing within the same workstation and using Ethernet communications to access system or controller information. An example is shown in Figure 2-1. The example shows the OPC client station hosting the OPC client application, the OPC Device Server, and the APACS+ API (Ethernet version) software. A tag server hosts several software elements and performs multiple functions. The OPC Device Server communicates through the Network Interface Manager 32-bit (NIM32) software in the System Tag Server to reach the MODULBUS/MODULNET network.

The NIM32 software functions as an Ethernet-to-MODULBUS/MODULNET interface. For earlier software platforms, a hardware-based NIM could be used instead. The architecture shown in Figure 2-1 has been simplified for clarity. In a practical implementation, redundant tag servers are commonly used.

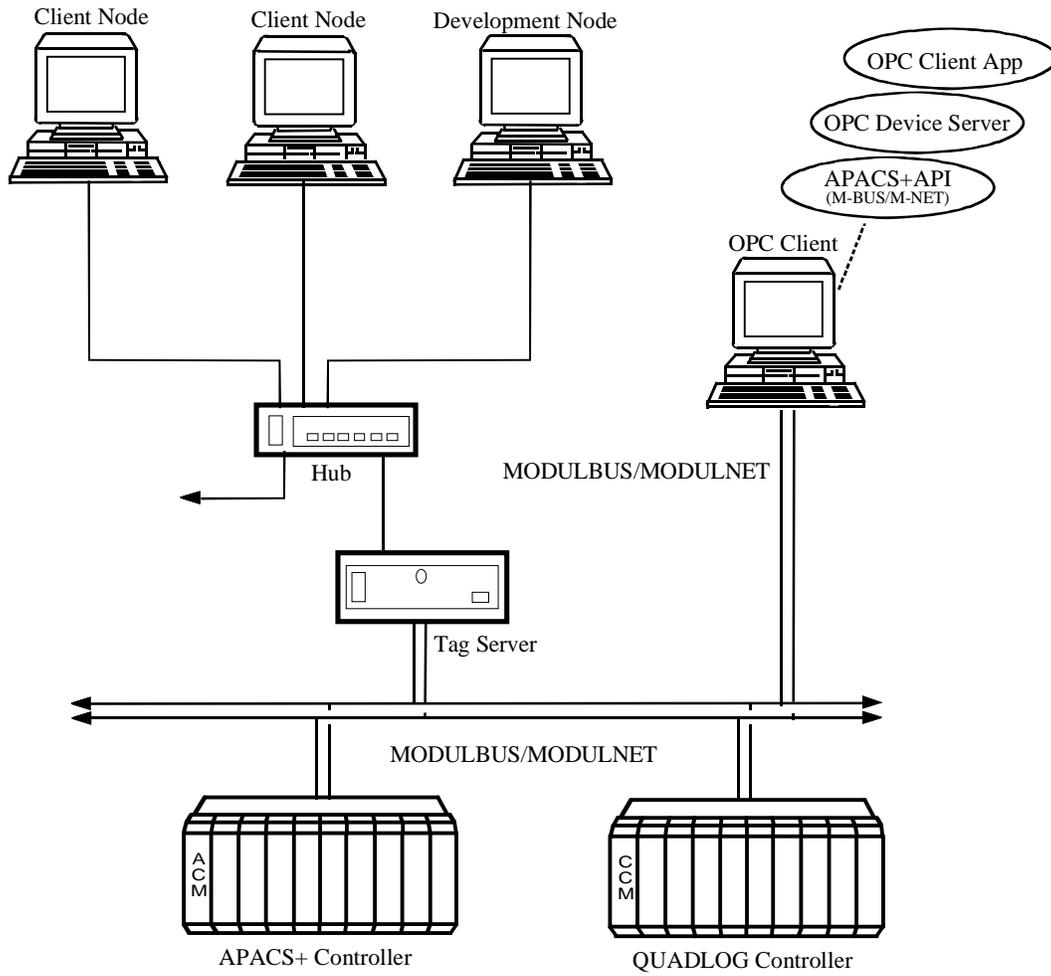
2.1.2 Local OPC Device Server and OPC Client Using MODULBUS/MODULNET

This section shows a local OPC Device Server and an OPC client application residing within the same workstation and using MODULBUS (M-BUS)/MODULNET (M-NET) communications to access system or controller information. An example is shown in Figure 2-2. The example shows the OPC client station hosting the OPC client application, the OPC Device Server, and the APACS+ API (M-BUS version). In this scheme, the OPC client station connects directly to the system's M-BUS/M-NET. The architecture shown here has been simplified for clarity. In a practical implementation, some redundant system elements are commonly used.



SG13641.DS4

Figure 2-1 Local OPC Device Server / Client Using Ethernet



SG13651.DS4

Figure 2-2 Local OPC Device Server/Client Using M-BUS/M-NET

2.2 Distributed OPC Device Server Architectures

2.2.1 Distributed OPC Device Server and OPC Client on Dual Workstations

This section highlights a Siemens-supported distributed architecture where an OPC Device Server and an OPC client application reside on separate workstations and communicate with each other via DCOM technology to access system or controller information. An example is shown in Figure 2-3. The example shows the OPC Device Server on one workstation and the OPC client application on another. A tag server hosts several software elements and performs multiple functions. The OPC Device Server communicates through the Network Interface Manager 32-bit (NIM32) software in the System Tag Server to reach the MODULBUS/MODULNET network.

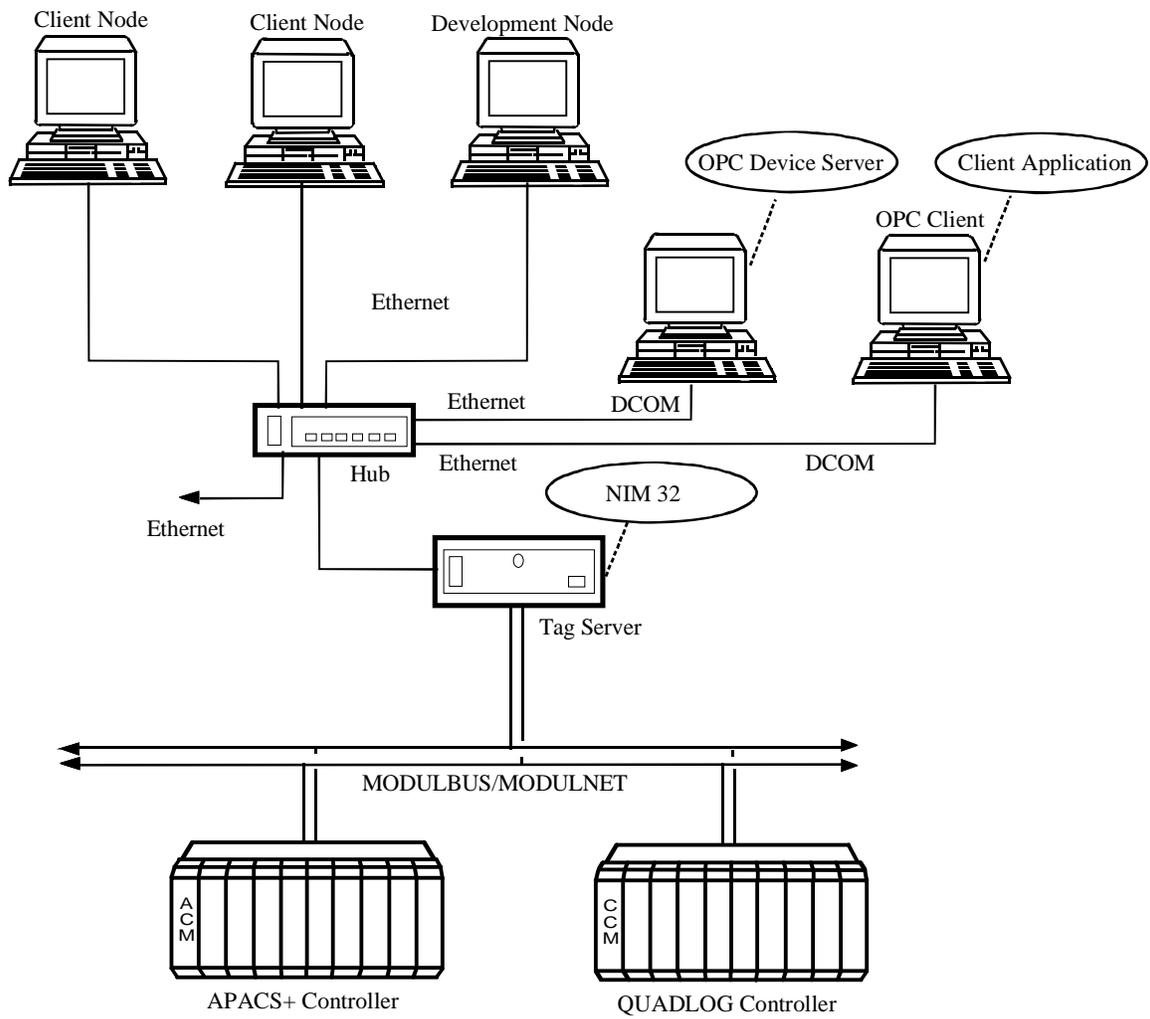
The NIM32 software functions as an Ethernet-to-MODULBUS/MODULNET interface. For earlier software platforms, a hardware-based NIM could be used instead. The architecture shown in Figure 2-3 has been simplified for clarity. In a practical implementation, dual tag servers are commonly used.

2.2.2 Distributed OPC Device Server and OPC Client on Shared Tag Server

This section highlights a Siemens-supported distributed architecture where an OPC Device Server resides in one system element and an OPC client application resides on a separate workstation. These OPC elements communicate with each other via DCOM technology to pass system or controller information. An example is shown in Figure 2-4. The example shows the OPC Device Server residing within a tag server and the OPC client application within a separate workstation. A tag server hosts several software elements and performs multiple functions. The scheme shown in Figure 2-4 has been simplified for clarity. In a practical implementation, redundant tag servers are commonly used.

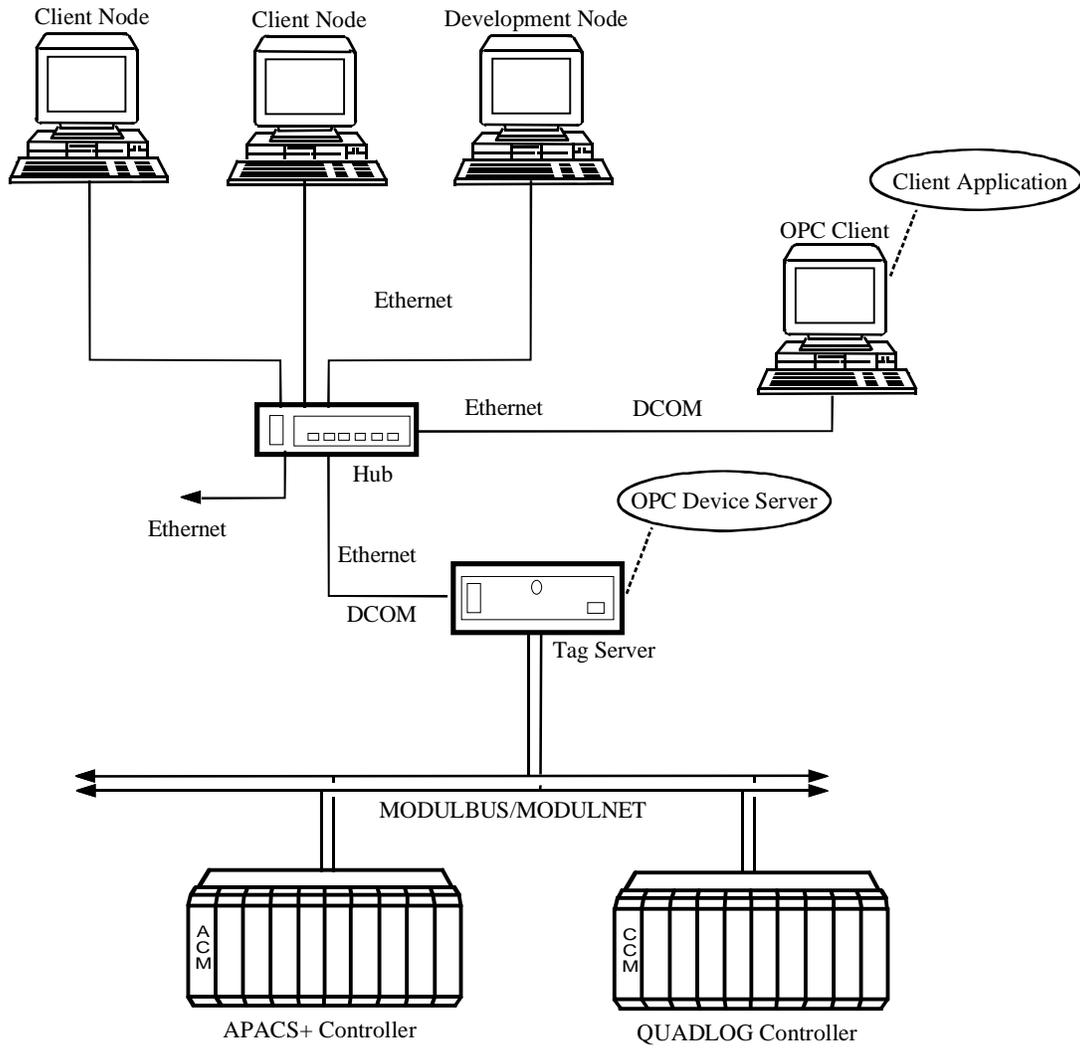
NOTE

Loading the OPC Device Server on a tag server does not increase availability (i.e., the OPC Device Server does not have a built-in redundancy model).



SG13681.DS4

Figure 2-3 Distributed OPC Device Server/Client Using DCOM Between Two Stations



SG13671.DS4

Figure 2-4 Distributed OPC Device Server/Client Using DCOM and Shared Tag Server

2.3 OPC Compliance

The OPC Device Server is compliant with the following OPC specifications:

- *OPC Common Specification 1.0*, OPC Foundation
- *OPC Data Access Custom Specification 2.04*, OPC Foundation
- *OPC Data Access Automation Specification 2.02*, OPC Foundation

#

3.0 Installation

This section describes how to install the OPC Device Server software. This software is distributed as a component of the full set of APACS+ CDs. Typically, this is:

- ProcessSuite CD 1
- ProcessSuite CD 2
- ProcessSuite CD 3
- ProcessSuite CD 4
- Control CD
- PROFIBUS Device Manager (PDM) CD
- COM PROFIBUS CD

The actual complement of CDs may change over time but the OPC Device Server software is distributed on the CD labeled **Control**. The CDs are usually accompanied by a license diskette. In addition, periodic updates to the OPC Device Server Software may be distributed on CD as part of an APACS+ Service Pack.

3.1 Platform Requirements

The platform requirements for the OPC Device Server are the same as that for ProcessSuite or other major APACS+ software components. Generally, this is a Windows® NT™ workstation which includes:

- Intel Pentium® III 450 MHz (or higher) microprocessor
- 128 MB (or more) 100 MHz SDRAM
- 6.4 GB hard disk drive
- 1.44 MB, 3.5" floppy disk drive
- 32X CD-ROM drive
- PCI expansion slots
- One (or two for dual Ethernet systems) Ethernet network card (PCI) that supports 10BASE-T and 100BASE-TX, RJ45 connector
or
- PCI-based M-BUS/M-NET adapter card.
- Mouse port or serial port
- Parallel port (for hard key/printing)

Note that platform requirement may change over time as APACS+ software evolves.

3.2 Installation Considerations

Installation of the OPC Device Server can be performed as part of a comprehensive ProcessSuite installation, such as a Development Node, that includes other software components. The OPC Device Server may also be included in a non-ProcessSuite node. Where applicable, the Windows NT user must be logged in as *Administrator*.

When installing the OPC Device Server software, the installer must select the proper Runtime API (Ethernet or M-BUS) software to match the physical connection to the system.

Each APACS+ CD has its own installation program. Generally, CD 1 to CD 4 must be installed before the Control CD.

Details of software installation are provided on the *Electronic Manuals* CD that is supplied with each system. In the *Getting Started* section, refer to *STEP 5 - Install Your Software*.

Authorization for operating these software components is managed by a companion license file (supplied on diskette) and a hard key. Licensing issues are detailed in section 3.2.1.

A Siemens workstation such as the Rack-mounted Industrial Server (RIS) can be used as the hardware platform for an OPC client workstation. The RIS supplies the needed computing power and has two Ethernet network adapters that are needed to connect to an APACS+ system with dual Ethernets. If third-party hardware is used to host the OPC client, note that two Ethernet network adapters are required to connect to a system's dual Tag Servers.

3.2.1 Software Licensing

A license diskette is provided for each ProcessSuite computer node. The license diskette identifies the unique serial number for a node and contains all the licensing information for that node.

For the APACS+ Control CD, the license diskette must be inserted in the floppy drive of your computer **before** installing software. If applicable, the install program prompts you to overwrite your current license file with the new ProcessSuite license file. The OPC Device Server software will not run if a missing or invalid license is detected on startup; however, an error message is sent to the NT Event log file.

The hard key (hardware key) is installed by attaching it to your computer's parallel port. The hardware key must be in place when running ProcessSuite and/or APACS+ software. If no key is detected, ProcessSuite displays a warning message and access to the programs is denied.

IMPORTANT

The hardware key supplied for each computer node is required for the **operation** of ProcessSuite and/or APACS+ applications; however, it is not required for software or license file installation.

Proper use of the hardware key is extremely important. You must remove the key from the computer if the parallel port is used for any operation other than printing. Do not perform tape backups, file transfer or I/O control, or use a product such as Laplink™ via the parallel port with the key installed, as it may permanently damage your key.

It is strongly recommended that you insure your key(s) for the full value of your ProcessSuite system. Siemens will not replace lost or stolen keys, unless you purchase a new ProcessSuite package.

3.2.2 M-BUS / M-NET Drivers

Installation considerations applicable to MODULBUS (M-BUS) and MODULNET (M-NET) Driver Software version 4.50 are listed in these products' release notes.



4.0 Server Startup and Operation

This section describes how to start, stop, and operate the OPC Device Server Software.

The OPC Device Server software starts automatically by default and operates in conjunction with its related software components; the APACS+ API software and the OPC client application. During software installation, the installer is prompted for the preferred Startup Mode of the OPC Device Server software; either Automatic or Manual. The default selection is Automatic (the OPC Device Server starts automatically when the host workstation is booted). The software; however, can be started and stopped manually if desired via the APACS+ control panel applet.

To access the APACS+ Control Panel applet:

1. Open the Control Panel.
2. Double click the APACS+ icon (Figure 4-1). This action opens the applet. If necessary, select the APACS OPC Device Server tab to open the OPC Device Server Control Panel Display (Figure 4-2).

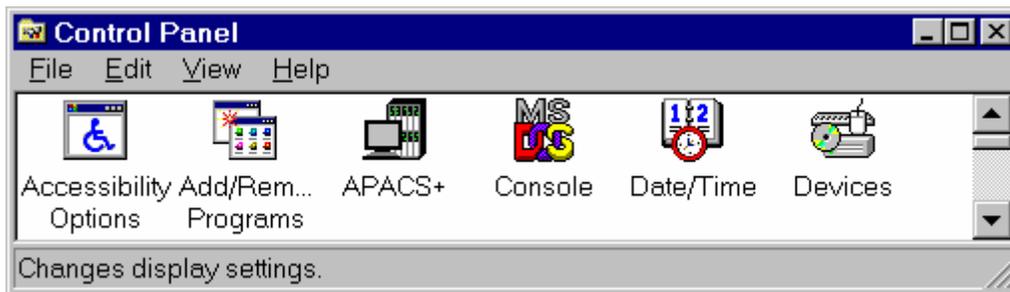


Figure 4-1 Icon for APACS+ Control Panel Applet

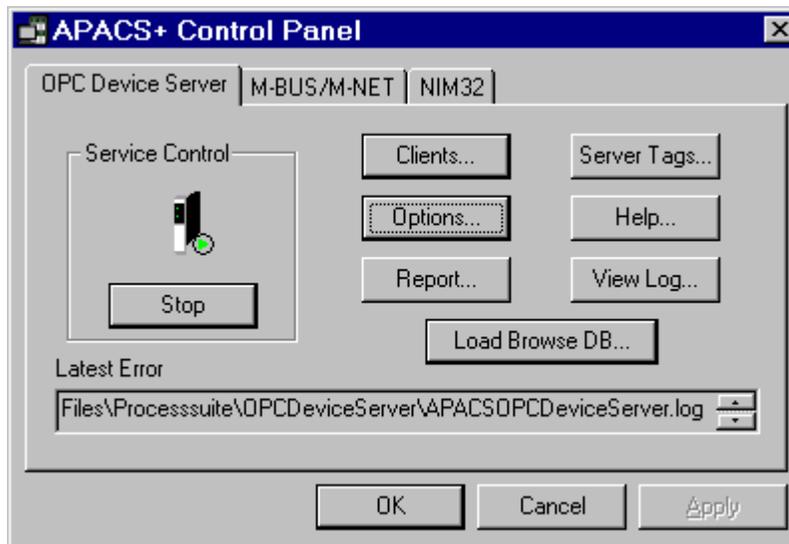
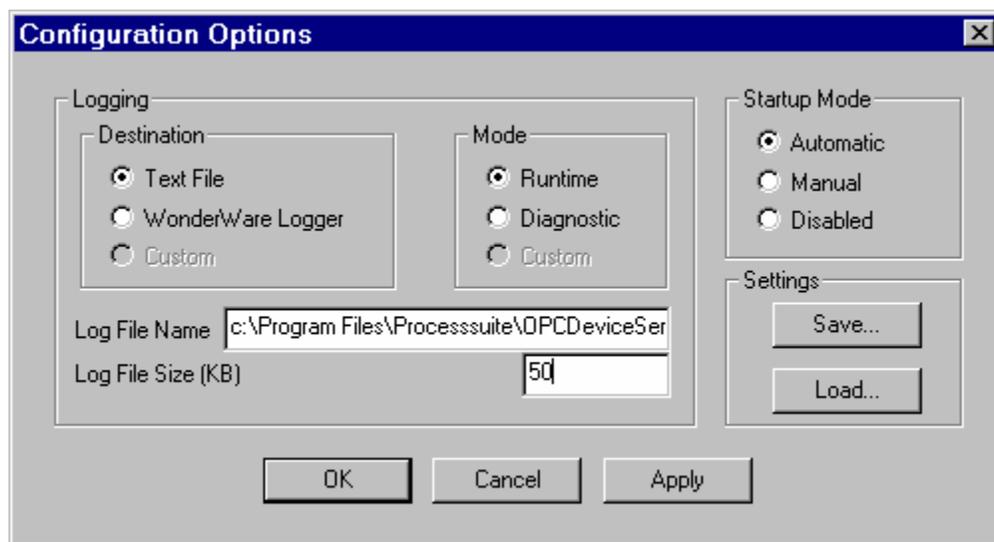


Figure 4-2 OPC Device Server Control Panel Display

To view or change the server's startup mode:

1. From the OPC Device Server Control Panel Display, press the Options... button to access the Configurations Options Display (Figure 4-3).
2. In the Startup Mode group, choose the desired startup option (Figure 4-3) then click OK. These modes are defined as follows:
 - Automatic – This (default) selection causes the OPC Device Server Software to start automatically whenever the workstation is turned on or rebooted.
 - Manual – This selection is used to place the starting or stopping of the OPC Device Server under manual control. When this option is set, the server can be started or stopped by clicking the button in the Service Control area. This mode also permits the OPC Device Server to start when an OPC client application makes a soft-connection to it.
 - Disabled - This selection disables the OPC Device Server software. In this mode, the OPC Device Server is inhibited from operating. For example, an OPC client cannot start the OPC Device Server.

**Figure 4-3 Configuration Options Display****4.1 Operation**

The OPC Device Server software provides a third party OPC (OLE for Process Control) OPC-compliant application with read and write access to an APACS+ Process Automation System's controllers. Since OPC has become a widely used process industry standard, it provides an easy path for accommodating third party applications. The OPC Device Server starts using default values for tunable and setup parameter values. This permits a soft-connection to be quickly established between the OPC client and the

system's controller(s). Once established, operation is mostly automatic; however, the software can be further setup as needed. Sever settings are detailed in section 5.0.

The operating state of the OPC Device Server can be monitored from its icon in the task bar system tray or the Service Control area of the OPC Device Server Control Panel Display (Figure 4-2). In each case, a status indicator's color and shape is determined by the server's operating state as follows;

- Green Pointer = Running
- Red Square = Stopped
- Yellow Pointer = Running with error(s)

Once it is established that the OPC Device Server is running (displaying a green pointer) all tags to be accessed can be configured in the OPC client. The details of initial tag configuration in each OPC client will be specific to the client's method of configuration and covered in the client's documentation. To aid in this configuration, the OPC Device Server supports a convenient client-side Browse function for calling up APACS+ tag information. An example client-side Browse screen is shown in Figure 4-4.

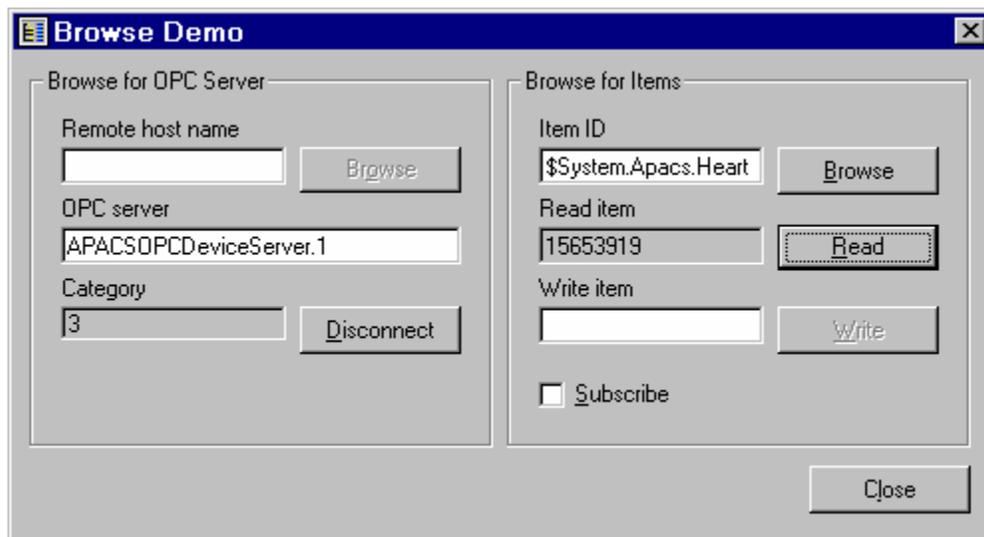


Figure 4-4 Example Browse Display

To enter Item ID information, use the Browse function to call up the hierarchical tag database that represents an OPC Device Server's internal variables (system tags) and a resource configuration such as that of a hardware control module (ACM, CCM) or a soft-module such as Control Engine. The Browse function presents this Item ID information with its proper syntax. This is either the system tag format where each tag has a "\$System." tag name root (example: **\$System.Version**) or in the standard format for an APACS+ controller tag path as detailed here. Note the elements within the braces are optional.

*resource.{program.{derived function block.{...}}}*variable

For example: **Reactors.Area2.Unit3.SteamTemp**

The variable can be any of the following:

- Local
- User-defined global (“[” optional)
- Embedded global (“#” required)
- Function block softlist parameter
- Function block I/O nub (“[” and “]” optional)
- I/O channel (“%” required)
- I/O channel softlist parameter (“%” required for I/O channel)
- Array variable element (“[” and “]” required)

If the variable is an array, the following syntax is used to read an array element. The element is the index of each dimension separated by a comma, as shown in the example.

```
resource.{program.{derived function block.{...}}variable[element]
```

For example: **Reactors.Area2.Unit3.SteamTable[3,2]**

NOTE

Variable names within APACS+ are case-sensitive.

The OPC Device Server provides the data value to the client and writes appropriate values to the APACS+ controllers for all configured tags. The APACS+ data type for each tag is converted to an equivalent OPC data type. For example, an APACS+ controller data type of REAL is converted to an OPC data type of R4. For a complete listing of all default data type conversions, refer to Appendix B. In addition, if the tag configured in the OPC client has been defined with a data type that does not match the equivalent data type, the OPC Device Server automatically handles the conversion. For example, if an OPC client requests a data type of real (R4), and the tag in the APACS+ Controller is integer (INT), the OPC Device Server converts the value to an OPC real type (R4) with an OPC data quality value of Good. If the data type cannot be converted, the OPC Device Server returns an OPC data quality value of Bad. The valid conversions of data types are detailed in Appendix B.

In addition to the OPC data quality reported from data type conversions, the OPC Device Server also passes through the standard APACS+ controller tag data quality to OPC data quality. For example, an APACS+ controller tag with a quality value of Bad is passed through as an OPC data quality value of Bad, and an APACS+ controller tag with a data quality value of Questionable is converted to an OPC data quality value of Uncertain. A complete table of data quality translations is provided in Appendix D.

4.2 Troubleshooting

Troubleshooting server operation is assisted through the use of diagnostic error code(s) and a diagnostic log file. Runtime error codes are detailed in Appendix A. The most recent runtime errors can be viewed in the Latest Error scroll box of the OPC Device Server Control Panel Display (Figure 4-2). Additional error codes and informative records (such as successful connections) are posted to a diagnostic log file. These error codes are also detailed in Appendix A.

Example log files are shown in Figure 4-5. Logging information is sent to the Wonderware® Logger

when the OPC Device Server is installed on a ProcessSuite node, otherwise, logging information is sent to a text file. The OPC Device Server also sends more sever error codes to the NT Event Logger. Logging operations are set up in the Options display (Figure 4-3). There are two logging modes: Runtime mode and Diagnostic mode. For troubleshooting purposes, the Diagnostic mode is to be used because it provides more detail.

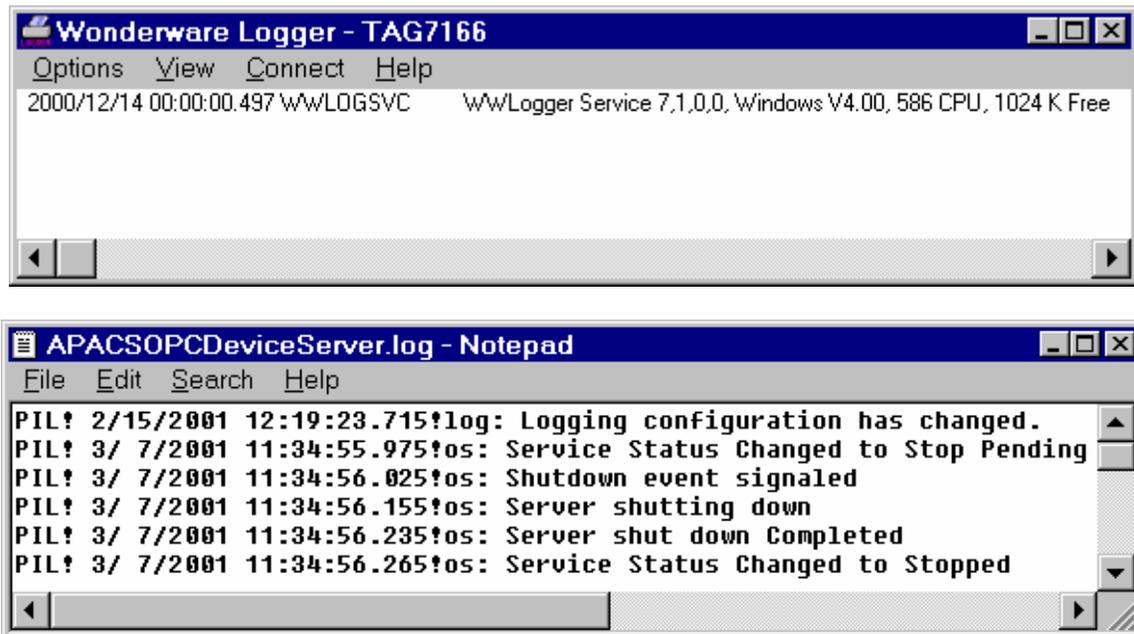


Figure 4-5 Example Log Files

The Runtime mode includes records such as connection lost, or licensing error, server start/stop events, bad tag requests, successful connections (client connection to/disconnection from the server), and failed Read/Writes – on a transaction level (not per tag). Generally, this level of detail is used in a steady-state situation where all configuration and troubleshooting is complete and you only want to be flagged when an anomaly occurs.

The Diagnostic mode includes all records included in the Runtime mode plus every successful transfer (on a transaction level), warnings, successful tag creation, failed read/writes (tag level) and Group creation/removal/orphaned. This level of detail is necessary for troubleshooting, but requires more processing overhead and may reduce OPC Device Server performance.

Each diagnostic record is time-stamped and has an accompanying description. You can allocate the amount of hard disk space that can be consumed by the diagnostic records. The default value is 5000 KB. This parameter only applies to the text file log. Once the logged records consume the allotted space, the software continues to log information by overwriting the oldest records.

To view a log file from system tray icon:

1. Right-click the server's icon in the task bar system tray.
2. From the resulting menu, select View Log File.

To view a log file from the Control Panel applet:

1. Access the APACS+ Control Panel Applet.
2. Click the View Log... button.



5.0 Device Server Settings

The OPC Device Server runs as a Windows™ NT service and access to it is provided via the APACS+ Control Panel Applet. The applet provides a graphical interface to facilitate server setup, monitoring, and status reports. The OPC Device Server's software settings are pre-set to default values. These values can be changed as needed to optimize or adapt Server operation. Server settings are covered in the following topic areas:

- APACS+ Control Panel Applet
- Clients
- Options
- Load Browse DB
- Server Tags
- Reports
- View Log

5.1 APACS+ Control Panel Applet

The APACS+ Control Panel Applet provides a graphical user interface (GUI) to the OPC Device Server. The GUI is an easy way to monitor or change parameters for logging options, error-handling options, certain system tag values (for tuning), and startup options. In addition, the applet can be used to accomplish the following:

- Access the server's online help file
- Indicate the state (operating mode) of the server
- Read and write a subset of system tags
- Reset statistical system tags
- Restore the default and user-defined settings from a file
- Save a Technical Information Center (TIC) report to a file
- Create or update an OPC browse database
- Save the default and user-defined settings to a file
- Start and stop server operation
- View selected system tags
- View the most recent high-severity error log message
- View the overall status of the server
- View the version of the server software

These topics are detailed in the subsections to follow.

There are two ways to open the APACS+ Control Panel Applet to gain access to OPC Device Server Settings. One method is to open the applet from Windows control panel. The other way is to use the OPC Device Server icon on the task bar system tray.

To access the applet from control panel:

1. Open the Control Panel.
2. Double click the APACS+ icon (Figure 5-1). This action opens the applet.

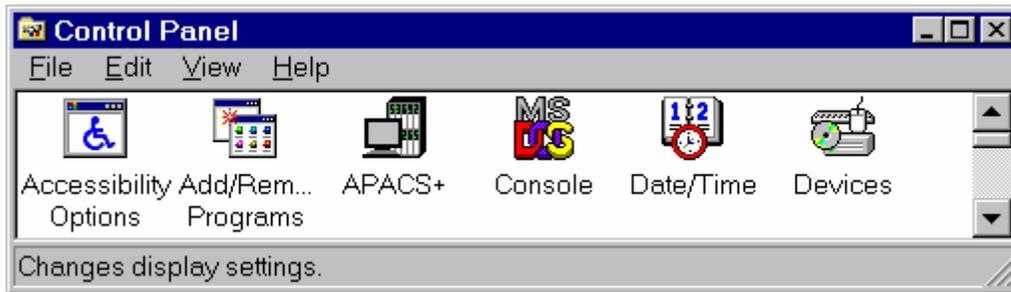


Figure 5-1 Icon for APACS+ Control Panel Applet

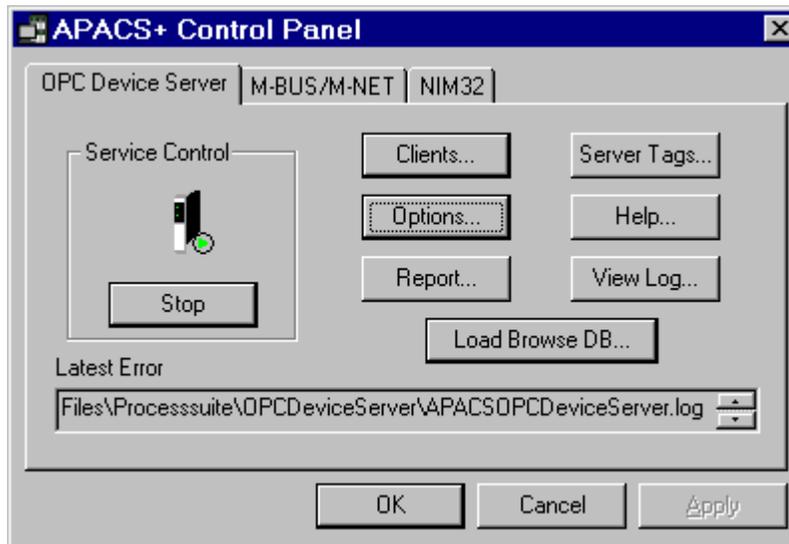


Figure 5-2 OPC Device Server Control Panel Display

To access the applet from system tray:

1. Locate the OPC Device Server icon in the task bar system tray (Figure 5-3).
2. Double click the icon to open the applet. If necessary, select the APACS OPC Device Server tab to open the OPC Device Server Control Panel Display (Figure 5-1).



Figure 5-3 OPC Server Icon in System Tray

5.2 Clients

The number of OPC clients using the OPC Device Server is limited only by memory and processing capacity of the host workstation. Typically, one to five clients are connected.

To view the current list of clients:

1. Open APACS+ Control Panel Applet. If necessary, select the APACS OPC Device Server tab to open the OPC Device Server Control Panel Display (Figure 4-2).
2. Press the Clients... button to open the Connected Clients dialog box (Figure 5-4).
3. Press the OK button to close the dialog box when finished.

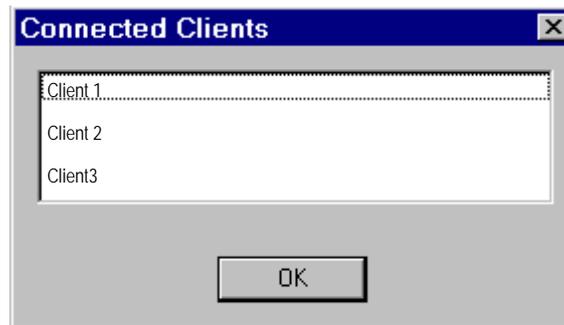


Figure 5-4 Connected Clients Dialog Box

5.3 Options

There are several user-settable OPC Device Server options for:

- Logging
- Runtime
- Startup Mode
- File Settings

These options are described in the subsections to follow. Their parameters are accessible from the Configuration Options dialog box shown in Figure 5-5. Open this dialog box by clicking on the Options... button on the APACS+ Control Panel Display shown in Figure 5-2. Each parameter has a default setting or value.

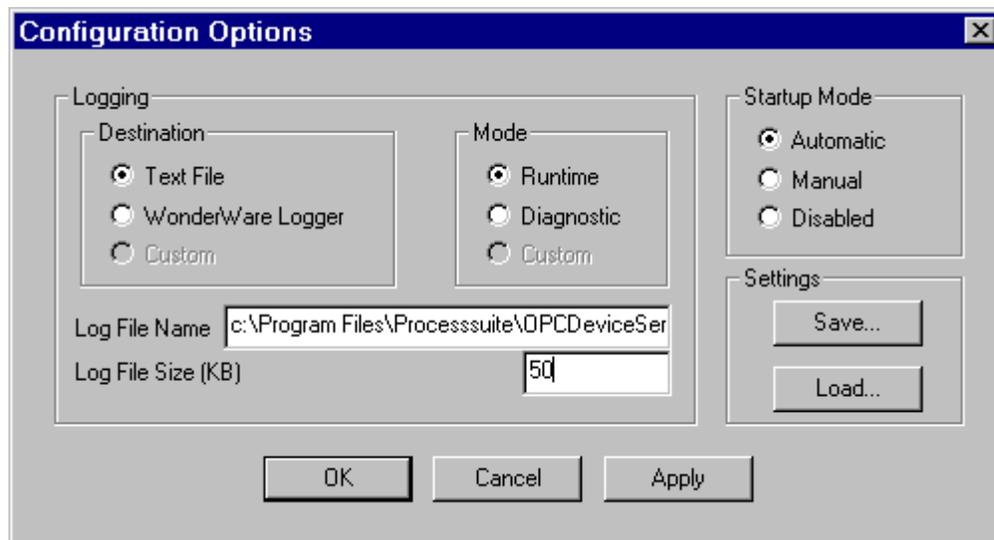


Figure 5-5 Configuration Options Dialog Box

5.3.1 Logging

There are several user-settable logging parameters for the OPC Device Server as listed here. A parameter only needs to be set if you want to change it from its default value. All log entries are automatically date/time-stamped.

The options in the Destination group are used to set the desired format for the contents to be stored in a log file.

Select the Wonderware Logger option to store file entries in the Wonderware logger format if the OPC Device Server is installed on a ProcessSuite node

Select the Text File option to store file entries in a text file, if the OPC Device Server is installed on a non-ProcessSuite node. (The Server will not log messages if another selection is made.) The text file can be viewed with any general-purpose text editor such as WordPad™. When this option is selected, you may also enter a specific directory path and filename for the log file (default = DeviceServer.log) in the Log File Name edit box. While the file can be remote, it is recommended that it be local. Otherwise, should network problems occur, the log itself would not be readily accessible, which would make it difficult to troubleshoot the network problem itself or to assess the impact of the network problem on the OPC Device Server.

The maximum log file size (default = 5000 KB) is adjustable via the Log File Size edit box. Once the file size reaches this parameter value, it wraps and overwrites its oldest data as new data arrives.

The Mode group options are used to set the desired level of detail stored in the log file. Select the Runtime (default) option during normal server operation. Select the Diagnostic option during initial

startup, or troubleshooting sessions so log entries will contain additional engineering information.

5.3.2 Startup Mode

Options to control how the OPC Device Server software starts are provided in the Startup Mode group as follows:

- Automatic (default) – Select this option to start the OPC Device Server software automatically whenever the workstation is turned on or rebooted.
- Manual – Select this option to place the starting or stopping of the OPC Device Server under manual control. When this option is set, the server can be started or stopped by clicking the button in the Service Control area of the APACS+ Control Panel Applet. This mode also permits the OPC Device Server start when an OPC client application makes a soft-connection to it.
- Disabled - This option disables the OPC Device Server software. In this mode, the OPC Device Server is inhibited from operating.

Before the first-time startup of the OPC client application, you must first configure the client to connect to the OPC Device Server. Subsequent startups of the client will automatically start the server.

To configure the OPC client application to connect to the OPC Device Server, you must configure each Item ID (tagname). The procedure will differ depending on the client used. The required Item ID format is detailed in section 4.1, “Operation”.

5.3.3 Settings

Options settings can be stored to and retrieved from a configuration file (*filename.cfg*) using the Save... and Load... buttons.

To store Options settings:

1. From the Configuration Options dialog box (Figure 5-5), click the Save... button in the Settings group area. This opens the Save Configuration dialog box (Figure 5-6).
2. As needed, enter a file name and directory path for the configuration file (default filename = setup_x.cfg) then click the Save button to store the file.



Figure 5-6 Save Configuration Dialog Box

To retrieve Options settings:

1. From the Configuration Options dialog box (Figure 5-5), click the Load... button in the Settings group area. This opens the Load Configuration dialog box (Figure 5-7).
2. As needed, navigate to the directory path and select the desired configuration file (*filename.cfg*) then click the Open button to retrieve the file.

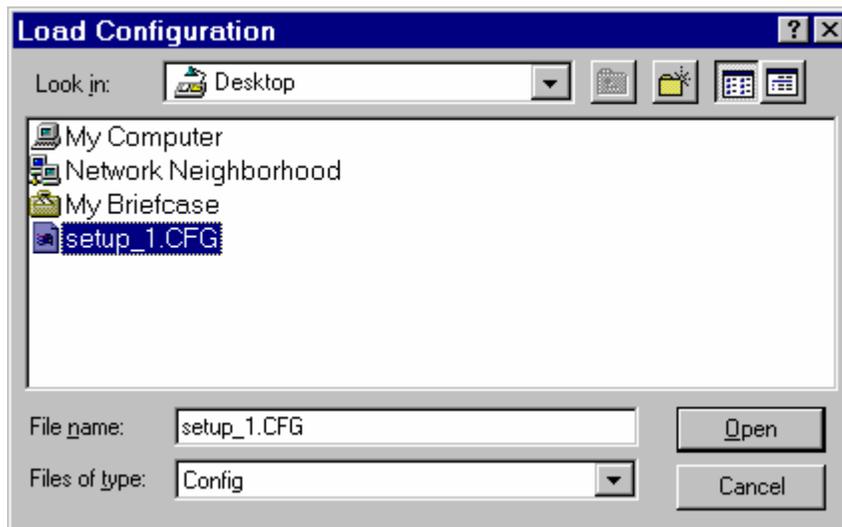


Figure 5-7 Load Configuration Dialog Box

5.4 Load Browse DB

To reduce the time needed to configure an OPC client application, the client's Browse function requires access to a hierarchical database that represents the target resource configuration (e.g. ACM, CCM, or Control Engine). This information is stored in a server-side browse database (MSDE SQL database) that is managed from the OPC Device Server tab of the APACS+ Control Panel applet.

The APACS+ Database Automation Utility assists with the initial creation of the browse database by converting a controller configuration file, exported from the *4-mation*TM Configuration Software as *filename.dbf* file, to a *filename.csv* (comma separated variable) file that can be used for creating the OPC browse database. Alternatively, a .csv file output from a Vision tag server database dump or a hand-edited text file can also be used. To create a hand-edited text file, copy the example custom.csv file supplied in the OPC Device Server's installation directory. The needed layout is described within the file, in comment form.

Creating the OPC browse database requires accessing the .csv output file since it contains APACS+ tag (Item ID) information and is usually located on a development node. This file can be in a local or remote system location. If the file is located at a remote "shared" location, the security level of the remote node must be configured to allow access. When accessing a .csv file, the user can choose to use the Load function to automatically clear the existing browse database information (if any), initialize a new database, and populate it with tag address information. Alternatively, the Append function is used to append new tags to an existing database from a .csv file. The Append function will not remove tags in the database that are not in the .csv file, and will ignore any duplicate tags.

To acquire tag information from a .csv file:

1. From the APACS+ Control Panel applet, select the APACS OPC Device Server tab to open the OPC Device Server Control Panel Display (Figure 5-2) then click the Load Browse DB... button. This opens the APACS+ OPC Browse DB Tag Loader dialog box (Figure 5-8).
2. In the Select .csv field, type in the directory path (or press the local Browse... button to navigate) to the target *filename.csv* file, then click the Append or Load button, as appropriate, to acquire tag information. A progress bar and status information is automatically displayed as needed.

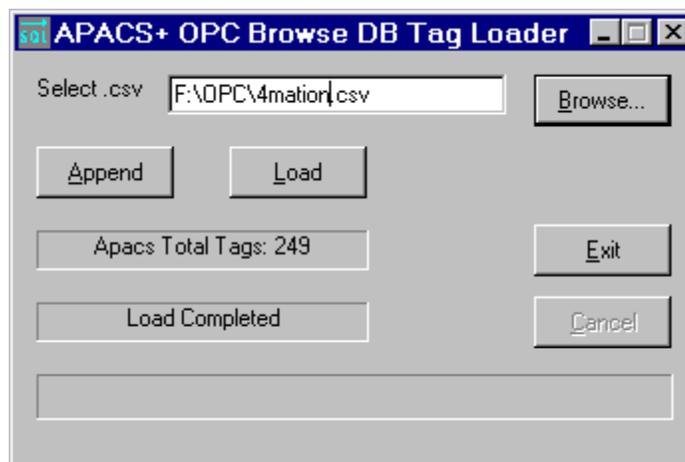


Figure 5-8 APACS+ OPC Browse DB Tag Loader Dialog Box

Once the OPC browse database is created, the OPC client application's Browse function needs to be set up to access the database. The client-side settings needed for this task are presented in section 6.0.

Loading a large OPC browse database with more than 1000 tag entries could affect workstation performance temporarily. To avoid this condition, start with a smaller database and append additional entries as needed.

5.5 Report

The OPC Device Server software has a function for generating and saving a report that may need to be referenced by Siemens' Technical Information Center (TIC) in the event a technical issue needs to be discussed or investigated. The TIC report is a flat text file that contains the OPC Device Server's configuration settings, registry based System tag values (the tags that can be viewed via the Control Panel Applet), the contents of the trace file, and the contents of the text log file (if so configured).

To save a TIC Report:

1. Open APACS+ Control Panel Applet. If necessary, select the APACS OPC Device Server tab to open the OPC Device Server Control Panel Display (Figure 4-2).
2. Press the Report... button to open the Save TIC Report dialog box (Figure 5-9).
3. As needed, enter a file name (*filename.txt*) and directory path for the TIC report file then click the Save button to store the file.

For TIC contact information, refer to section 1.2, *Product Support*.



Figure 5-9 Save TIC Report Dialog Box

5.6 Server Tags

The OPC Device Server software provides a way to view a subset of system tag information, including communications settings, via the APACS+ Control Panel Applet. Certain parameters can also be edited or reset if desired. (A complete set of system tags are listed in Appendix C.)

To view system tags:

1. Open APACS+ Control Panel Applet. If necessary, select the OPC Device Server tab to open the OPC Device Server Control Panel Display (Figure 4-2).
2. Press the Server Tags... button to open the Device Server Tags dialog box (Figure 5-10).
3. As needed, observe or reset statistical system tags. To view communications settings, click the Advanced... button to open the Communications Settings dialog box (Figure 5-11).

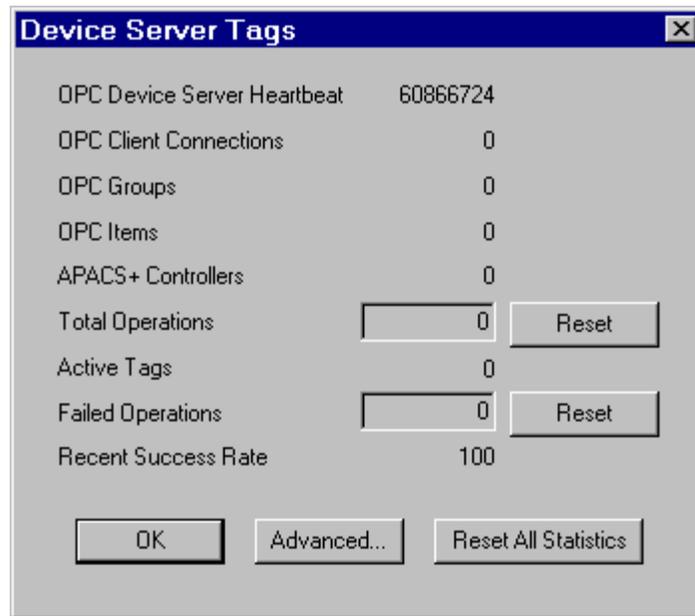


Figure 5-10 Device Server Tags Dialog Box

The system tags listed in Figure 5-10 are defined as follows:

- **OPC Device Server Heartbeat** - The server constantly updates this value every few seconds. It is used as a visual aid to verify that the server is operating.
- **OPC Client Connections** - This is the number of clients that are currently maintaining connections to the server. Essentially, this is the number of clients that are accessing the server.
- **OPC Groups** - The total number of groups-of-information created by all attached OPC clients.
- **OPC Items** - The total number of OPC items being read by all of the connected clients. If two clients are reading the same APACS+ tag, this item count is incremented by two.
- **APACS+ Controllers** - This is the total number of APACS+ controllers from which APACS+ tags have been read.
- **Total Operations** - This is the total number of reads and writes that have occurred to the APACS+ controllers. This can be reset in order to get statistical information about the server (i.e., the number of operations over time.).
- **Active Tags** - This is the total number of active APACS+ tags. If two OPC clients are reading the same APACS+ tag, this count will only increment by one.

- **Failed Operations** - This is the total number of read/write operations to the APACS+ controllers that failed. This can be reset in order to get statistical information about the server (i.e., the number of failed operations over time.)
- **Recent Success Rate** - This value represents the percentage of the last 100 read/write operations to the APACS+ controllers that did not fail.

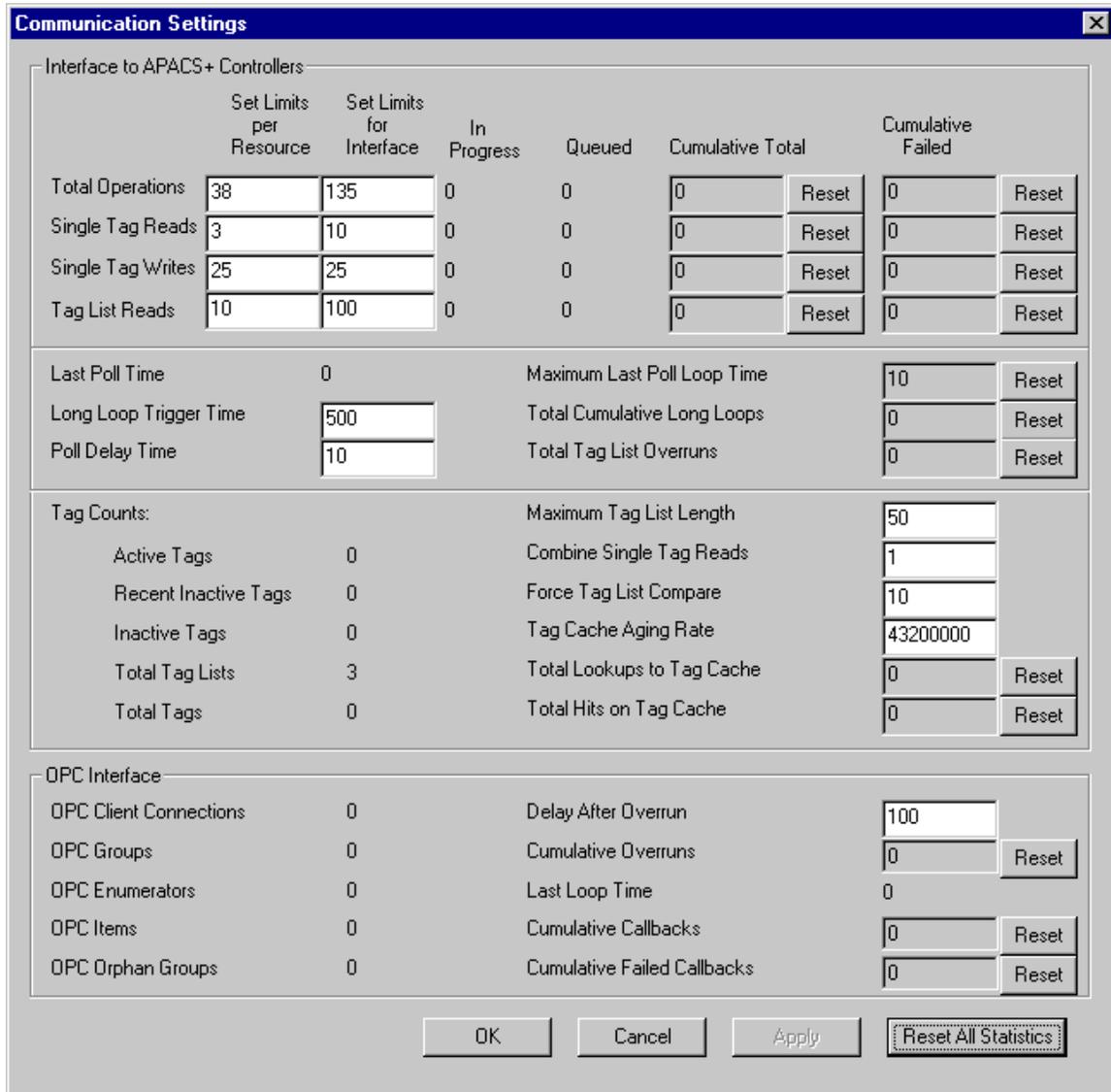


Figure 5-11 Communications Settings Dialog Box

The system tags applicable to communications settings, listed in Figure 5-11, are defined as follows:

Interface to APACS+ Controllers

- **Total Operations: Limits per resource** - This is the number of operations (reads, writes, normal updates) that can be outstanding between the controller and the server at any time. If there are more operations pending in the server, they are waiting for operations at the controller to complete.
- **Total Operations: Limits per Interface** - This is the total number of operations (reads, writes and updates) that can be outstanding between all of the controllers and the server at any time.
- **Total Operations: In progress** - This is the total number of operations that are currently outstanding between all controllers and the server.
- **Total Operations: Queued** - This is the number of operations that are waiting to be processed by an APACS+ controller because there are currently operations outstanding, and these operations would put the number above one of the limits.
- **Total Operations: Cumulative Total** - This is the total number of reads and writes that have occurred to the APACS+ controllers. This can be reset in order to get statistical information about the server (i.e., the number of operations over time.).
- **Total Operations: Cumulative Failed** - This is the total number of read/write operations to the APACS+ controllers that failed. This can be reset in order to get statistical information about the server (i.e., the number of operations over time.).
- **Single Tag Reads: Limit per Resource** - This is the total number of reads (OPC Asynch Read or OPC Synch Read) that can be outstanding between an APACS+ controller and the server. This number should be well below the Total Operations limit in order to keep read requests from preventing other data from updating.
- **Single Tag Reads: Limit per Interface** - This is the total number of reads (OPC Asynch Read or OPC Synch Read) that can be outstanding between all APACS+ controllers and the server. This number should be well below the Total Operations limit in order to keep read requests from preventing other data from updating.
- **Single Tag Reads: In Progress** - This is the number of single tag reads currently outstanding between an APACS+ controller and the server.
- **Single Tag Reads: Queued** - This is the number of single tag reads that are waiting to be processed by an APACS+ controller because there are currently single tag reads outstanding, and these reads would put the number above one of the limits.
- **Single Tag Reads: Cumulative Total** - This is the total number of single tag reads that have occurred between the server and the APACS+ controllers. This can be reset in order to get statistical information about the server (i.e., the number of reads over time.).

- **Single Tag Reads: Cumulative Failed** - This is the total number of single tag reads that have failed between the server and the APACS+ controllers. This can be reset in order to get statistical information about the server (i.e., the number of reads that failed over time.).
- **Single Tag Writes: Limit per Resource** - This is the total number of writes (OPC Asynch Write or OPC Synch Write) that can be outstanding between an APACS+ controller and the server. This number should be well below the Total Operations limit to keep write requests from preventing other data from updating.
- **Single Tag Writes: Limit per Interface** - This is the total number of writes (OPC Asynch Write or OPC Synch Write) that can be outstanding between all APACS+ controllers and the server. This number should be well below the Total Operations limit to keep write requests from preventing other data from updating.
- **Single Tag Writes: In Progress** - This is the number of single tag writes currently outstanding between an APACS+ controller and the server.
- **Single Tag Writes: Queued** - This is the number of single tag writes that are waiting to be processed by an APACS+ controller because there are currently single tag writes outstanding, and these writes would put the number above one of the limits.
- **Single Tag Writes: Cumulative Total** - This is the total number of single tag writes that have occurred between the server and the APACS+ controllers. This can be reset to get statistical information about the server (i.e., the number of writes over time.).
- **Single Tag Writes: Cumulative Failed** - This is the total number of single tag writes that have failed between the server and the APACS+ controllers. This can be reset to get statistical information about the server (i.e., the number of writes that failed over time.).
- **Tag List Reads: Limit per Resource** - This is the total number of update operations (normal background processing) that can be outstanding between an APACS+ controller and the server. This number should be on the order of the Total Operations limit in order to keep the data updating as fast as possible. The data tags are broken up into multiple lists for updating at the controller to optimize performance.
- **Tag List Reads: Limit per Interface** - This is the total number of update operations (normal background processing) that can be outstanding between all APACS+ controllers and the server. This number should be on the order of the Total Operations limit to keep the data updating as fast as possible.
- **Tag List Reads: In Progress** - This is the number of update operations currently outstanding between an APACS+ controller and the server.
- **Tag List Reads: Queued** - This is the number of update operations that are waiting to be processed by an APACS+ controller because there are currently update operations outstanding, and these update operations would put the number above one of the limits.

- **Tag List Reads: Cumulative Total** - This is the total number of update operations that have occurred between the server and the APACS+ controllers. This can be reset to get statistical information about the server (i.e., the number of updates over time.).
- **Tag List Reads: Cumulative Failed** - This is the total number of update operations that have failed between the server and the APACS+ controllers. This can be reset in order to get statistical information about the server (i.e., the number of updates that failed over time.).

Tag Counts

- **Active Tags** - This is the total number of active APACS+ tags. If two OPC clients are reading the same APACS+ tag, this count will only increment by one.
- **Recent Inactive Tags** - This is the number of APACS+ tags that have become inactive within the last 12 hours.
- **Inactive Tags** - This is the number of APACS+ tags that have been inactive for more than 12 hours. These need to be re-verified by the APACS+ controller.
- **Total Tag Lists** - This is the total number of APACS+ tag lists. The tags are broken up into multiple tag lists to optimize read performance.
- **Total Tags** - This is the total number of APACS+ tags that are currently created in the server.
- **Maximum Tag List Length** - This is the maximum number of tags allowed in a tag list. This number can be changed to further optimize tag update performance.
- **Combine Single Tag Reads** - This setting allows multiple single tag reads to be combined into a larger list in order to optimize performance of single tag reads.
- **Force Tag List Compare** - This is the number of updates from the APACS+ controller where the value is actually read from the APACS+ controller whether it has changed or not, and injected into the server. Normally, data updated from an APACS+ controller only occurs when a value has changed.
- **Tag Cache Aging Rate** - When a tag is created in the server, then deleted, information about the tag is stored in a cache so that future operations to the same tag can be quicker. This value represents the time, in milliseconds, that a tag will be held in the cache.
- **Total Lookups to Tag Cache** - This is the total number of times that the tag cache has been checked for a tag that was previously used. This can be reset to get statistical information (i.e., the number of tag look-ups over time.)
- **Total Hits on Tag Cache** - This is the total number of times that the tag cache was checked, and the requested item was found in it. This can be reset in order to get statistical information (i.e., the number of successful tag look-ups over time.).

OPC Interface

- **OPC Client Connections** - This is the number of clients that are currently maintaining connections to the server. Essentially, this is the number of clients that are accessing the server.
- **OPC Groups** - The total number of groups of information created by all attached OPC clients.
- **OPC Enumerators** - This is the total number of OPC enumerators that are currently allocated.
- **OPC Items** - The total number of OPC items being read by all of the connected clients. If two clients are reading the same APACS+ tag, this item count will be incremented by two.
- **OPC Orphan Groups** - This is the number of groups that have been removed, but there is still one or more COM interface pointer references outstanding. When all of these pointers are released, the group deletes itself.
- **Delay After Overrun** - This is the amount of time to wait after a client update takes longer than the OPC group update rate. This allows some dead time to get other work done.
- **Cumulative Overruns** - This is the total number of OPC Group overruns that have occurred. This can be reset in order to get statistical information (i.e., the number of overruns over time). An OPC Group overrun is defined as an “OnDataChange” processing cycle that took longer than the configured OPC group update rate. This cycle includes both the time it takes to check the server’s read cache (for values that have changed) and the time needed to complete sending a data change callback to the connected OPC client (including COM/DCOM communications time).
- **Last Loop Time** - This is the amount of time, in milliseconds, that it took to complete an “OnDataChange” processing cycle as described above in **Cumulative Overruns**.
- **Cumulative Callbacks** - This is the total number of attempted OPC Data Access 2.0x data callbacks that have occurred between the server and all connected OPC clients. This total is inclusive for all connected OPC clients across all types of Data Access 2.0x data callbacks (READ/WRITE/REFRESH/CHANGE/CANCEL). This total can be reset to get statistical information about the server (i.e., the number of callbacks over time.).
- **Cumulative Failed Callbacks** - This is the total number of attempted OPC Data Access 2.0x data callbacks that have failed between the server and all connected OPC clients. This total is inclusive for all connected OPC clients across all types of Data Access 2.0x data callbacks (READ/WRITE/REFRESH/CHANGE/CANCEL). The difference between this total and the “Cumulative Callbacks” total represents the total number of successful callback attempts. This total can be reset in order to get statistical information about the server (i.e., the number of callbacks that failed over time.).

5.7 Help

The OPC Device Server software provides a way to view its on-line help information via the APACS+ Control Panel Applet.

To view on-line help information:

1. Open APACS+ Control Panel Applet. If necessary, select the APACS OPC Device Server tab to open the OPC Device Server Control Panel Display (Figure 4-2).
2. Press the Help... button to open the OPC Device Server help file.
3. Navigate the help file as needed.

5.8 View Log

The OPC Device Server software provides a way to view the current log file via the APACS+ Control Panel Applet. The log file lists error code messages and informative records (such as successful connections). During server operations, logging information is sent to a standard text file or the Wonderware® Logger (default).

To view a text log file:

1. Open APACS+ Control Panel Applet. If necessary, select the OPC Device Server tab to open the OPC Device Server Control Panel Display (Figure 4-2).
2. Press the View Log... button to open the current log file. Error codes are detailed in section 4.2.2. An example log file is shown in Figure 4-4.

NOTE

The Wonderware Logger creates a new log file every 24 hours or each time the workstation is stopped and restarted. Wonderware log files have an .eah extension and are stored in the ...\\PROCESSUITE\\COMMON directory. To view this type of log file, open it directly with either Windows Wordpad or the Wonderware Log Viewer utility. For more information about Wonderware log files, consult the Wonderware documentation.

5.9 Distributed OPC Device Server to OPC Client Communications (DCOM)

Distributed architectures where an OPC Device Server and an OPC client application reside on separate workstations or nodes depend on DCOM (Distributed COM) technology that is provided with Windows. DCOM is based on Microsoft's COM (Component Object Model) software architecture, which defines object interfaces. DCOM defines the communications that allows objects to be run remotely over a network.

Examples of distributed architectures featuring OPC components are presented in section 2.2. A typical distributed arrangement is to locate an OPC Device Server on the same node as a System Tag Server while the OPC client application resides on another node. To implement a distributed architecture requires both the server-side and client-side nodes be set up for DCOM.

5.9.1 DCOM Set-Up for OPC Device Server

DCOM-based communications are set up within Windows (NT 4.0 or higher). To access the DCOM configuration area of Windows, enter **DCOMCNFG** on the Run command line. The user interface for changing DCOM settings on Windows NT and Windows 2000 workstations is different than the user interface provided on Windows XP and Windows 2003 Server workstations. In addition, Windows XP Service Pack 2 and Windows 2003 Server Service Pack 1 both added a software firewall, which if left enabled, will require the user to make some configuration setting changes in order to support DCOM network communication between an OPC client workstation and a separate OPC server workstation.

Section 5.9.2 describes the DCOM setting changes as viewed from the **DCOMCNFG** user interface when invoked on Windows XP and Windows 2003 Server workstations. Section 5.9.3 describes how to change the software firewall settings on Windows XP and Windows 2003 Server workstations when the firewall is enabled. Section 5.9.4 describes the DCOM setting changes as viewed from the **DCOMCNFG** user interface when invoked on Windows NT and Windows 2000 workstations.

5.9.2 Windows XP and 2003 Server DCOM Set-Up

Service Pack 2 for Windows XP and Service Pack 1 for Windows 2003 Server have made some security enhancements to DCOM; two in particular need to be taken into consideration when using OPC on a network:

1. The default Launch and Access permissions dialogs have been modified to allow the user to configure "limits" on the permissions given to applications using DCOM.
2. For each user now defined in the Launch and Access permissions, both local and remote access can be explicitly defined.

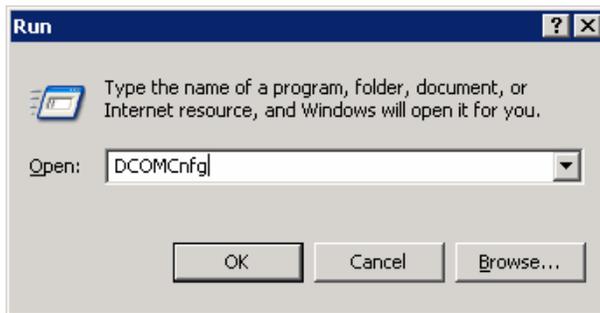
Launch permissions define who can launch a COM based application (such as an OPC server) both over the network or locally. Access permissions define who can access that application once it has been launched. Launch and Access permissions can be set one of three ways:

1. explicitly defined,
2. by default, or
3. programmed.

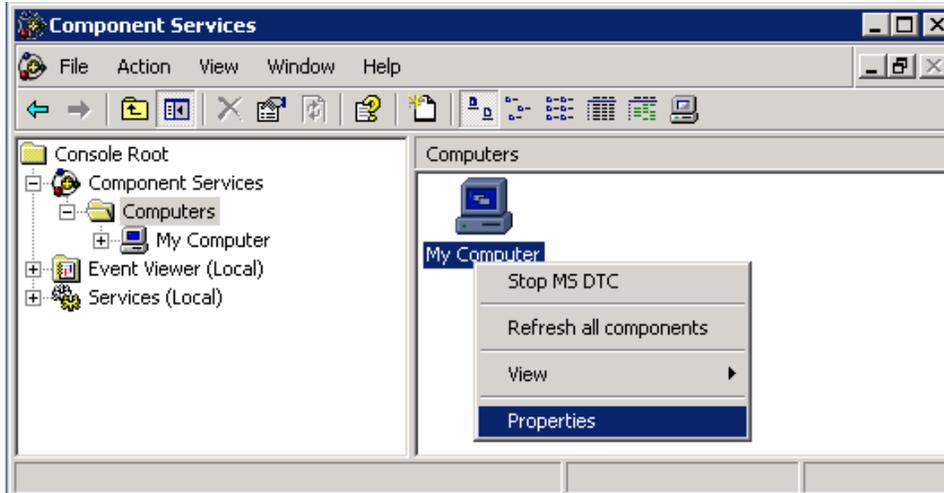
Because an application could set its own permissions programmatically, the explicitly defined or default settings, although set properly, may not be used and therefore the user is not able to explicitly have

control over these settings. To overcome this security flaw, Microsoft has added “limits” to the DCOM security settings from Launch and Access to limit the permissions that an application can use. This limit prevents the application from using permissions beyond what is specified in the DCOM configuration settings. By default the limits set by Service Pack 2 will not allow for OPC communications over the network. In addition to the new permissions limits, one must now specify if the user or group specified has permissions locally or remotely (or both). In order for OPC applications to work over the network with DCOM, the permissions must be set such that remote users can launch and/or access the OPC servers and clients on the machine.

1. From the Windows **Start** button, select **Run** and type **DCOMCnfg** in the **Open:** edit box. Click the **OK** button.

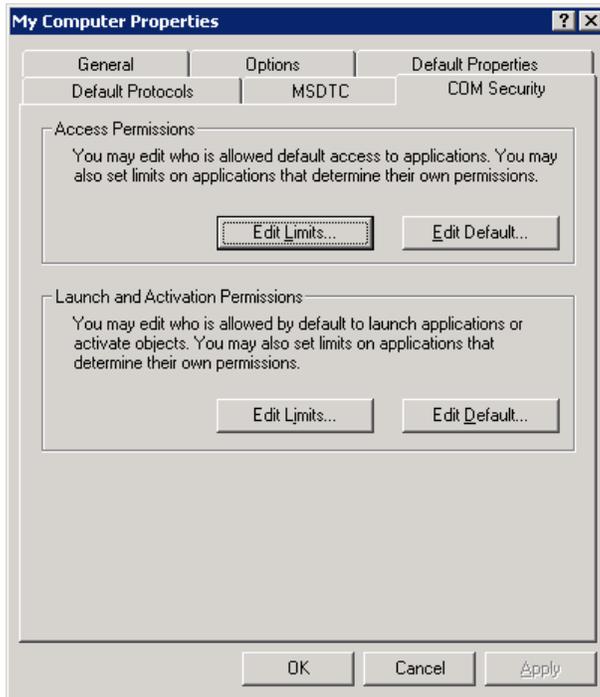


The **Component Services** dialog box opens.

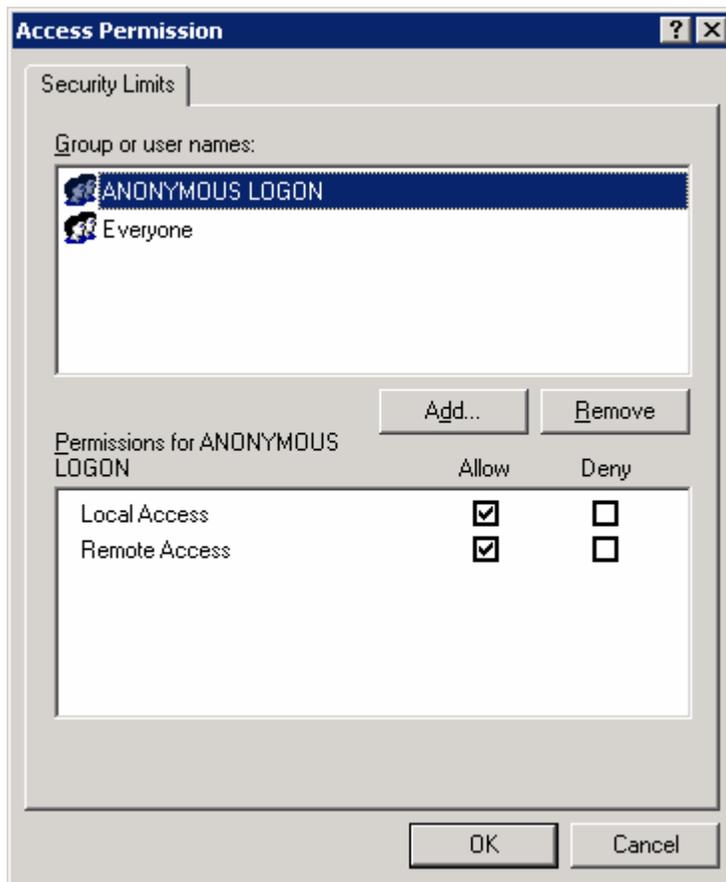


2. Under the **Console Root** folder, expand **Component Services**.
3. Under **Component Services**, expand the **Computers** folder.
4. Right-click **My Computer** and select **Properties**.

The **My Computer Properties** dialog box opens.



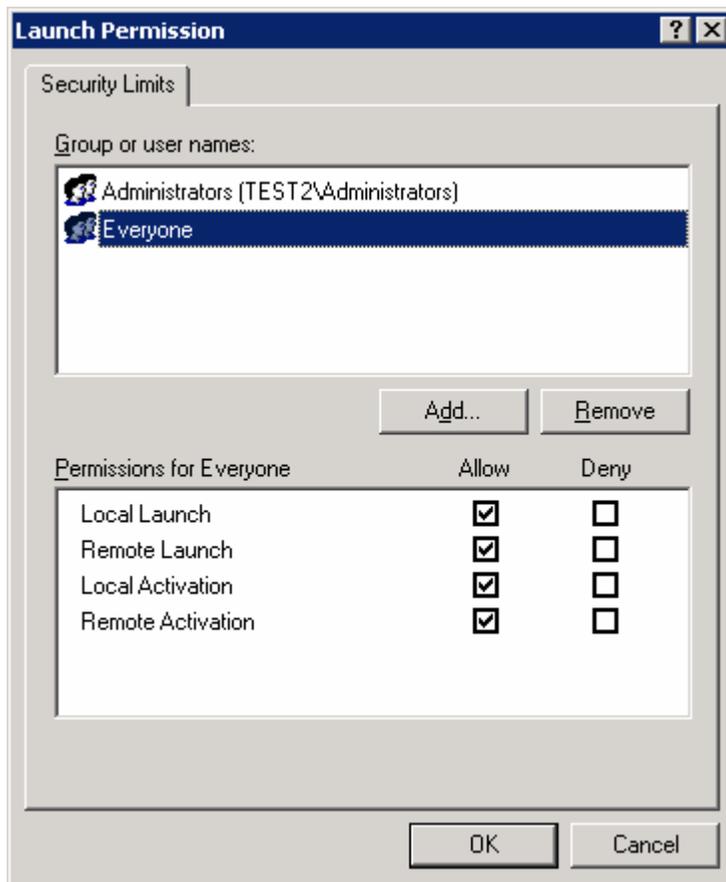
5. Select the **COM Security** tab. In the **Access Permissions** pane, click the **Edit Limits...** button. The **Access Permission** dialog box opens.



- From the **Security Limits** tab, in the **Group or user names:** pane, select **ANONYMOUS LOGON**. In the **Permissions for ANONYMOUS LOGON** pane, select the **Allow** check box for **Remote Access**. Click the **OK** button.

Note: This setting is necessary for OPCEnum.exe to function and for some OPC Servers and Clients that set their DCOM 'Authentication Level' to 'None' in order to allow anonymous connections. If you do not use OPCEnum, you may not need to enable remote access for anonymous users.

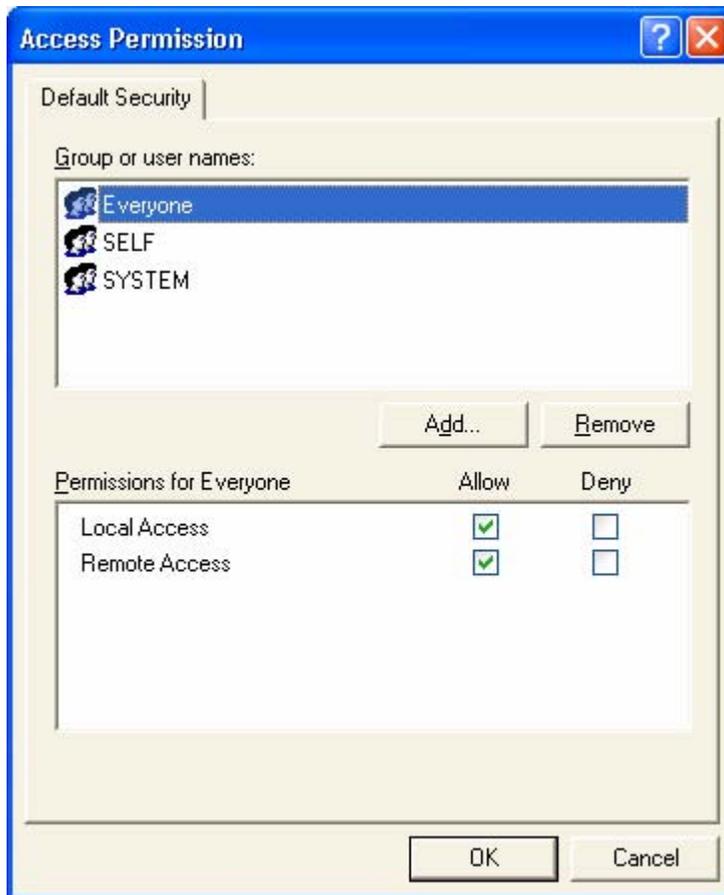
- From the **My Computer Properties** dialog box, select the **COM Security** tab. In the **Launch and Activations Permissions** pane, click the **Edit Limits...** button. The **Launch Permission** dialog box opens.



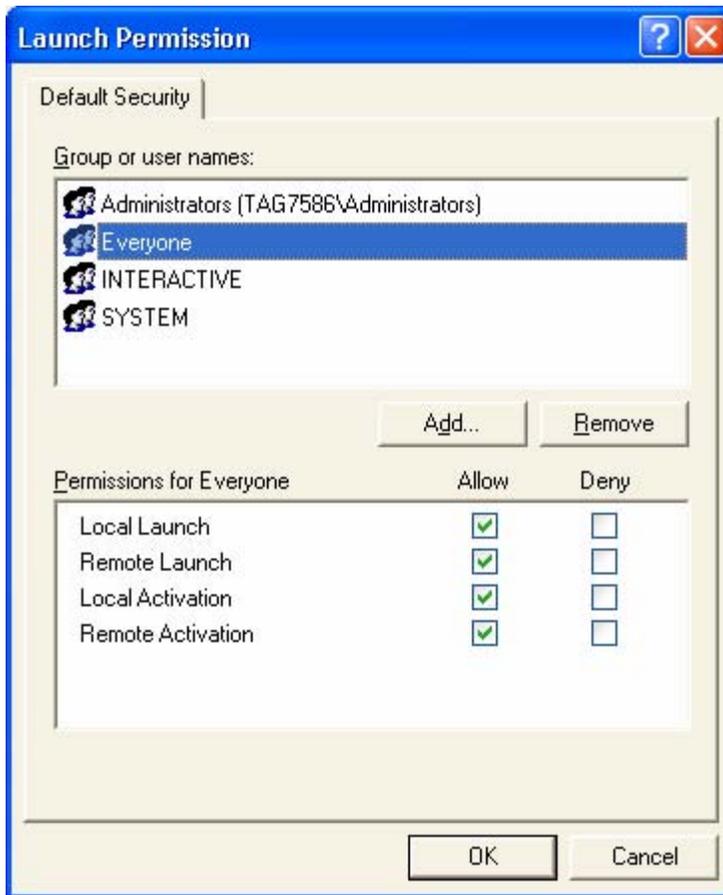
- From the **Security Limits** tab, in the **Group or user names:** pane, select **Everyone**. In the **Permissions for Everyone** pane, select the **Allow** check boxes for **Remote Launch** and **Remote Activation**. Click the **OK** button.

Note: Since Everyone includes all authenticated users, it is often desirable to add these permissions to a smaller subset of users. One suggested way to accomplish this is to create a group named “OPC Users” and add all user accounts to this group that will execute any OPC Server or Client. Then substitute “OPC Users” everywhere that Everyone appears in these configuration dialogs.

9. From the **My Computer Properties** dialog box, select the **Com Security** tab. In the **Access Permissions** pane, click the **Edit Default...** button. The **Access Permission** dialog box opens.



10. From the **Default Security** tab, in the **Group or user names:** pane, select the **Everyone** user or for each user (or group) that participates in OPC communication (e.g. “OPC Users”). In the **Permissions for Everyone** pane, select the **Allow** check boxes for **Local Access** and **Remote Access**. Click the **OK** button.
11. From the **My Computer Properties** dialog box, select the **Com Security** tab. In the **Launch and Activation Permissions** pane, click the **Edit Default...** button. The **Launch Permission** dialog box opens.



- From the **Default Security** tab, in the **Group or user names:** pane, select the **Everyone** user or for each user (or group) that participates in OPC communication (e.g. “OPC Users”). In the **Permission for Everyone** pane, select the **Allow** check boxes for **Local Launch, Remote Launch, Local Activation, and Remote Activation**. Click the **OK** button.

5.9.3 Windows XP and 2003 Server Software Firewall Set-Up

The Windows Firewall allows traffic across the network interface when initiated locally, but by default stops any incoming “unsolicited” traffic. However, this firewall is “exception” based, meaning that the administrator can specify applications and ports that are exceptions to the rule and can respond to unsolicited requests. The firewall exceptions can be specified at two main levels, the application level and the port and protocol level. The application level is where you specify which applications are able to respond to unsolicited requests and the port and protocol level is where you can specify the firewall to allow or disallow traffic on a specific port for either TCP or UDP traffic. To make any OPC client/server application work via DCOM, changes need to be made on both levels.

By default the windows firewall is set to “On”. This setting is recommended by Microsoft and by OPC to give your machine the highest possible protection. For trouble shooting, you may wish to temporarily turn off the firewall to prove or disprove that the firewall configuration is the source of any communication failure.

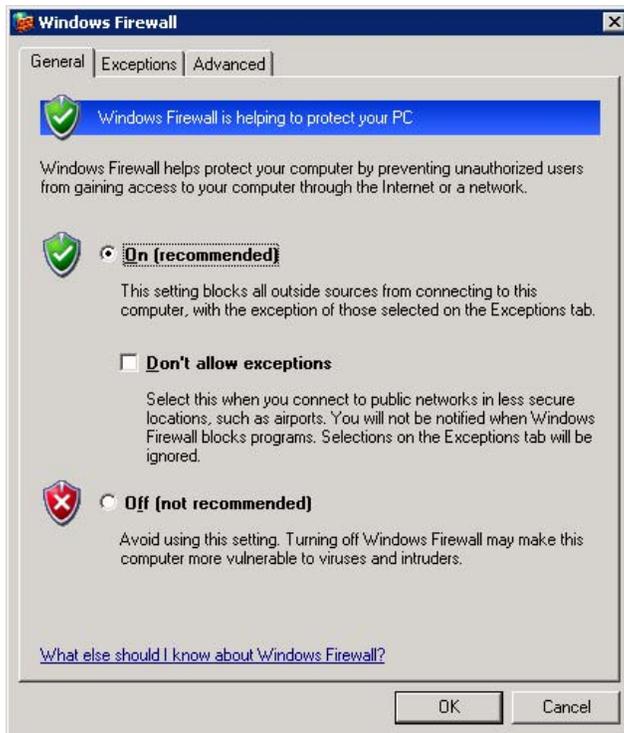
It may be appropriate to permanently turn off the firewall if the machine is sufficiently protected behind a corporate firewall. When turned off, the individual firewall settings outlined here need not be performed to allow OPC communication.

Configuring the Firewall

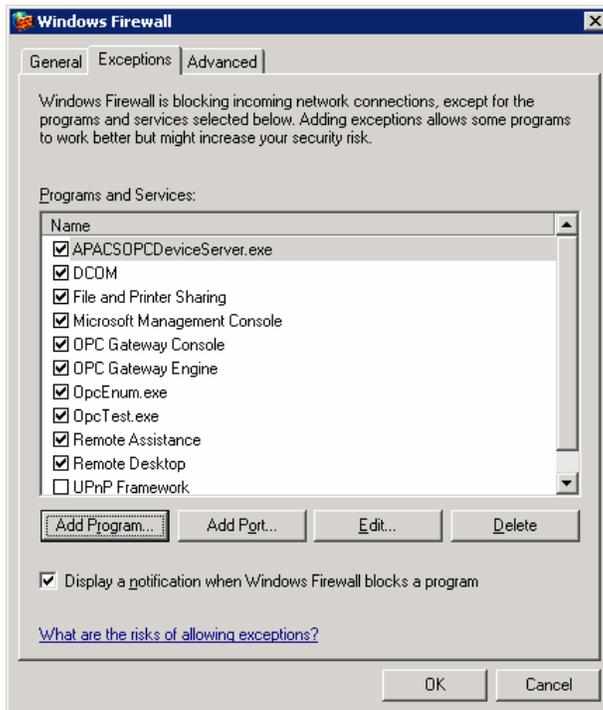
1. From the Windows **Start** button, select **Program > Accessories > System Tools > Security Center**. The **Windows Security Center** dialog box opens.



2. Click the **Windows Firewall** icon.
The **Windows Firewall** dialog box opens.



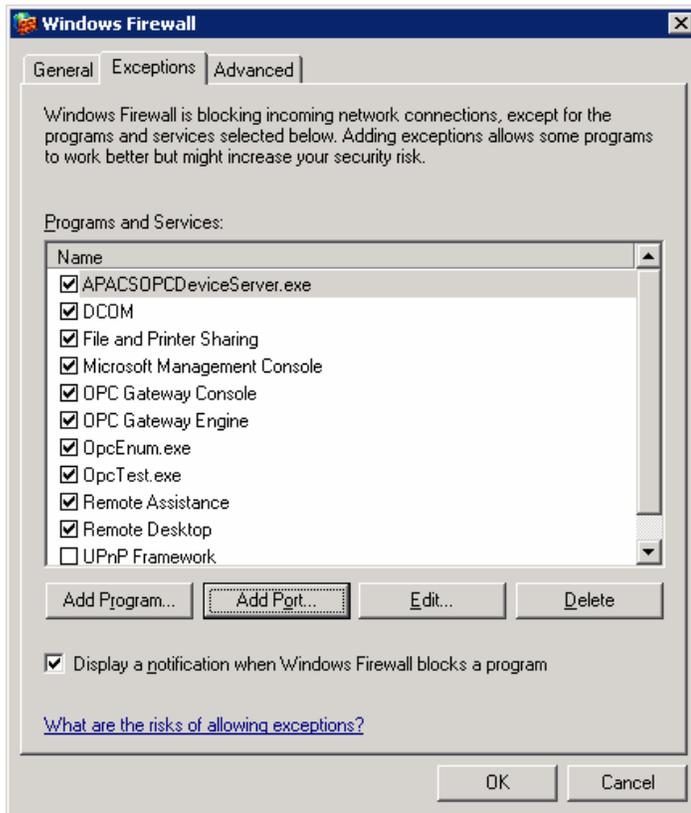
3. Select the **Exceptions** tab, and click the **Add Program...** button to add all OPC Clients and Servers to the exception list. Also add Microsoft Management Console (used by the DCOM configuration utility in the next section) and the OPC utility OPCEnum.exe found in the Windows\System32 directory.



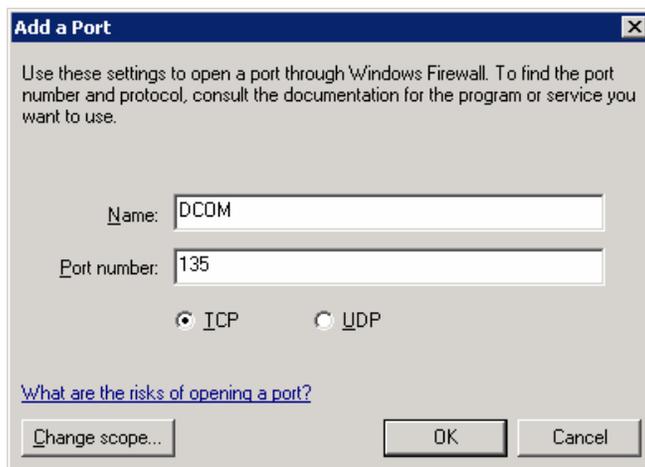
The **Add a Program** dialog box opens.



4. In the **Add a Program** dialog, there is a listing of most applications on the machine, but note that not all of them show up on this list. Use the **Browse** button to find other executables installed on the computer.
Note: Only EXE files are added to the exceptions list. For in-process OPC Servers and Clients (DLLs and OCXs) you will need to add the EXE applications that call them to the list instead.
5. Click the **OK** button after adding each program.
6. From the **Windows Firewall** dialog box, select the **Exceptions** tab and click the **Add Port...** button.



The **Add a Port** dialog box opens.



7. Add TCP port 135 as it is needed to initiate DCOM communications, and allow for incoming echo requests. In the **Add a Port** dialog box, fill out the fields as follows:
Name: DCOM
Port number: 135
Select the **TCP** radio button. Click the **OK** button.
8. Click the **OK** button on the **Windows Firewall** dialog box to complete the set up.

5.9.4 Windows NT and 2000 DCOM Set-Up

DCOM-based communications are set up within Windows (NT 4.0 or higher). To access the DCOM configuration area of Windows, enter **DCOMCNFG** on the Run command line. This opens the Distributed COM Configuration Properties window at the Applications tab level as shown in Figure 5-12. From this point you must take two set-up paths:

- The first path requires that you systematically step through the Default Properties, Default Security, and Default Protocol tabs while setting properties as indicated in Figures 5-13 to 5-15. Except where noted, default values are used.
- The second path requires that you select the APACS+ OPC Device Server from the displayed list of DCOM applications then press the **Properties...** button. Systematically step through the General, Location, Security, Identity, and Endpoints tabs while setting properties as indicated in Figures 5-16 to 5-20. Except where noted, the default values are used.

When a remote OPC client is to connect to the APACS+ OPC Device Server, the same *username* and *password* used to log on that computer must be entered in the **User Manager** table of the APACS+ OPC Device Server computer. It does not matter to what group it is assigned (User or Administrator) but it is important that both the *username* and *password* be identical.

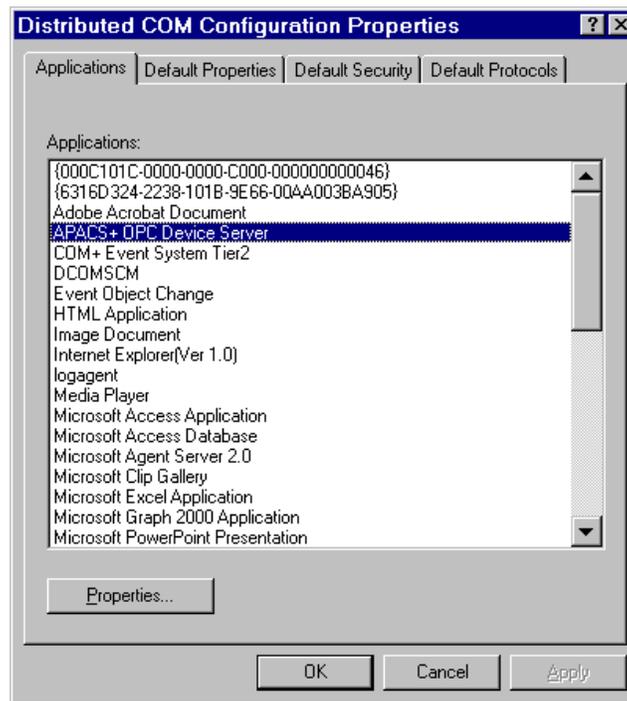
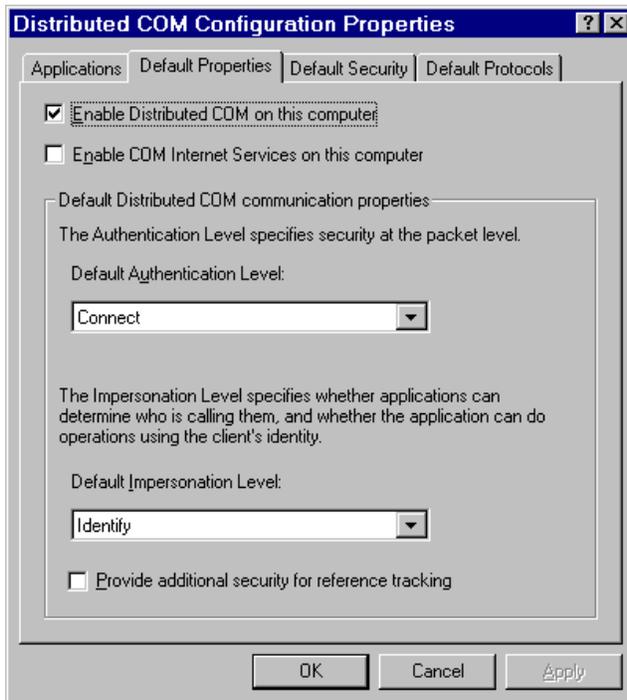


Figure 5-12 DCOM Configuration, Applications Tab

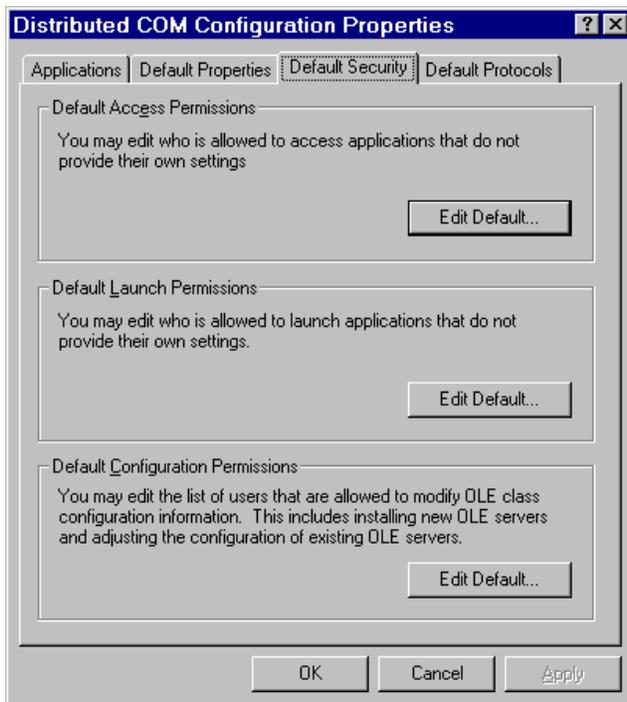


(Select checkbox: “Enable Distributed COM on this computer”)

(For “Default Authentication Level,” choose Connect.)

(For “Default Impersonation Level,” choose Identify.)

Figure 5-13 DCOM Configuration, Default Properties Tab

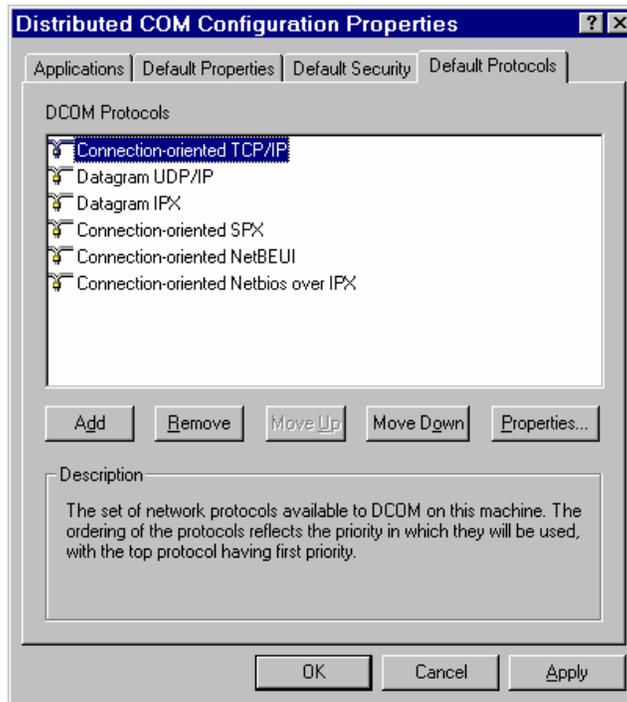


(For Default Access Permissions, make sure “Everyone - Allow Access” is listed)

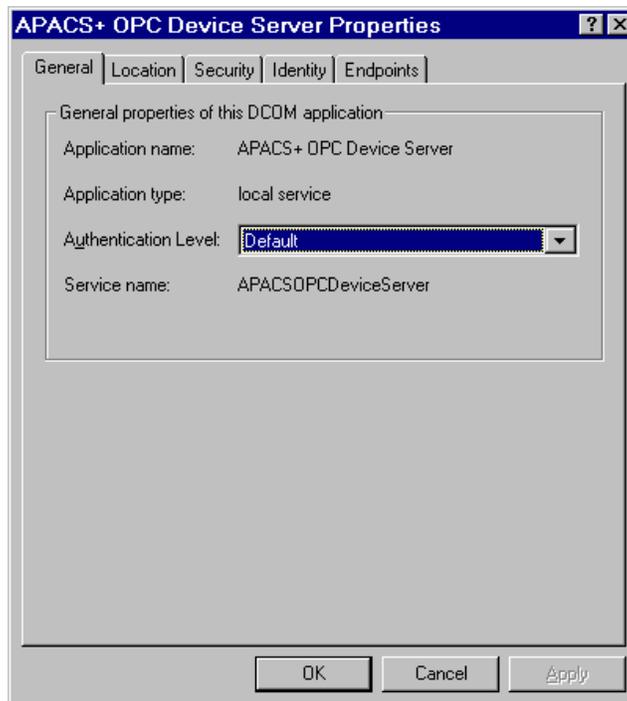
(For Default Launch Permissions, make sure “Everyone - Allow Access” is listed)

(For Default Configuration Permissions, make sure “Everyone - Read” is listed)

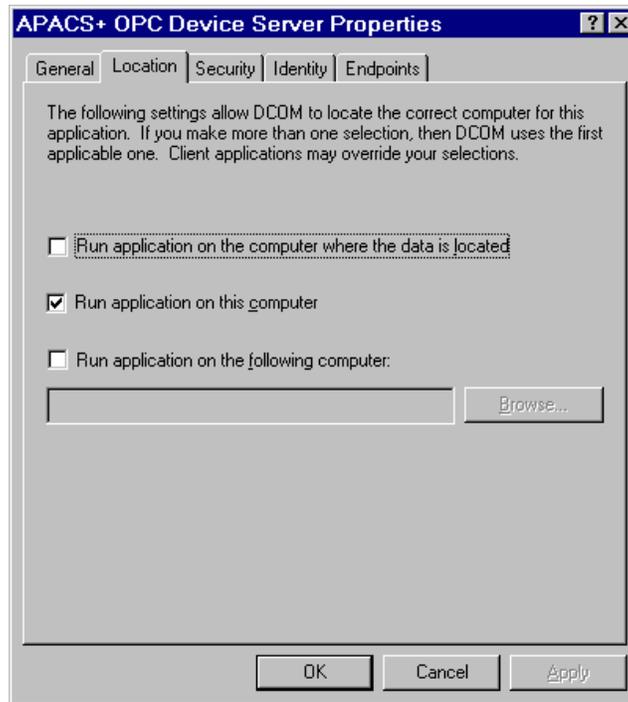
Figure 5-14 DCOM Configuration, Default Security Tab



(Make sure “Connection-oriented TCP/IP” is listed on top)
Figure 5-15 DCOM Configuration, Default Protocols Tab

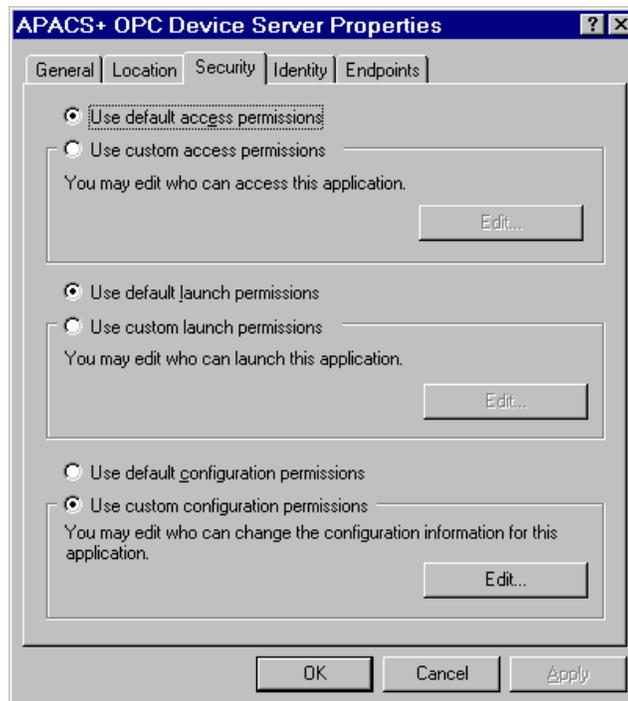


(For Application Level, use “Default” setting)
Figure 5-16 OPC Device Server Properties, General Tab



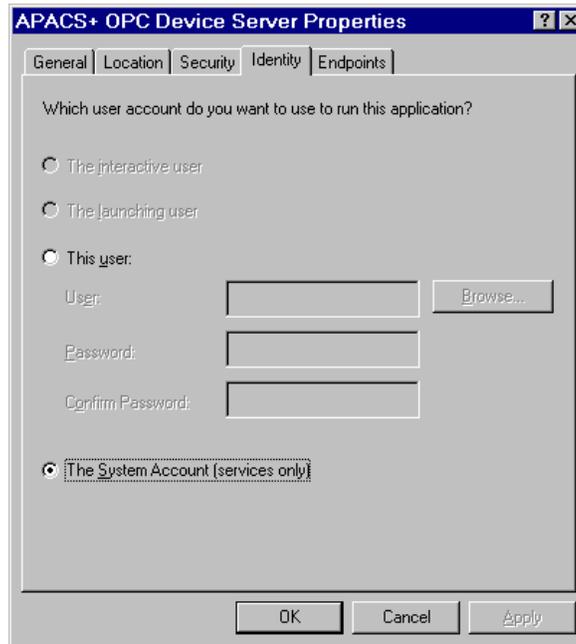
(Check “Run application on this computer”)

Figure 5-17 OPC Device Server Properties, Location Tab



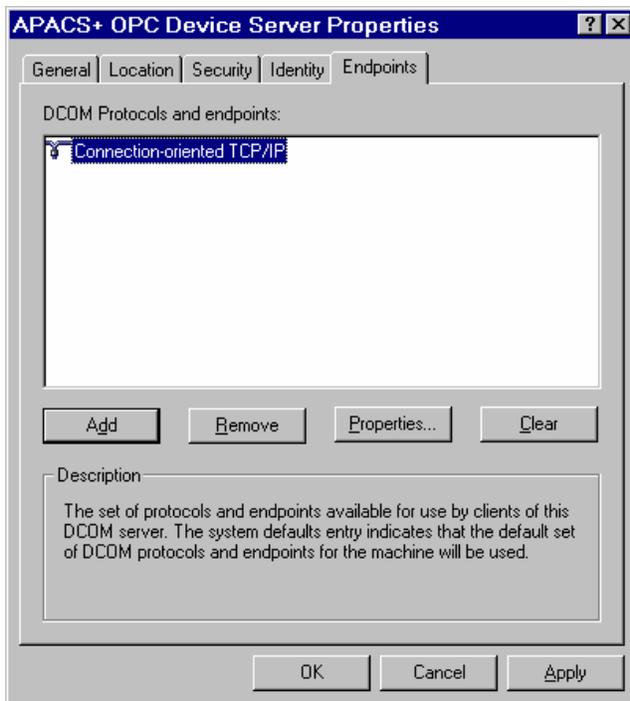
(Check “Use custom configuration permissions” then press Edit...button. For Registry Key Permissions, make sure everyone has “Read” access)

Figure 5-18 OPC Device Server Properties, Security Tab



(Check “The System Account on this computer”)

Figure 5-19 OPC Device Server Properties, Identity Tab



Press the Add button to open the Select DCOM Protocol and End point dialog box.

Select “Connection-oriented TCP/IP” in the drop down list box.

In the Endpoint Assignment group, select “Use Default endpoint.”

Press OK to accept changes.

Figure 5-20 OPC Device Server Properties, Endpoints Tab

6.0 Client Browse Function Settings

To reduce the time needed to initially configure an OPC client application, the client's Browse function can be used to access both OPC Device Server system tags and APACS+ tag address (Item ID) information from a browse database. The actual implementation of a client's Browse function may vary from client to client so this section uses an example client as represented in Figure 6-1.

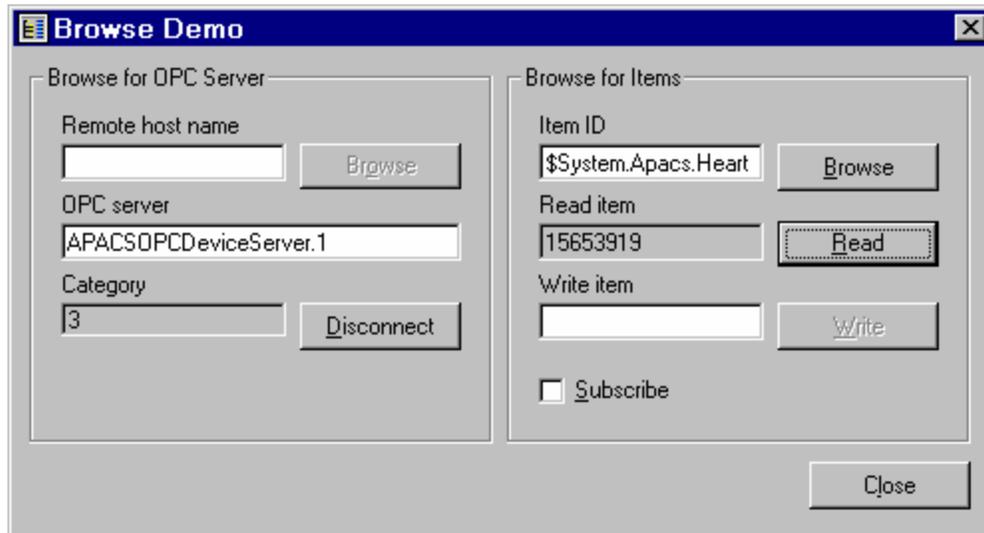


Figure 6-1 Example Client Browse Function

To set up a client Browse function:

1. From the OPC client application, locate and connect to a target OPC Device Server. In the example OPC client, this is done in the left panel. Once this is done, the connection to the OPC Device Server is automatic or you may have to use a Connect/Disconnect button as shown in the example.
2. After a server connection is established, you are free to use the Browse function (in the right panel) to navigate the hierarchical database of APACS+ tags and read/write variable values as permissions or user-selected filtering criteria allow. A browse navigation example is shown in Figure 6-2 and an example of browse filtering is shown in Figure 6-3. The example shown allows you configure the filtering of OPC items according to criteria such as OPC Device Server name.
3. As needed, you can Copy/Paste APACS+ and system tag information from the Browse function into the OPC client application for its initial set up so it can automatically communicate with the OPC Device Server thereafter.

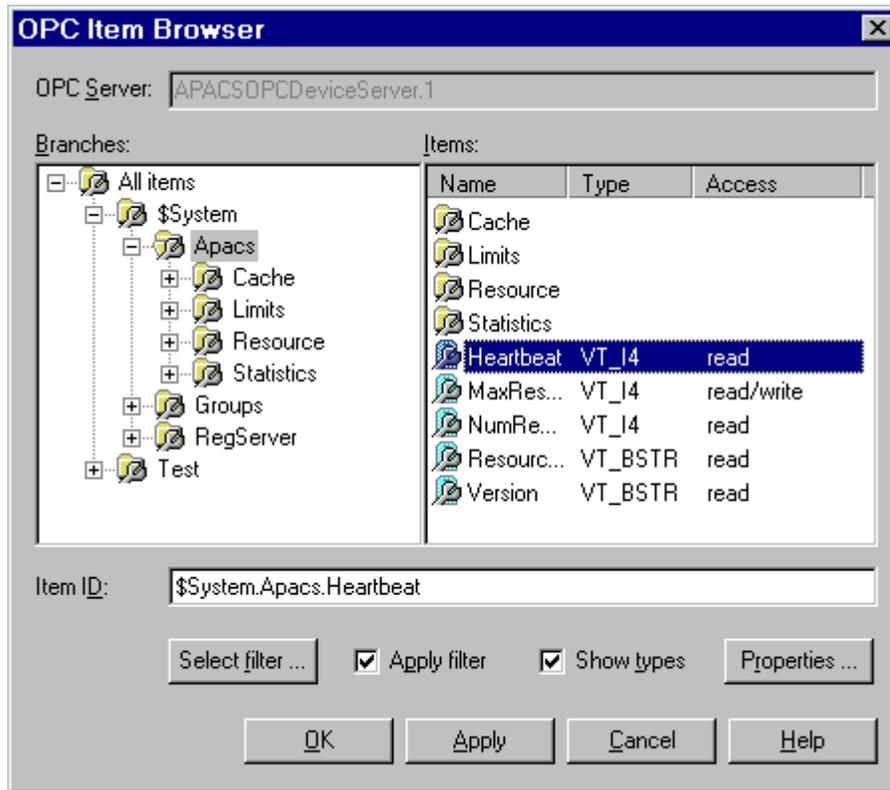


Figure 6-2 Navigating the Browse Database



Figure 6-3 Example of Browse Filtering

#

A.0 APPENDIX A – Error Codes

A.1 Runtime Error Codes Displayed in Control Panel Applet

The runtime error messages listed in Table A-1 are viewable via the control panel applet and saved to the log file. These error codes apply to system or application errors.

Table A-1 Runtime Error Codes

Runtime Error Code	User Action
APACS+ OPC Device Server Aborting: Out of Memory	There are not enough system resources to run the server. Free up some resources for the server
Browse database access error; attempting to reconnect...	<ul style="list-style-type: none"> The browse database was loaded concurrent with an OPC client browse request (try another browse request). The local SQL Server NT service was stopped. (Start the service and try another other browse request) NT/SQL authentication problem. Check SQL Server configuration.
Browse database connection error; attempting to reconnect...	<ul style="list-style-type: none"> The local SQL Server NT service is stopped. Start the local SQL Server NT Service. The browse database is inaccessible or has not been created. Check connection or create new database. The local SQL Server is not installed. Install it.
Browse database connection re-established.	The APACS+ OPC Device Server has successfully re-connected to the SQL Server and its associated browse database. No user action is required.
Could not create APACS+ OPC Device Server Shutdown event <event name>: <error string>	The server may not be able to be shut down correctly. There may not be enough system resources for the program.
Could not Install Service Handler: <error string>	There was a problem with starting this service. The program may need to be reinstalled. If the problem persists, contact TIC.
Couldn't Create Service: <error string>	Installation failure. Make sure you have administrative access.
Couldn't delete Service: <error string>	Uninstall failure. Make sure you have administrative access.
Couldn't open APACS+ OPC Device Server Service: <error string>	There was an installation failure. Reinstall the server.
Couldn't open service manager: <error string> Couldn't query status of installed APACS+ OPC Device Server Service: <error string>	There is a problem with the system. Contact TIC.
Couldn't query status of installed APACS+ OPC Device Server Service: <error string>	There was an installation failure. Reinstall the server.
Couldn't stop installed APACS+ OPC Device Server Service: <error string>	The server is in a corrupt state. Restart the computer. Contact TIC if the problem persists.
Error occurred waiting for Shutdown event: <error string>	The server may not shut down. The PC may need to be rebooted.
Installed Service did not respond to stop request.	The server is in a corrupt state. Restart the computer.

Runtime Error Code	User Action
	Contact TIC if the problem persists.
Scantime (<scan time> = <now>-<next read start>) < readtime (<read time> = <now>-<last read start>	No user action is needed.
Server not properly registered	There was an installation problem. Make sure the server was installed and the account had administrative rights.
ServiceMain thread failed to start	There was an installation failure. Reinstall the server.
ServiceMain thread failed to start, running as Server	The server is running, but not as a service. The computer may need to be rebooted, and the server may need to be reinstalled.
Software License Expired or Invalid, Application cannot be Run	The license for the server is no longer valid. Make sure hard-key is installed. Contact TIC to get a replacement if necessary.
Software license invalid for feature "OPCDeviceServer"	You are not licensed to run the OPC Device server. Contact Siemens to purchase a license.
Unable to Delete Registry Key: <key name>/FileName; Error: <result>	The registry value was not deleted. No user action is required.
Unable to Delete Registry Key: <key name>/MaxFileSize; Error: <result>	The registry value was not deleted. No user action is required.
Unable to Initialize COM Security: <error string>	The COM/DCOM settings may be corrupt. Uninstall and reinstall the server. Contact TIC if the problem persists.
Unable to Initialize COM: <error string>	The COM/DCOM settings may be corrupt. Uninstall and reinstall the server. Contact TIC if the problem persists.
Unable to Read Registry Key: <key name>/FileName; Error: <result>; Using: <file name>	The registry value was not present. Using a default value. No user action is required.
Unable to Read Registry Key: <key name>/MaxFileSize; Error: <result>; Using: <size>	The registry value was not present. Using a default value. No user action is required.
Unable to RegisterCategories: <error string>	Installation problem. Make sure you have administrative rights.
Unable to RegisterClassImplCategories: <error string>	Installation problem. Make sure you have administrative rights.
Unable to RegisterClassObjects with COM: <error string>	The COM/DCOM settings may be corrupt. Uninstall and reinstall the server. Contact TIC if the problem persists.
Unable to RevokeClassObjects with COM: <error string>	The server was unable to shutdown gracefully. Reboot the computer.
Unable to UnRegisterClassImplCategories: <error string>	The server was unable to unregister gracefully. Contact TIC if there are any problems with other OPC servers or clients.
Unable to Write Registry Key: <key name>/FileName; Error: <result>	Unable to update the registry value. A default value will be used next time. Contact TIC if this problem persists.
Unable to Write Registry Key: <key name>/MaxFileSize; Error: <result>	Unable to update the registry value. A default value will be used next time. Contact TIC if this problem persists.

A.2 Extended Error Codes Displayed in Log Files

The extended error messages listed in Table A-2, and those in Table A-1, are saved to log files in the following formats:

Text File Log Format:

OPC Device Server Logging Message Format – Text File Log Device

<Logging Category/Detail>!<date/time of message>!<subsystemID>: <message>

Example message: *PIL!12/19/2000 10:31:17.422!os: APACS+ OPC Device Server: Version 1.00*

- PIL! Category (Program Info); Detail (Low)
- 12/19/2000 10:31:17.422! Date/Time message was generated
- os: OS Subsystem generated the message
- APACS+ OPC Device Server: Version 1.00 Log Message Text

Wonderware Logger Format:

OPC Device Server Logging Message Format – WonderWareLogger Log Device

<date/time of message>/APXOPCDS/<Logging Category/Detail> <subsystemID>: <message>

Example message: *2000/12/20 10:00:39.326/APXOPCDS/PIL opcs: Client_0xcb4050: OPC Client connection established*

- 2000/12/20 10:00:39.326/ Date/Time message was generated
- APXOPCDS/ APACS+ OPC Device Server application identifier
- opcs: opcs subsystem generated this message
- client_0xcb4050: OPC Client connection established Log Message Text

Error Log Legends:

- Run-time Logging Message Category/Detail Legend
 - EAL = Application Error – Low Detail
 - ESL = System Error – Low Detail
 - PIL = Program Info – Low Detail
 - WAL = Application Warning – Low Detail
 - WML = Minor Warning – Low Detail
 - WSL = Subsystem Warning – Low Detail
- Diagnostic Logging Message Category/Detail Legend
 - EAM Application Error – Medium Detail
 - ESM System Error – Medium Detail
 - PIM Program Info – Medium Detail
 - PPL Periodic Info – Low Detail
 - PPM Periodic Info – Medium Detail
 - WAM Application Warning – Medium Detail
 - WMM Minor Warning – Medium Detail
 - WSM Subsystem Warning – Medium Detail

- OPC Device Server Subsystem Identifier Legend
 - apacs = APACS+ Controller Subsystem
 - bbone = Device Server Backbone Subsystem
 - license = Licensing Interface Subsystem
 - opcs = OPC Subsystem
 - os = Operating System Interface Subsystem
 - regs = Registry Server Subsystem
 - stag = System Tag Server Subsystem

Table A-2 Extended Error Codes

Extended Error Code	User Action
Application Warning Messages	
Another instance of the APACS+ OPC Device Server is already running.	No need to run the server, it is already running.
Command Line: <command line utilized to start the APACS+ OPC Device Server>	No user action is needed.
Error Shutting Down APACS API	No user action is needed.
Incorrect Type for Registry Value: <full registry key name> <registry item name>	The registry value is the wrong type. A default value is being used. No user action is required.
Invalid Command Line Option: <command line option in error>	The command line option attempted is not valid. It will be ignored.
Invalid Write discarded <registry item name>	Unable to update the registry value. A default value will be used next time. Contact TIC if this problem persists.
Kernel Named Event Create: <event name> Failed: <error string>	Unable to create the event. There may not be enough system resources to run this program. If a duplicate event error occurs, contact TIC.
License failure, tag <APACS tag name> not created.	This station is not licensed to run the OPC Device Server. Contact Siemens to purchase a license.
Missing Entry for Registry Value: <full registry key name> <registry item name>	The registry value is not present. A default value will be used. No user action is needed.
Registry Key Close: <full key name> Failed: <error string>	No user action is needed.
Registry Key Create: <full key name> Failed: <error string>	Statistical information could not be written to the registry. No user action is needed.
Registry Key Delete Value: <full key name> <value name> Failed: <error string>	Unable to delete a registry value. No user action is needed.
Registry Key Delete: <full key name> <sub key name> Failed: <error string>	Unable to delete a registry value. No user action is needed.
Registry Key Delete: <full key name> Failed: <error string>	Unable to delete a registry value. No user action is needed.

Extended Error Code	User Action
Registry Key Flush: <full key name> Failed: <error string>	Statistical information could not be written to the registry. No user action is required.
Registry Key GetNumSubKeys: <full key name> Failed: <error string>	Registry keys or values may not be present. Default values will be used. No user action is required.
Registry Key GetNumValues: <full key name> Failed: <error string>	Registry keys or values may not be present. Default values will be used. No user action is required.
Registry Key NotifyChange: <full key name> Failed: <error string>	The server cannot respond to configuration changes without restarting. Restart server/computer. If problem persists, contact TIC.
Registry Key Open: <full key name> Failed: <error string>	Unable to open a registry value. Some settings may not be read. Restart the program. Restart the computer. If the problem persists, contact TIC.
Registry Key Read Value: <full key name> <value name> Failed: <error string>; Using a default	The registry value was not present. The default value was used. No user action is needed.
Registry Key Read Value: <full key name> <value name> Failed: <error string>; Using <value>	The registry value was not present. The default value was used. No user action is required.
Registry Key ReadDWORDValue: <full key name> <value name> Failed: <error string>	The registry value was not present. The default value was used. No user action is required.
Registry Key ReadDWORDValue: <full key name> <value name> Failed: Value is not a DWORD	The registry value is the wrong type. A default value is being used. No user action is required.
Registry Key ReadStringValue: <full key name> <value name> Failed: <error string>	The registry value was not present. The default value was used. No user action is required.
Registry Key ReadStringValue: <full key name> <value name> Failed: Value is not a DWORD	The registry value is the wrong type. A default value is being used. No user action is required.
Registry Key ReadValue: <full key name> <value name> Failed: <error string>	The registry value was not present. The default value was used. No user action is required.
Registry Key ReadValue: <full key name> <value name> Failed: Unsupported Type: <type>	The registry value is the wrong type. A default value is being used. No user action is required.
Registry Key ReadValueType: <full key name> <value name> Failed: <error string>	The registry value is the wrong type. A default value is being used. No user action is required.
Registry Key Write Value: <full key name> <value name> Failed: <error string>	Statistical information could not be written to the registry. No user action is required.
Tag not removed: <system tag name>	No user action is required.
Timeout stopping <thread name> Thread, ID=<thread ID>	No user action is required.
Unable to convert to local time: time {High=0x<high part>,Low=0x<low part>}: <error string>	There may be an invalid time specified on one of the systems. Make sure all systems (including APACS) have valid times.
Unable to convert to UTC time: time {High=0x<high part>,Low=0x<low part>}: <error string>	There may be an invalid time specified on one of the systems. Make sure all systems (including APACS) have valid times.

Extended Error Code	User Action
Unable to Start <thread name> Thread: <error string>	There may not be enough resources to run the server. Free up some resources for the server.
Program Information Messages	
ADD succeeded for item "<Opc itemID>"	This is an informational message. No user action is required.
APACS+ OPC Device Server: Version <x.yy>	This is an informational message. No user action is required.
AsyncIO CHANGE callback complete	This is an informational message. No user action is required.
AsyncIO READ callback complete	This is an informational message. No user action is required.
AsyncIO REFRESH callback complete	This is an informational message. No user action is required.
AsyncIO WRITE callback complete	This is an informational message. No user action is required.
AsyncIO2 CANCEL callback succeeded	This is an informational message. No user action is required.
AsyncIO2 CHANGE callback succeeded	This is an informational message. No user action is required.
AsyncIO2 READ callback succeeded	This is an informational message. No user action is required.
AsyncIO2 REFRESH callback succeeded	This is an informational message. No user action is required.
AsyncIO2 WRITE callback succeeded	This is an informational message. No user action is required.
Delaying shutdown for <delay time period> seconds...	This is an informational message. No user action is required.
Group creation successful	This is an informational message. No user action is required.
Group destruction successful	This is an informational message. No user action is required.
Logging configuration has changed.	This is an informational message. No user action is required.
OPC Client connection established	This is an informational message. No user action is required.
OPC Client disconnected from server	This is an informational message. No user action is required.
<Opc Group name> has been orphaned	This is an informational message. No user action is required.
<Opc Group name> has been removed	This is an informational message. No user action is required.
<Opc Group name>: Orphaned	This is an informational message. No user action is required.

Extended Error Code	User Action
<Opc Group name>: Removed	This is an informational message. No user action is required.
Run with Command Line: <application startup full command line>	This is an informational message. No user action is required.
Running as Service	This is an informational message. No user action is required.
Server running	This is an informational message. No user action is required.
Server shut down Completed	This is an informational message. No user action is required.
Server shutting down	This is an informational message. No user action is required.
Server starting	This is an informational message. No user action is required.
Service Status Changed to <NT service state>	This is an informational message. No user action is required.
Service stopped	This is an informational message. No user action is required.
Shutdown event signaled	This is an informational message. No user action is required.
Starting as an NT Service	This is an informational message. No user action is required.
Valid license detected for feature "OPCDeviceServer"	This is an informational message. No user action is required.
Subsystem Warning Messages	
ADD failed for item "<Opc itemID>" - <error Text>	Unable to add the tag. Make sure the name is valid.
ADDGROUP failed - <error text>	If the error isn't memory related, contact TIC.
ADDITEMS transaction failed - <error text>	If the error isn't a bad tag name, or memory related, contact TIC.
AsyncIO CHANGE callback failed - <error text>	If the error isn't a bad tag name, or memory related, contact TIC.
AsyncIO READ transaction failed - <error text>	If the error isn't a bad tag name, or memory related, contact TIC.
AsyncIO REFRESH transaction failed - <error text>	If the error isn't a bad tag name, or memory related, contact TIC.
AsyncIO WRITE transaction failed - <error text>	If the error isn't a bad tag name, or memory related, contact TIC.
AsyncIO(2) CHANGE callbacks halted for this group	The client application connection failed. Restart the client.
AsyncIO2 CANCEL callback failed - <error text>	If the error isn't a bad tag name, or memory related, contact TIC.
AsyncIO2 CHANGE callback failed - <error text>	If the error isn't a bad tag name, or memory related, contact TIC.

Extended Error Code	User Action
AsyncIO2 READ callback failed - <error text>	If the error isn't a bad tag name, or memory related, contact TIC.
AsyncIO2 READ transaction failed - <error text>	If the error isn't a bad tag name, or memory related, contact TIC.
AsyncIO2 REFRESH callback failed - <error text>	If the error isn't a bad tag name, or memory related, contact TIC.
AsyncIO2 REFRESH transaction failed - <error text>	If the error isn't a bad tag name, or memory related, contact TIC.
AsyncIO2 WRITE callback failed - <error text>	If the error isn't a bad tag name, or memory related, contact TIC.
AsyncIO2 WRITE transaction failed - <error text>	If the error isn't a bad tag name, or memory related, contact TIC.
CACHE READ failed for item <itemID> - <error text>	If the error isn't a bad tag name, or memory related, contact TIC.
DEVICE READ failed for item <itemID> - <error text>	If the error isn't a bad tag name, or memory related, contact TIC.
DEVICE WRITE failed for item <itemID> - <error text>	If the error isn't a bad tag name, or memory related, contact TIC.
Group Creation - Unable to determine system time bias	If the error isn't a bad tag name, or memory related, contact TIC.
Group Creation - Unable to start the group update thread	Contact TIC.
SyncIO READ transaction failed - <error text>	If the error isn't a bad tag name, or memory related, contact TIC.
SyncIO WRITE transaction failed - <error text>	If the error isn't a bad tag name, or memory related, contact TIC.



B.0 APPENDIX B – Data Types

For reference, Table B-1 maps APACS+ controller data types to OPC data types. Table B-2 maps OPC Canonical Types to OPC Requested Types that are supported.

Table B-1 APACS+ to OPC Data Types

APACS+ Data Type (DTYPE)	OPC VARTYPE (VT)	Comment
NULL	EMPTY	-
SINT	I2	0 to 255 range enforced on Write
USINT	UI1	-
INT	I2	-
UINT	I4	0 to 65535 range enforced on Write
DINT	I4	-
UDINT	R8	0 to 2 ³² -1 range enforced on Write
BYTE	UI1	-
WORD	I4	0 to 65535 range enforced on Write
DWORD	R8	I4 or R4 would lose precision
TIME	I4	In milliseconds. 1 MS bit of precision lost, 0 to 24.8 hrs
REAL	R4	-
TIMEOFDAY	I4	In milliseconds since midnight (UTC). 1 MS bit of precision lost, 0 to 24.8 hrs
DATE	DATE	Time shown as 0:00:00. Date range 1970 to 2038.
DATETIME	DATE	UTC time. Date range 1970 to 2038
ERROR_SUMMARY	I4	Conversion to signed value loses 1 MS bit of precision. Values > 32767 shown as negative.
LREAL	R8	-
STRING	BSTR	254 character limit, not including single quotes. Single quotes removed on Read (added on Write) in Device Server.
LINT	R8	Loses 4 decimal digits of precision.
ULINT	R8	Loses 4 decimal digits of precision.
LWORD	R8	Loses 4 decimal digits of precision.
BYTEARRAY	EMPTY	Not supported
ARRAY	EMPTY	Not supported
BOOL	BOOL	-
other	EMPTY	Not Expected

Table B-2 OPC Canonical Types to OPC Requested Types

FROM TYPE	TO TYPE																			
	VT_EMPTY	VT_NULL	VT_UI1	VT_I2	VT_I4	VT_R4	VT_R8	VT_I8	VT_CY	VT_DATE	VT_BSTR	VT_DISPATCH	VT_ERROR	VT_BOOL	VT_VARIANT	VT_UNKNOWN	VT_RESERVED	VT_BYREF	VT_ARRAY	
VT_EMPTY																				
VT_NULL																				
VT_UI1			V	V	V	V	V				V			V						
VT_I2			R	V	V	V	V				V			V						
VT_I4			R	R	V	V	V			RU	V			V						
VT_R4			RU	RU	RU	V	V			V	V			U						
VT_R8			RU	RU	RU	R	V			V	V			U						
VT_I8																				
VT_CY																				
VT_DATE					RU	R	V			V	V									
VT_BSTR			R	R	R	R	R			R	V			R						
VT_DISPATCH																				
VT_ERROR																				
VT_BOOL			V	V	V	V	V				V			V						
VT_VARIANT																				
VT_UNKNOWN																				
VT_RESERVED																				
VT_BYREF																				
VT_ARRAY																				

Legend

- V = Quality unconditionally GOOD - Converts without data loss
- U = Quality unconditionally UNCERTAIN - Conversion changes meaning of value
- R = Quality GOOD if passes range check, otherwise BAD
(returns HRESULT = DISP_E_OVERFLOW if range check fails)
- RU = Quality UNCERTAIN if passes range check, otherwise BAD
- (Blank) = Quality BAD – Not Convertible

Microsoft's ::VariantChangeType() conversion rules:

1. Reals (float and double) will always convert to a VT_BOOL
(|RealNum| > .5 is TRUE)(|RealNum| <= .5 is FALSE)
2. Reals (float and double) will convert to an integer type by first rounding (x<=.5 rounds down), truncating the fractional part, then performing a range check of the destination type. If range check passes, conversion is successful.
3. Integers will convert to a smaller integral type as long as the result passes a range check of the destination type.
4. If result fails range check of destination type, conversion aborts with HRESULT = DISP_E_OVERFLOW.
5. If src variant is a BSTR and is being converted to a numerical type, conversion will fail if the BSTR contains an invalid numerical digit or punctuation and the HRESULT returned is DISP_E_TYEMISMATCH.
6. The type VT_I8 is not supported in the VC++ 5.0 VARIANT definition and was removed from all the translation tables.

C.0 APPENDIX C – System Tags

This section provides several tables that present the system tags. All system tags are available to be configured in the OPC client using the System Tag Name listed in the tables, as the Item ID in the client.

- Table C-1 - Basic System Tags
- Table C-2 - APACS+ System Tags
- Table C-3 - APACS+ System Tags Per Controller
- Table C-4 - OPC System Tags

Basic System Tags

Table C-1 lists Basic System Tags. They report status information about the overall product and the Registry System Tag interface.

Table C-1 Basic System Tags

System Tag Name	Type	RW	Range	Default	Description
\$System.Version	BSTR	R		1.00	Version number of the product, major.minor.service_pack.build
\$System.RegServer.Limits.OverRunDelay	I4	RW		1000	Number of milliseconds to delay after registry update which overran, before attempting to start the next update. This parameter can be used to prevent an excessively slow registry from soaking the OPC Device Server capacity.
\$System.RegServer.Statistics.LastLoopTime	I4	R		0	Number of milliseconds spent synchronizing the registry, for the most recent registry key which was updated" This does not include delay time between updates.
\$System.RegServer.Statistics.MinLoopTime	I4	RW	mL-ML	R(LastLoopTime)	Min LastLoopTime
\$System.RegServer.Statistics.MaxLoopTime	I4	RW	mL-ML	R(LastLoopTime)	Max LastLoopTime
\$System.RegServer.Statistics.TotalOverRuns	I4	RW	mL-ML	R(0)	Count of number of times any group had a LastLoopTime which exceeded the scan rate for the group.
\$System.RegServer.NumItems	I4	R	0-ML	NA	Number of registry Server Item instances within the Atlas product (ie. number of Items within all the ItemValues registry keys)
\$System.RegServer.MaxItems	I4	RW	ml-ML	R(NumItems)	Max RegServer.NumItems

Key:

R = Readable

W = Writeable

ML = Max Long, $(2^{31})-1$

mL = Min Long , $-(2^{31})$

NA = not applicable

R(#) in Default column indicates an initial value as well as what the variable is reset to via \$System.ResetStatistics

APACS+ System Tags

Table C-2 lists APACS+ System Tags. These tags provide current operational counters, limits, and statistics about the APACS+ subsystem within each Siemens OPC product.

Table C-2 APACS+ System Tags

System Tag Name	CPA	Type	RW	Range	Default	Description
\$\$System.Apacs.Heartbeat	Y	I4	R	ml-ML	NA	This tag is incremented (at approximately the PollDelay rate) to reflect that the server is still operational and that the APACS+ API is not hung. Clients can monitor this tag is changing to determine that the server is still working.
\$\$System.APACS+.Limits.TagListLength	Y	I4	RW	1-1000	50	Sets the maximum number of tags per APACS+ API Taglist.
\$\$System.Apacs.Limits.ForceTagListCompare	Y	I4	RW	0-ML	10	The number of scans of a taglist after which a full compare of all tags on the taglist is forced (ie. change list is ignored). If set to 0, all scans do a full compare.
\$\$System.Apacs.Limits.CombineRead	Y	I4	RW	0-1000	3	Minimum number of single tag reads outstanding at once to cause a taglist to be used to read them, as opposed to doing single tag reads. Setting to 0 disables the use of taglists for single tag reads. Setting to 1 will cause single tag reads to always use taglists.
\$\$System.Apacs.Limits.PollLoopTime	Y	I4	RW	100-10000	500	Maximum time in milliseconds expected for a good Exec Loop. When a loop exceeds this limit, TotalLongPollLoops is incremented.
\$\$System.Apacs.Limits.PollDelay	Y	I4	RW	10-100	10	Number of milliseconds to delay between API polling loops
\$\$System.Apacs.Limits.OperationsActive	Y	I4	RW	1-1000	100	Maximum total of Reads, ReadLists and Writes to initiate within the APACS+ API at once.
\$\$System.Apacs.Limits.ReadsActive	Y	I4	RW	1-1000	10	Maximum total single tag reads to initiate within the APACS+ API at once.
\$\$System.Apacs.Limits.ReadListsActive	Y	I4	RW	1-1000	100	Maximum total taglist reads to initiate within the APACS+ API at once.

System Tag Name	C P A	Type	RW	Range	Default	Description
\$System.Apacs.Limits.WritesActive	Y	I4	RW	1-1000	10	Maximum total single tag writes to initiate within the APACS+ API at once
\$System.Apacs.Resource.Limits.OperationsActive	Y	I4	RW	1-1000	10	Maximum total of Reads, ReadLists and Writes to initiate against any one APACS+ resource at once. If this 1, only a single operation will be done against the resource at once. This prevents write's from overlapping with reads and prevents single tag operations from overlapping with taglist operations, however overall throughput of the OPC Device Server may be reduced.
\$System.Apacs.Resource.Limits.ReadsActive	Y	I4	RW	1-1000	3	Maximum total single tag reads to initiate against any one APACS+ resource at once.
\$System.Apacs.Resource.Limits.ReadListsActive	Y	I4	RW	1-1000	10	Maximum total taglist reads to initiate against any one APACS+ resource at once. If this is 1, only a single taglists will be read at once against a resource, reducing load on the resource and causing "taglist staggering", however overall throughput of the OPC Device Server may be reduced
\$System.Apacs.Resource.Limits.WritesActive	Y	I4	RW	1-1000	3	Maximum total single tag writes to initiate against any one APACS+ resource at once.

System Tag Name	CPA	Type	RW	Range	Default	Description
\$\$System.Apacs.ResourceNames	N	STR	R	NA	NA	A comma separated list of all APACS+ Resources which the OPC Device Server has been used to access. (this includes resources requested by clients which failed to be resolved). Note: even once all tags on a given resource are made inactive, the resource still remains in this list. This is necessary since this list also reflects the resource names available for resource level tuning and statistics. Hence tuning parameters set when a resource is first access will not be lost as tags go inactive and are re-activated for the resource.
\$\$System.Apacs.NumResources	Y	I4	R	0-ML	NA	Number of resources in ResourceNames
\$\$System.Apacs.MaxResources	Y	I4	RW	ml-ML	R(NumResources)	Max NumResources
\$\$System.Apacs.Statistics.SuccessRate	Y	I4	R	0-100	R(100)	Out of the last 100 operations, how many were successful (if < 100 tag accesses has been performed since OPC Device Server startup, shows 100-number of failures to date).
\$\$System.Apacs.Statistics.TotalOperations	Y	I4	RW	ml-ML	R(0)	Cumulative total tag reads, writes and taglist reads completed (success and failure)
\$\$System.Apacs.Statistics.TotalReads	Y	I4	RW	ml-ML	R(0)	Cumulative total single tag reads completed (success and failure)
\$\$System.Apacs.Statistics.TotalReadLists	Y	I4	RW	ml-ML	R(0)	Cumulative total taglist reads completed (success and failure)
\$\$System.Apacs.Statistics.TotalWrites	Y	I4	RW	ml-ML	R(0)	Cumulative total single tag writes completed (success and failure)
\$\$System.Apacs.Statistics.TotalFailedOperations	Y	I4	RW	ml-ML	R(0)	Cumulative total tag reads, writes and taglist reads completed with failure
\$\$System.Apacs.Statistics.TotalFailedReads	Y	I4	RW	ml-ML	R(0)	Cumulative total single tag reads completed with failure
\$\$System.Apacs.Statistics.TotalFailedReadLists	Y	I4	RW	ml-ML	R(0)	Cumulative total taglist reads completed with failure
\$\$System.Apacs.Statistics.TotalFailedWrites	Y	I4	RW	ml-ML	R(0)	Cumulative total single tag writes completed with failure

System Tag Name	CPA	Type	RW	Range	Default	Description
\$\$System.Apacs.Statistics.NumOperationsActive	Y	I4	R	0-ML	NA	Total Number of reads/readlists and writes presently in progress
\$\$System.Apacs.Statistics.NumReadsActive	Y	I4	R	0-ML	NA	Number of Single Tag reads presently in progress
\$\$System.Apacs.Statistics.NumReadListsActive	Y	I4	R	0-ML	NA	Number of Taglist reads presently in progress
\$\$System.Apacs.Statistics.NumWritesActive	Y	I4	R	0-ML	NA	Number of Single Tag writes presently in progress
\$\$System.Apacs.Statistics.MaxOperationsActive	N	I4	RW	ml-ML	R(NumOperationsActive)	Max NumOperationsActive
\$\$System.Apacs.Statistics.MaxReadsActive	N	I4	RW	ml-ML	R(NumReadsActive)	Max NumReadsActive
\$\$System.Apacs.Statistics.MaxReadListsActive	N	I4	RW	ml-ML	R(NumReadListsActive)	Max NumReadListsActive
\$\$System.Apacs.Statistics.MaxWritesActive	N	I4	RW	ml-ML	R(NumWritesActive)	Max NumWritesActive
\$\$System.Apacs.Statistics.NumOperationsQueued	Y	I4	R	0-ML	NA	Total Number of resources presently queued to read, readlist or write. (Note this count represents resources needing to do work, not number of tags/taglists needing to be read/written).
\$\$System.Apacs.Statistics.NumReadsQueued	Y	I4	R	0-ML	NA	Total Number of resources presently queued to do single tag reads. (Note this count represents resources needing to do work, not number of tags needing to be read).
\$\$System.Apacs.Statistics.NumReadListsQueued	Y	I4	R	0-ML	NA	Total Number of resources presently queued to do taglist reads. (Note this count represents resources needing to do work, not number of taglists needing to be read).
\$\$System.Apacs.Statistics.NumWritesQueued	Y	I4	R	0-ML	NA	Total Number of resources presently queued to do single tag writes. (Note this count represents resources needing to do work, not number of tags needing to be written).
\$\$System.Apacs.Statistics.MaxOperationsQueued	N	I4	RW	ml-ML	R(NumOperationsQueued)	Max NumOperationsQueued
\$\$System.Apacs.Statistics.MaxReadsQueued	N	I4	RW	ml-ML	R(NumReadsQueued)	Max NumReadsQueued
\$\$System.Apacs.Statistics.MaxReadListsQueued	N	I4	RW	ml-ML	R(NumReadListsQueued)	Max NumReadListsQueued
\$\$System.Apacs.Statistics.MaxWritesQueued	N	I4	RW	ml-ML	R(NumWritesQueued)	Max NumWritesQueued

System Tag Name	CPA	Type	RW	Range	Default	Description
\$\$System.Apacs.Statistics.TotalPollLoops		I4	RW	ml-ML	R(0)	Count of number of executed thread loops through polling all active controller operations.
\$\$System.Apacs.Statistics.LastPollLoopTime	Y	I4	R	0-ML	NA	Most recent Loop time observed for exec thread loop, in milliseconds
\$\$System.Apacs.Statistics.MaxPollLoopTime	Y	I4	RW	ml-ML	R(LastPollLoopTime)	Max LastPollLoopTime
\$\$System.Apacs.Statistics.MinPollLoopTime	N	I4	RW	ml-ML	R(MinPollLoopTime)	Min LastPollLoopTime
\$\$System.Apacs.Statistics.TotalLongPollLoops	Y	I4	RW	ml-ML	R(0)	Count of number of loops which exceeded Limits.PollLoopTime to execute, indicating excessive API delays, API problems or unexpected resolving of tags synchronously in Async calls.
\$\$System.Apacs.Statistics.NumTags	N	I4	R	0-ML	NA	Number of APACS+ API Data Tags allocated.
\$\$System.Apacs.Statistics.MaxItems	N	I4	RW	ml-ML	R(NumItems)	Max NumTags
\$\$System.Apacs.Statistics.NumTagLists	Y	I4	R	0-ML	NA	Number of APACS+ API Data Tag lists allocated (including the 3 used for the tag cache (includes temporary taglists created to optimize groups of reads)
\$\$System.Apacs.Statistics.MaxTagLists	N	I4	RW	ml-ML	R(NumTagLists)	Max NumTagLists
\$\$System.Apacs.Statistics.TotalOverRuns	N	I4	RW	ml-ML	R(0)	Number of taglists which have exceeded their scan rate (could partially be due to delays in starting the scan due to limits on active operations)
\$\$System.Apacs.Statistics.LastScanTime		I4	R	0-ML	0	Most recent Scan time observed for a taglist, in milliseconds. Scantime is the total from the scheduled start of the taglist read to the completion of the read
\$\$System.Apacs.Statistics.MaxScanTime		I4	RW	ml-ML	R(LastScanTime)	Max LastScanTime
\$\$System.Apacs.Statistics.MinScanTime		I4	RW	ml-ML	R(LastScanTime)	Min LastScanTime
\$\$System.Apacs.Statistics.LastReadListTime		I4	R	0-ML	0	Most recent read time observed for a taglist in milliseconds. Read time is the time from the actual start of the taglist read to the completion of the read
\$\$System.Apacs.Statistics.MaxReadListTime		I4	RW	ml-ML	R(LastReadListTime)	Max LastReadListTime
\$\$System.Apacs.Statistics.MinReadListTime		I4	RW	ml-ML	R(LastReadListTime)	Min LastReadListTime

System Tag Name	CPA	Type	RW	Range	Default	Description
\$\$System.Apacs.Cache.AgingRate	Y	I4	RW	2000-ML	12*60*60*1000 (12 hours)	Rate at which tags should be aged, in milliseconds. The operation of cache aging is a 2 step process. Hence tags will stay in the cache for AgingRate to 2*AgingRate once they are not in use anymore. Any changes to the aging rate will take effect within 1 second of the change, even if the change is a large increase or decrease in the aging rate. This is useful, especially if the aging rate is being decreased significantly from the default of 12 hours.
\$\$System.Apacs.Cache.NumTags	Y	I4	R	0-ML	NA	Total of NumActiveTags, NumRecentInactiveTags and NumLongInactiveTags.
\$\$System.Apacs.Cache.MaxTags		I4	RW	ml-ML	R(NumTags)	Max NumTags
\$\$System.Apacs.Cache.NumActiveTags	Y	I4	R	0-ML	NA	Number of tags presently active (being scanned or having a single tag read/write done, ie. associated with an ActiveItem).
\$\$System.Apacs.Cache.MaxActiveTags		I4	RW	ml-ML	R(NumActiveTags)	Max NumTags
\$\$System.Apacs.Cache.NumRecentInactiveTags	Y	I4	R	0-ML	NA	The number of tags which have recently (<m_AgingRate) gone inactive. Tags in this category are subject to move to the LongInactive status on the next AgingRate loop of the cache aging thread.
\$\$System.Apacs.Cache.MaxRecentInactiveTags		I4	RW	ml-ML	R(NumRecentInactiveTags)	Max NumTags
\$\$System.Apacs.Cache.NumLongInactiveTags	Y	I4	R	0-ML	NA	The number of tags which have been inactive for a long time (0 to m_AgingRate. Tags in this category are subject to be removed from the cache on the next AgingRate loop of the cache aging thread.
\$\$System.Apacs.Cache.MaxLongInactiveTags		I4	RW	ml-ML	R(NumLongInactiveTags)	Max NumTags
\$\$System.Apacs.Cache.TotalLookups	Y	I4	RW	ml-ML	R(0)	The running total of lookups performed against the cache
\$\$System.Apacs.Cache.TotalHits	Y	I4	RW	ml-ML	R(0)	The running total of lookups which found the desired tag in the cache

System Tag Name	C P A	Type	RW	Range	Default	Description
\$System.Apacs.Version		STR	R	NA	NA	Version string of the APACS+ API exactly as reported by the API. Only versions 4.35 or higher are supported.

Key:

R = Readable

W = Writeable

ML = Max Long, $(2^{31})-1$ mL = Min Long, $-(2^{31})$

NA = not applicable

CPA = Viewable in Control Panel Applet

R(#) in Default column indicates an initial value as well as what the variable is reset to via \$System.ResetStatistics

APACS+ System Tags Per Controller

Table C-3 lists APACS+ System Tags Per Controller. The controller-specific APACS+ system tags are a set of tags that only exist when an APACS+ controller is presently being scanned by the Siemens OPC product. One complete set of the controller-specific system tags exists for every controller presently being scanned by the OPC Device Server). The tag name syntax for each set of controller-specific system tags is:

\$System.Apacs.RESOURCE.X

Where: RESOURCE is a specific APACS+ controller name

Table C-3 APACS+ System Tags Per Controller

System Tag Name	Type	RW	Range	Default	Description
\$System.Apacs.CTRL2.Statistics.Rate	I4	R	0-100	R(100)	Out of the last 100 operations, how many were successful (if < 100 tag accesses has been performed since OPC Device Server startup, shows 100-number of failures to date).
\$System.Apacs.CTRL2.Statistics.TotalOperations	I4	RW	ml-ML	R(0)	Cumulative total tag reads, writes and taglist reads completed (success and failure) against the resource
\$System.Apacs.CTRL2.Statistics.TotalReads	I4	RW	ml-ML	R(0)	Cumulative total single tag reads completed (success and failure) against the resource
\$System.Apacs.CTRL2.Statistics.TotalReadLists	I4	RW	ml-ML	R(0)	Cumulative total taglist reads completed (success and failure) against the resource
\$System.Apacs.CTRL2.Statistics.TotalWrites	I4	RW	ml-ML	R(0)	Cumulative total single tag writes completed (success and failure) against the resource
\$System.Apacs.CTRL2.Statistics.TotalFailedOperations	I4	RW	ml-ML	R(0)	Cumulative total tag reads, writes and taglist reads completed with failure against the resource
\$System.Apacs.CTRL2.Statistics.TotalFailedReads	I4	RW	ml-ML	R(0)	Cumulative total single tag reads completed with failure against the resource
\$System.Apacs.CTRL2.Statistics.TotalFailedReadLists	I4	RW	ml-ML	R(0)	Cumulative total taglist reads completed with failure against the resource
\$System.Apacs.CTRL2.Statistics.TotalFailedWrites	I4	RW	ml-ML	R(0)	Cumulative total single tag writes completed with failure against the resource

System Tag Name	Type	RW	Range	Default	Description
\$\$System.Apacs.CTRL2.Statistics.NumOperationsActive	I4	R	0-ML	NA	Total Number of reads/readlists and writes presently in progress against the resource
\$\$System.Apacs.CTRL2.Statistics.NumReadsActive	I4	R	0-ML	NA	Number of Single Tag reads presently in progress against the resource
\$\$System.Apacs.CTRL2.Statistics.NumReadListsActive	I4	R	0-ML	NA	Number of Taglist reads presently in progress against the resource
\$\$System.Apacs.CTRL2.Statistics.NumWritesActive	I4	R	0-ML	NA	Number of Single Tag writes presently in progress against the resource
\$\$System.Apacs.CTRL2.Statistics.MaxOperationsActive	I4	RW	ml-ML	R(NumOperationsActive)	Max NumOperationsActive
\$\$System.Apacs.CTRL2.Statistics.MaxReadsActive	I4	RW	ml-ML	R(NumReadsActive)	Max NumReadsActive
\$\$System.Apacs.CTRL2.Statistics.MaxReadListsActive	I4	RW	ml-ML	R(NumReadListsActive)	Max NumReadListsActive
\$\$System.Apacs.CTRL2.Statistics.MaxWritesActive	I4	RW	ml-ML	R(NumWritesActive)	Max NumWritesActive
\$\$System.Apacs.CTRL2.Statistics.NumOperationsQueued	I4	R	0-ML	NA	Total Number of reads, readlists or writes presently queued against the resource
\$\$System.Apacs.CTRL2.Statistics.NumReadsQueued	I4	R	0-ML	NA	Total Number of single tag reads presently queued against the resource
\$\$System.Apacs.CTRL2.Statistics.NumReadListsQueued	I4	R	0-ML	NA	Total Number of taglist reads presently queued against the resource
\$\$System.Apacs.CTRL2.Statistics.NumWritesQueued	I4	R	0-ML	NA	Total Number of single tag writes presently queued against the resource
\$\$System.Apacs.CTRL2.Statistics.MaxOperationsQueued	I4	RW	ml-ML	R(NumOperationsQueued)	Max NumOperationsQueued
\$\$System.Apacs.CTRL2.Statistics.MaxReadsQueued	I4	RW	ml-ML	R(NumReadsQueued)	Max NumReadsQueued
\$\$System.Apacs.CTRL2.Statistics.MaxReadListsQueued	I4	RW	ml-ML	R(NumReadListsQueued)	Max NumReadListsQueued
\$\$System.Apacs.CTRL2.Statistics.MaxWritesQueued	I4	RW	ml-ML	R(NumWritesQueued)	Max NumWritesQueued
\$\$System.Apacs.CTRL2.Statistics.NumItems	I4	R	0-ML	NA	Number of items active in the resource.
\$\$System.Apacs.CTRL2.Statistics.MaxItems	I4	RW	ml-ML	R(NumItems)	Max NumItems
\$\$System.Apacs.CTRL2.Statistics.NumTagLists	I4	R	0-ML	NA	Number of taglists active in the resource (includes temporary taglists created to optimize groups of reads)
\$\$System.Apacs.CTRL2.Statistics.MaxTagLists	I4	RW	ml-ML	R(NumTagLists)	Max NumTagLists

System Tag Name	Type	RW	Range	Default	Description
\$System.Apacs.CTRL2.Statistics.TotalOverRuns	I4	RW	mL-ML	R(0)	Number of taglists which have exceeded their scan rate (could partially be due to delays in starting the scan due to limits on active operations)
\$System.Apacs.CTRL2.Statistics.LastScanTime	I4	R	0-ML	0	Most recent Scan time observed for a taglist, in milliseconds. Scantime is the total from the scheduled start of the taglist read to the completion of the read
\$System.Apacs.CTRL2.Statistics.MaxScanTime	I4	RW	mL-ML	R(LastScanTime)	Max LastScanTime
\$System.Apacs.CTRL2.Statistics.MinScanTime	I4	RW	mL-ML	R(LastScanTime)	Min LastScanTime
\$System.Apacs.CTRL2.Statistics.LastReadListTime	I4	R	0-ML	0	Most recent read time observed for a taglist in milliseconds. Read time is the time from the actual start of the taglist read to the completion of the read
\$System.Apacs.CTRL2.Statistics.MaxReadListTime	I4	RW	mL-ML	R(LastReadListTime)	Max LastReadListTime
\$System.Apacs.CTRL2.Statistics.MinReadListTime	I4	RW	mL-ML	R(LastReadListTime)	Min LastReadListTime

Key:

R = Readable

W = Writeable

ML = Max Long, $(2^{31})-1$ mL = Min Long, $-(2^{31})$

NA = not applicable

R(#) in Default column indicates an initial value as well as what the variable is reset to via \$System.ResetStatistics

OPC System Tags

Table C-4 lists OPC System Tags. These tags provide current operational counters, limits, and statistics about the OPC subsystem within each OPC Device Server.

Table C-4 OPC System Tags

System Tag Name	CPA	Type	RW	Range	Default	Description
\$\$System.NumServers	Y	I4	R	0-ML	NA	Number of connected OPC clients.
\$\$System.MaxServers		I4	RW	mL - ML	R(0)	Max NumServers.
\$\$System.ClientNames		STR	R	NA	NA	A comma separated list of all the OPC Client Names which are using the OPC Device Server. For OPC 1.0 Clients and OPC 2.0 Clients which have not used the SetClientName OPC interface, the name will be shown as a hex value (the server handle for the interface). In the event that client names have included comas, the parsing of this into individual names may be ambiguous. It is recommended that client names not use comas.
\$\$System.NumGroups	Y	I4	R	0-ML	NA	Number of OPC Group instances within the OPC Device Server (i.e., number of groups within all clients currently connected.)
\$\$System.MaxGroups		I4	RW	mL-ML	R(0)	Max NumGroups
\$\$System.NumEnumerators	Y	I4	R	0-ML	NA	Number of OPC Group/Item enumerator instances within the OPC Device Server (i.e., number of group/item enumerators within all clients currently connected)
\$\$System.MaxEnumerators		I4	RW	mL-ML	R(0)	Max NumEnumerators

System Tag Name	CPA	Type	RW	Range	Default	Description
\$System.Groups.Limits.AsyncWritesActive	N	14	RW	1-1000	25	Maximum total AsyncWrite transactions allowed to be active at the same time. Upon reaching this limit, subsequent AsyncWrite transactions will be queued. Recommend setting this limit to be 5 more than the total number of OPC groups that will request writes at the same time.
\$System.Groups.Limits.AsyncWritesQueued	N	14	RW	1-1000	400	Maximum total AsyncWrite transactions allowed to be queued before rejecting any subsequent AsyncWrite transactions. Recommend setting this limit to be the total number of OPC groups that will request AsyncWrites at the same time multiplied by 15-20 secs of backlog queuing.
\$System.Groups.AsyncLimitsArePerGroup	N	BOOL	RW	On/Off	Off	Treats the "\$System.Groups.Limits.AsyncWritesActive" and "\$System.Groups.Limits.AsyncWritesQueued" tuning variables as limits across all OPC groups if Off or limits per OPC group if On. AsyncReads and AsyncRefreshes are not affected by the value of this tuning variable.
\$System.Groups.RefreshCacheAfterWrite	N	BOOL	RW	On/Off	Off	Off means that an Apacs tag READ will not occur immediately following each tag WRITE in an AsyncWrite transaction leaving the server's cached value stale until the next scheduled scan cycle. On will complete an Apacs tag READ immediately after each tag WRITE in an AsyncWrite transaction. Best tag WRITE throughput is realized when this value is Off.

System Tag Name	CPA	Type	RW	Range	Default	Description
\$\$System.Groups.Limits.OverRunDelay	Y	I4	RW		100	Number of milliseconds to delay after an active group advise which overran, before attempting to start the next advise. This parameter can be used to prevent an excessively large group or an overloaded OPC client from consuming the OPC Device Server capacity.
\$\$System.Groups.Statistics.LastLoopTime	Y	I4	R		0	Number of milliseconds spent issuing the advise to a OPC client, for the most recent client group which was "advised". This does not include delay time between scans.
\$\$System.Groups.Statistics.MinLoopTime		I4	RW	mL-ML	R(LastLoopTime)	Min LastLoopTime
\$\$System.Groups.Statistics.MaxLoopTime		I4	RW	mL-ML	R(LastLoopTime)	Max LastLoopTime
\$\$System.Groups.Statistics.NumAsyncWrites Active	N	14	R	0-ML	0	Instantaneous count of AsyncWrite transactions that are currently active. This count is across all OPC groups and all OPC client connections, regardless of the value for "\$System.Groups.AsyncLimitsArePerGroup". The value will be less than or equal to "\$System.Groups.Limits.AsyncWritesActive" only when the value for "\$System.Groups.AsyncLimitsArePerGroup" is Off.
\$\$System.Groups.Statistics.NumAsyncWrites Queued	N	14	R	0-ML	0	Instantaneous count of AsyncWrite transactions that are currently queued. This count is across all OPC groups and all OPC client connections The value will be less than or equal to "\$System.Groups.Limits.AsyncWritesQueued" only when the value for "\$System.Groups.AsyncLimitsArePerGroup" is Off.

System Tag Name	CPA	Type	RW	Range	Default	Description
\$\$System.Groups.Statistics.NumAsyncReadsActive	N	14	R	0-ML	0	Instantaneous count of AsyncRead transactions that are currently active. This count is across all OPC groups and all OPC client connections, regardless of the value for "\$System.Groups.AsyncLimitsArePerGroup".
\$\$System.Groups.Statistics.NumAsyncRefreshesActive	N	14	R	0-ML	0	Instantaneous count of AsyncRefresh transactions that are currently active. This count is across all OPC groups and all OPC client connections, regardless of the value for "\$System.Groups.AsyncLimitsArePerGroup".
\$\$System.Groups.Statistics.NumAsyncOperationsActive	N	14	R	0-ML	0	Instantaneous count of all Async transactions (READ/REFRESH/WRITE) that are currently active. This count is across all OPC groups and all OPC client connections, regardless of the value for "\$System.Groups.AsyncLimitsArePerGroup".
\$\$System.Groups.Statistics.MaxAsyncWritesActive	N	14	RW	ml-ML	R(NumAsyncWritesActive)	Max NumAsyncWritesActive
\$\$System.Groups.Statistics.MaxAsyncWritesQueued	N	14	RW	ml-ML	R(NumAsyncWritesQueued)	Max NumAsyncWritesQueued
\$\$System.Groups.Statistics.MaxAsyncReadsActive	N	14	RW	ml-ML	R(NumAsyncReadsActive)	Max NumAsyncReadsActive
\$\$System.Groups.Statistics.MaxAsyncRefreshesActive	N	14	RW	ml-ML	R(NumAsyncRefreshesActive)	Max NumAsyncRefreshesActive
\$\$System.Groups.Statistics.MaxAsyncOperationsActive	N	14	RW	ml-ML	R(NumAsyncOperationsActive)	Max NumAsyncOperationsActive

System Tag Name	CPA	Type	RW	Range	Default	Description
\$System.Groups.Statistics.TotalCallbacks	Y	14	RW	ml-ML	R(0)	This is the total number of attempted OPC Data Access 2.0x data callbacks that have occurred between the server and all connected OPC clients. This total is inclusive for all connected OPC clients across all types of Data Access 2.0x data callbacks (READ/WRITE/REFRESH/CHANGE/CANCEL). This total can be reset to get statistical information about the server (i.e., the number of callbacks over time.).
\$System.Groups.Statistics.TotalCancelCallbacks	N	14	RW	ml-ML	R(0)	This is the total number of attempted OPC Data Access 2.0x CANCEL data callbacks that have occurred between the server and all connected OPC clients. This total is inclusive for all connected OPC clients.
\$System.Groups.Statistics.TotalChange Callbacks	N	14	RW	ml-ML	R(0)	This is the total number of attempted OPC Data Access 2.0x CHANGE data callbacks that have occurred between the server and all connected OPC clients. This total is inclusive for all connected OPC clients.
\$System.Groups.Statistics.TotalFailedCallbacks	Y	14	RW	ml-ML	R(0)	This is the total number of attempted OPC Data Access 2.0x data callbacks that have failed between the server and all connected OPC clients. This total is inclusive for all connected OPC clients across all types of Data Access 2.0x data callbacks (READ/WRITE/REFRESH/CHANGE/CANCEL). The difference between this total and the "Cumulative Callbacks" total represents the total number of successful callback attempts. This total is can be reset in order to get statistical information about the server (i.e., the number of callbacks that failed over time.).

System Tag Name	CPA	Type	RW	Range	Default	Description
\$System.Groups.Statistics.TotalFailedCancelCallbacks	N	14	RW	ml-ML	R(0)	This is the total number of attempted OPC Data Access 2.0x CANCEL data callbacks that have failed between the server and all connected OPC clients. This total is inclusive for all connected OPC clients. The difference between this value and the "TotalCancelCallbacks" value represents the total number of successful CANCEL callbacks.
\$System.Groups.Statistics.TotalFailedChangeCallbacks	N	14	RW	ml-ML	R(0)	This is the total number of attempted OPC Data Access 2.0x CHANGE data callbacks that have failed between the server and all connected OPC clients. This total is inclusive for all connected OPC clients. The difference between this value and the "TotalChangeCallbacks" value represents the total number of successful CHANGE callbacks.
\$System.Groups.Statistics.TotalFailedReadCallbacks	N	14	RW	ml-ML	R(0)	This is the total number of attempted OPC Data Access 2.0x READ data callbacks that have failed between the server and all connected OPC clients. This total is inclusive for all connected OPC clients. The difference between this value and the "TotalReadCallbacks" value represents the total number of successful READ callbacks.
\$System.Groups.Statistics.TotalFailedRefreshCallbacks	N	14	RW	ml-ML	R(0)	This is the total number of attempted OPC Data Access 2.0x REFRESH data callbacks that have failed between the server and all connected OPC clients. This total is inclusive for all connected OPC clients. The difference between this value and the "TotalRefreshCallbacks" value represents the total number of successful REFRESH callbacks.

System Tag Name	CPA	Type	RW	Range	Default	Description
\$System.Groups.Statistics.TotalRefresh Callbacks	N	I4	RW	ml-ML	R(0)	This is the total number of attempted OPC Data Access 2.0x REFRESH data callbacks that have occurred between the server and all connected OPC clients. This total is inclusive for all connected OPC clients.
\$System.Groups.Statistics.TotalFailedWrite Callbacks	N	I4	RW	ml-ML	R(0)	This is the total number of attempted OPC Data Access 2.0x WRITE data callbacks that have failed between the server and all connected OPC clients. This total is inclusive for all connected OPC clients. The difference between this value and the "TotalWriteCallbacks" value represents the total number of successful WRITE callbacks.
\$System.Groups.Statistics.TotalOverRuns	Y	I4	RW	ml-ML	R(0)	Count of number of times any group had a LastLoopTime which exceeded the scan rate for the group.
\$System.Groups.Statistics.TotalRead Callbacks	N	I4	RW	ml-ML	R(0)	This is the total number of attempted OPC Data Access 2.0x READ data callbacks that have occurred between the server and all connected OPC clients. This total is inclusive for all connected OPC clients.
\$System.Groups.Statistics.TotalWrite Callbacks	N	I4	RW	ml-ML	R(0)	This is the total number of attempted OPC Data Access 2.0x WRITE data callbacks that have occurred between the server and all connected OPC clients. This total is inclusive for all connected OPC clients.
\$System.NumItems	Y	I4	R	0-ML	NA	Number of OPC Item instances within the OPC Device Server (ie. number of Items within all groups within all clients currently connected.)
\$System.MaxItems		I4	RW	ml-ML	R(NumItems)	Max NumItems

System Tag Name	CPA	Type	RW	Range	Default	Description
\$System.NumOrphanGroups	Y	I4	R	0-ML	NA	Number of OPC Groups for which the connection to the client at the server level was shutdown without shutting down the connection to the OPCGroup Interface for groups within that server. (In which case the OPC Device Server must maintain an empty "OrphanGroup" in case the client makes future use of that OPC Group Interface. Such a Orphan is marked as removed, and any OPC operations except for Releasing the interface will report E_FAIL).
\$System.MaxOrphanGroups		I4	RW	ml-ML	R(NumOrphanGroups)	Max NumOrphanGroups

Key:

R = Readable

W = Writeable

ML = Max Long, (2³¹)-1

mL = Min Long , -(2³¹)

NA = not applicable

CPA = Viewable in Control Panel Applet

R(#) in Default column indicates an initial value as well as what the variable is reset to via \$System.ResetStatistics



D.0 APPENDIX D – Data Quality

For reference, Table D-1 maps APACS+ to OPC data quality:

Table D-1 APACS+ to OPC Data Quality Conversions

APACS+ Data Quality	OPC Data Quality
DATA_GOOD	OPC_QUALITY_GOOD
DATA_BAD	OPC_QUALITY_BAD
DATA_QUESTIONABLE	OPC_QUALITY_UNCERTAIN
DATA_UNAVAILABLE	OPC_QUALITY_BAD, OPC_QUALITY_OUT_OF_SERVICE

■

E.0 APPENDIX E – Program Boundaries and Considerations

OPC Device Server software boundaries and considerations are as follows:

- All optional OPC interfaces are supported except for the following:
 - OPC Server Object: IOPCServerPublicGroups, IOPCBrowseServerAddressSpace, and IpersistFile
 - OPC Group Object: IOPCPublicGroupStateMgt
- Since a database is not required, OPC reports EUNType as None and does not support Deadband within the OPC Device Server (Deadband can be set and retrieved for an OPCGroup, however; the value will not affect any operations).
- Since all tag addressing information is contained in the Item ID, the OPC Access Path is ignored. When queried, "" is reported as the access path.
- Locale information is not used. All timestamps and information are reported in the locale of the system running the OPC Device Server.
- The Runtime API installed on the PC (M-BUS or Ethernet) selects the communication path to be used for all APACS+ Controllers .
- APACS+ Resources cannot be named "\$System". This name causes selection of in-memory "System tags" within the OPC Device Server.
- The startup of APACS+ communications (via the APACS Data Services API) can take 10 seconds or more. The OPC Device Server will be able to access tags after initialization is completed.
- The OPC Device Server should always be used with the APACS+ Runtime API at Service Pack E or higher.
- Single tag Reads and Writes can be issued to a controller at the same time that a Taglist Read is occurring. In this case, the API should still return up-to-date data in the Taglist Read, even if the single Tag Read occurred after the taglist fetched the value for the given tag. However if API anomalies are discovered in this area, the \$System.Apacs.Resource.Limits.OperationsActive can be set to a value of 1. This only allows one operation (Taglist Read, Tag Read or Tag Write) to be issued against a controller at a time, however; this can slightly reduce the performance of the OPC Device Server.

■

