

Wonderware® OPCLink

User's Guide

Revision H
February 2003

Invensys Systems, Inc.

All rights reserved. No part of this documentation shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the Invensys Systems, Inc. No copyright or patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this documentation, the publisher and author assume no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein.

The information in this documentation is subject to change without notice and does not represent a commitment on the part of Invensys Systems, Inc. The software described in this documentation is furnished under a license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of these agreements.

© 2003 Invensys Systems, Inc. All Rights Reserved.

Invensys Systems, Inc.
26561 Rancho Parkway South
Lake Forest, CA 92630 U.S.A.
(949) 727-3200
<http://www.wonderware.com>

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Invensys Systems, Inc. cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Alarm Logger, ActiveFactory, ArchestrA, Avantis, DBDump, DBLoad, DTAnalyst, FactoryFocus, FactoryOffice, FactorySuite, hotlinks, InBatch, InControl, IndustrialRAD, IndustrialSQL Server, InTouch, InTrack, MaintenanceSuite, MuniSuite, QI Analyst, SCADAAlarm, SCADASuite, SuiteLink, SuiteVoyager, WindowMaker, WindowViewer, Wonderware, and Wonderware Logger are trademarks of Invensys plc, its subsidiaries and affiliates. All other brands may be trademarks of their respective owners.

Contents

Introduction	1
Communication Protocols	1
Accessing Remote Items via OPCLink	2
Optimizing an OPCLink Network	3
OPCLink Main Window	4
File Menu	5
New	5
Open	5
Save As	5
Exit	5
Configuring OPCLink	6
Configuring a Topic Definition	7
Topic Definition	7
OPCLink Topic Definition	8
Configuration File Format	14
Saving Multiple Configuration Files	14
Configuring OPCLink (Server) Settings	15
Server Settings	15
OPC Servers	17
The Synchronous OPC server activation	17
Local Server CoCreate Option	18
Poke Retries	18
Configuring the Logger	19
Configuring Security	20
Data Menu	21
Data Monitor	21
Dump Active Data	23
DB Dump	24
OPC Server Status	24
Help Menu	26
Contents	26
How to Use Help	26
About OPCLink	26
Item Names	27
Predefined Item/Point Names	28
UPDATEINTERVAL Item	28

STORESETTINGS Item	28
ITEMCOUNT Item	29
ERRORCOUNT Item.....	29
WRITECOMPLETE Item	29
STATUS Item.....	30
READCOMPLETE Item.....	30
POLLNOW Item	30
Monitoring OPCLink Communications to OPC Servers	31
Using the Status Item in Excel.....	31
Monitoring the Status of an InTouch I/O Conversation	31
Using DDEStatus and IOStatus in Excel.....	31
Reading Values from OPCLink into Excel.....	32
Writing Values to OPCLink from Excel.....	33
Troubleshooting I/O Server Communication Problems.....	34
Debugging Communication Between InTouch and an I/O Server	34
Debugging Communication Between SuiteLink and an I/O Server	36
Debugging Communication Between an I/O Server and a PLC.....	37
Special Wonderware Logger Messages	45

Wonderware OPCLink



Introduction

The Wonderware® OPCLink is a Microsoft® Windows® application program that acts as a communication protocol converter. This allows other Windows application programs access to data from local or remote OPC servers. OPCLink detects automatically which OPC data access version is implemented in OPC servers (up to version 2.04) and uses the latest version available. OPCLink connects to OPC servers, converts client commands to OPC protocol and transfers data back to clients using DDE, FastDDE, or SuiteLink™.

Communication Protocols

Dynamic Data Exchange (DDE) is a communication protocol developed by Microsoft to allow applications in the Windows environment to send/receive data and instructions to/from each other. It implements a client-server relationship between two concurrently running applications. The server application provides the data and accepts requests from any other application interested in its data. Requesting applications are called clients. Some applications such as InTouch and Microsoft Excel can simultaneously be both a client and a server.

FastDDE provides a means of packing many proprietary Wonderware DDE messages into a single Microsoft DDE message. This packing improves efficiency and performance by reducing the total number of DDE transactions required between a client and a server. Although Wonderware's FastDDE has extended the usefulness of DDE for our industry, this extension is being pushed to its performance constraints in distributed environments.

NetDDE™ extends the standard Windows DDE functionality to include communication over local area networks and through serial ports. Network extensions are available to allow DDE links between applications running on different computers connected via networks or modems. For example, NetDDE supports DDE between applications running on IBM® compatible computers connected via LAN or modem and DDE-aware applications running on non-PC based platforms under operating environments such as VMS™ and UNIX®.

SuiteLink uses a TCP/IP based protocol and is designed specifically to meet industrial needs such as data integrity, high-throughput, and easier diagnostics. This protocol standard is supported on Microsoft Windows NT 4.0 and Windows 2000.

SuiteLink is not a replacement for DDE, FastDDE, or NetDDE. The protocol used between a client and a server depends on your network connections and configurations. SuiteLink was designed to be the industrial data network distribution standard and provides the following features:

- Value Time Quality (VTQ) places a time stamp and quality indicator on all data values delivered to VTQ-aware clients.
- Extensive diagnostics of the data throughput, server loading, computer resource consumption, and network transport are made accessible through the Microsoft Windows NT and Windows 2000 operating systems performance monitor. This feature is critical for the scheme and maintenance of distributed industrial networks.
- Consistent high data volumes can be maintained between applications regardless if the applications are on a single node or distributed over a large node count.
- The network transport protocol is TCP/IP using Microsoft's standard WinSock interface.

Accessing Remote Items via OPCLink

The communication protocol addresses an element of data in a conversation that uses a three-part naming convention that includes the application name, topic name and item name. The following briefly describes each portion of this naming convention:

application name The name of the Windows program (server) that will be accessing the data element. In the case of data coming from or going to OPC servers, the application portion of the address is **OPCLINK**.

topic name Meaningful names are configured in the server to identify specific devices. These names are then used as the topic name in all conversations to that device. For example, **S7400_EMU**.

Note: You can define multiple topic names for the same device (PLC) to poll different points at different rates.

item name A specific data element within the specified topic. For example, when using OPCLink, an item can be a relay, timer, counter, register, etc., defined in the OPC server.

Note: The item/point names are predefined by the server. The term "point" is used interchangeably with the term "item" in this user's guide.

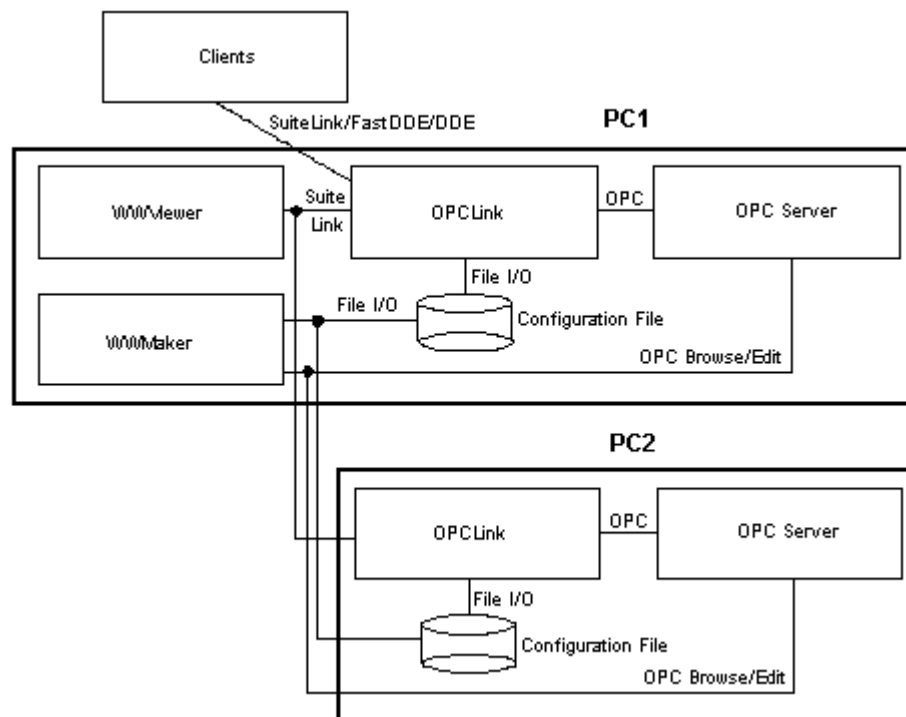
☞ For more information on item/point names, see the "Item Names" section later in this user's guide.

Optimizing an OPCLink Network

OPCLink can connect to local or remote OPC servers and can connect to local or remote clients. The robust communications protocol, SuiteLink, is recommended for all remote connections. This means an OPC server will be more stable and deliver higher performance if OPCLink and the OPC server are installed on the same computer and its clients use SuiteLink remote connections if needed.

You can connect OPCLink via DCOM to remote OPC servers, however, it is **not** recommended.

Graphical Representation of OPC Connection



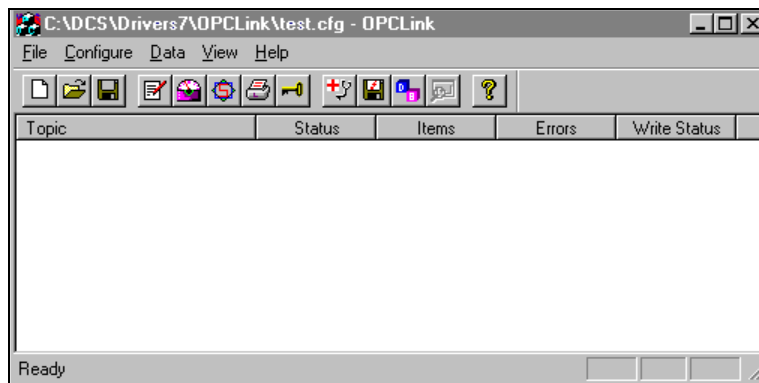
OPCLink Main Window



Double-click on the OPCLink icon to start.

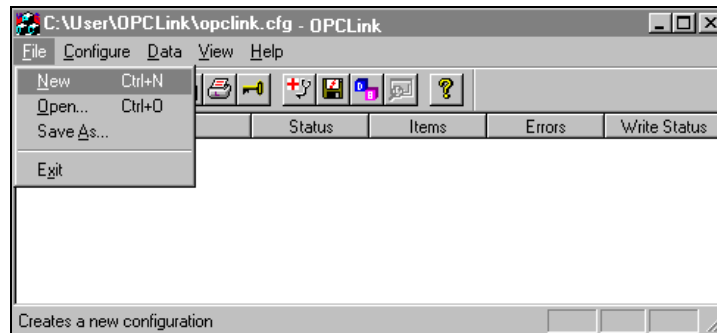
When OPCLink starts, the main window will appear.

The current configuration file and its full path are displayed in the main window title bar. If any topic is active, it will appear in the client area of the main window called the topic monitor list.



File Menu

File menu options deal with loading and saving the internal topic configuration database.



New

Clears the internal topic database for a new topic configuration. This menu is disabled when OPCLink is active.

Open

Replaces the internal topic database with the information contained in the new topic configuration file. This file is automatically loaded the next time OPCLink is started. This menu is disabled when OPCLink is active.

Save As

Saves the internal topic database under a new name. This option is only needed if you want to save the configuration under a different name. If you edit the internal topic database, the data is stored automatically in the current configuration file. This menu is disabled when OPCLink is active.

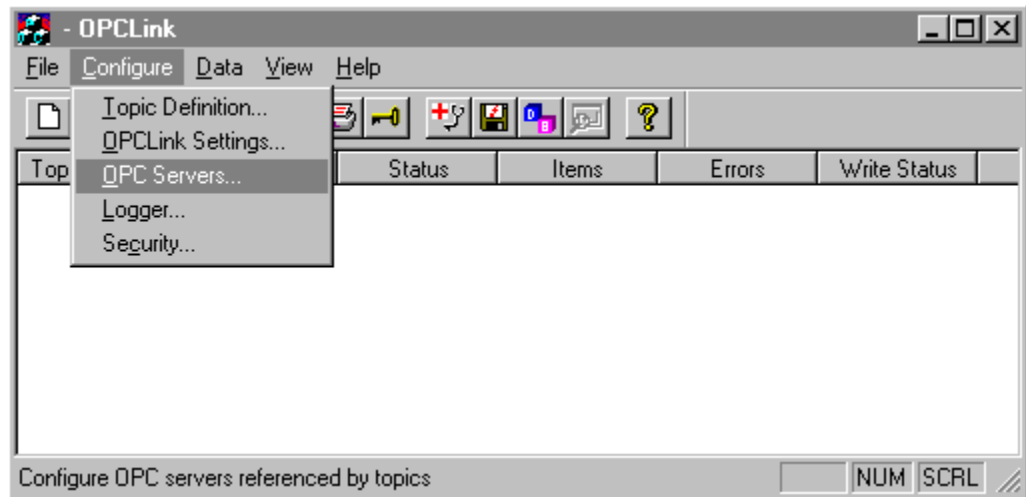
Exit

Terminates OPCLink.

Configuring OPCLink

Once OPCLink has been installed, some configuration is required. Configuring OPCLink automatically saves the data in a configuration file. If no configuration file is selected, the user is prompted to select a file name.

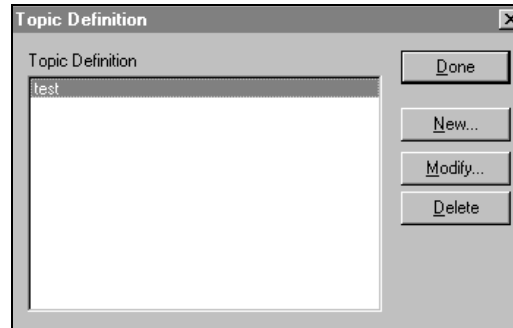
To access the options used for the various configurations, open the **Configure** menu:



Note: If any of the options appear grayed, then these options are not available with this software version.

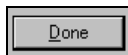
Configuring a Topic Definition

Use the **Topic Definition** option from the Configure menu to create new, modify or delete topic definitions. One or more topic definitions must exist for each OPC server communicating with OPCLink. Each topic definition must contain a unique name. When this option is selected, the **Topic Definition** dialog box will appear:



Topic Definition

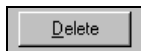
Note: Once topics have been defined, their names will be listed in the **Topics** section of this dialog box.



Click this button to close the dialog box and accept any new definitions, modifications or deletions made.



To modify or view an existing topic definition, select its name in the list and click on this button. The **OPCLINK Topic Definition** dialog box (described below) will appear displaying the selected topic definition.



To delete an existing topic definition, select its name in the list and click on this button. A message box will appear prompting you to confirm the deletion.



To add a new topic definition, click on this button. The **OPCLINK Topic Definition** dialog box will appear:

The screenshot shows the 'OPCLink Topic Definition' dialog box with the following settings:

- Topic Name: S7400_SN_104
- Node Name: PcNode1
- OPC Server Name: ArcheztA.DASS7.1
- OPC Path: SNIE.SPS1.SPS1_CP1
- Update Interval: 1000 ms
- Enable access to update interval:
- Poke asynchronously:
- Mode After Poke: None
- Transaction Timeout: 60000 ms
- Poke mode:
 - Control mode
 - Transition mode
 - Full optimization
- Lifeclock Settings:
 - Lifeclock:
 - Timeout: 0 ms

OPCLink Topic Definition

Topic Name: S7400_EMU

Enter a unique name for the PLC in this field.

Note: When communicating with InTouch, this **exact** name is used as the topic name in the Access Name definition.

Node Name:

If you want to use DCOM for connecting to a remote OPC server, you must specify the name of the machine where the remote OPC server resides.

Note: You have to execute DCOMCNFG from the **Start\Run** dialog of the remote OPC server machine to allow OPCLink or other OPC clients to access.

OPC Server Name: OPCSimaticNet

Select the name of the OPC server that will be used by this topic. The list box shows the registered OPC servers in the system.

OPC Path: [S7:EcoscreenEMUJEMU[S7TCPIP]

Enter the name of the OPC path used by this topic. This OPC Path is the first part of a full OPC item name string common to all items used in this topic. In a hierarchical address space, it usually is the item name without the leaf section. OPCLink combines the OPC Path and I/O name to construct the fully qualified OPC name.

Note: If the OPC server supports browsing you can view it's exposed address space via the **Browse** button.

Update Interval: [1000] ms

Enter the frequency (in milliseconds) that OPCLink will acquire data for the items/points associated with this topic. If you enter zero, OPCLink will not gather data from the OPC server.

Note: Different items/points can be polled at different rates by defining multiple topic names for the same PLC and setting different update rates for each topic.

Enable access to update interval

Select this option to enable client access to the update interval. If this box is checked, a client can read and write this topics update interval through the I/O item name: UPDATEINTERVAL. If this box is not checked, the client only reads the update interval.

Poke asynchronously

Select this option to process all pokes within this topic as asynchronous pokes. All OPC servers should support both synchronous and asynchronous pokes.

Mode After Poke: [None] ▼

In OPC, reading and poking are absolutely independent from each other. However, to avoid inconsistencies in your client showing the value, you can select how OPCLink will process reading after a poke occurred.

None: Standard behavior as suggested by OPC Foundation. This is the fastest way to read and poke values. Select this option if you need fast updates of reading values and flickering of poked values in the client is not important for you. Note that with some OPC servers (the minority of them) the client may not show the correct value of the device after the poke. This selection will process the poke only.

Refresh after poke: Selecting this option forces OPCLink to refresh all values on that topic after a poke occurred. This guarantees that the latest values in the device will be seen in the client. Nevertheless, there may be a flickering of the values because in the meantime regular-read messages may arrive.

Demand read after poke: Selecting this option forces OPCLink to refresh all the poked values after the poke has been processed. During this time, all-read messages from the OPC server, related to the poked items, will be blocked. This prevents the client from flickering and will show the latest values of the device in the client.

For example:

OPC server value = 5

Client pokes a value of 3

OPC server value initially changes to 3

But the OPC server value is rapidly reset to a value of 5 by some other method

OPC server may not report the transition of value from 3 to 5

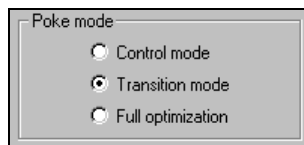
Results will be, the OPC server value is 5, but the client value is still 3

With Refresh after poke or Demand read after poke enabled, both the OPC server and the client will show the same value as 5 afterwards.



Enter the timeout for transactions (in milliseconds). OPCLink will monitor all transactions (pokes, demand reads etc.) for completion with respect to this timeout.

Note: Transactions are finite operations on a fixed list of items. Transaction timeouts should never occur. Every OPC server is required to complete transactions with or without errors. This timeout is a safety procedure designed to avoid leaving any items in the pending state when transactions are not completed due to a malfunctioning OPC server. This timeout should be set to a value, which allows the OPC server to complete any transaction within this time frame (including large recipe downloads etc.).



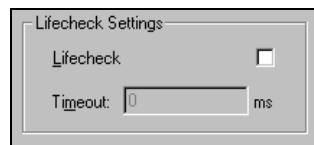
With these three radio buttons you are able to tune the poking behavior to the PLC. The three modes behave as follows.

Control mode: This mode makes no folding of write values. This should be selected when using this topic with control clients like InBatch and InControl. If selected the server processes all poked values in the order they are received from a client and makes no folding of poke values.

Transition mode: This mode implements poke optimization except the server receives more than one value per item in one timeslice. If this happens only the first, the second and the last value is poked. This is the default selection and preserves InTouch sliders from stuttering.

Full optimization: This mode tells the server that he should build poke messages regardless of folding and poke order aspects.

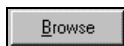
Note: The highest performance can be achieved with the **Full optimization** mode. But if you must be aware of the poke order and that no intermediate poke value is lost you have to choose the **Control mode**.



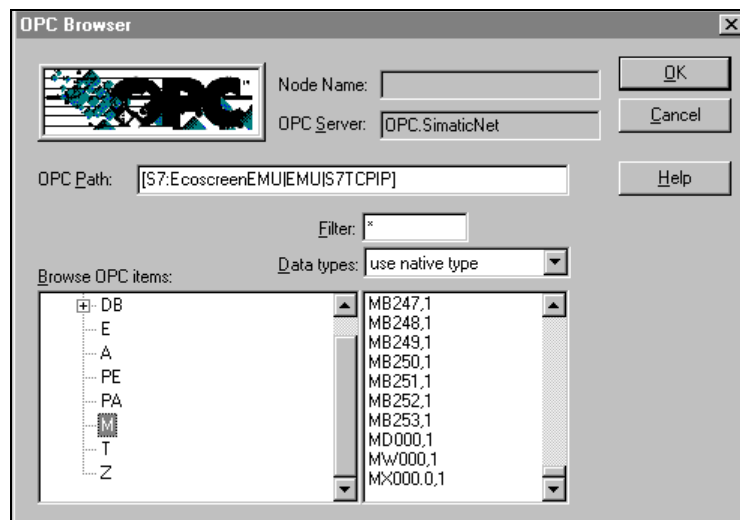
OPC and the underlying COM layer do not notify clients if a server has exited or has ceased to supply data. When the 'Lifecycle' option is enabled, OPCLink checks regularly if it receives data from an OPC server. If data does not arrive in a certain timeout interval, OPCLink forces a refresh through an asynchronous call. If this call fails or the OPC server fails to respond in a timeout interval the topic status is set to 0 (BAD) and all items are set to bad quality (communication failure). Timeout interval is calculated according to the following formula:

$$\text{timeout} = \text{updateinterval} + \text{OPC communication timeout}$$

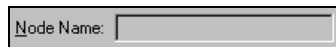
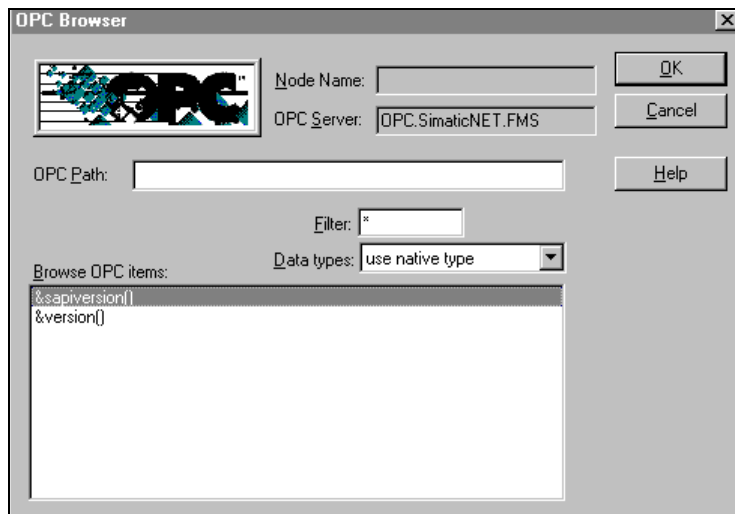
where OPC communication timeout is set to the value specified in the 'Timeout:' edit box (in milliseconds).



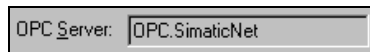
Click Browse to access the OPC Browser. This dialog box allows you to browse through exposed address spaces of OPC servers.



If the OPC server has a flat address space, the Browser window is not divided.



Here you can see which PC the currently connected OPC server resides.



This control displays the currently connected OPC server.

OPC Path: [S7:EcoscreenEMUIEMUIS7TCPIP]

The OPC Path edit field is automatically filled while browsing through the OPC servers address space. You can override it by manually entering a string.

Note: The OPC Path is the first part of a full OPC item name string common to all items that will be used in this topic. In a hierarchical address space it usually is the item name without the leaf section (combined branches in the ‘Browse OPC items’ tree view). OPCLink will combine the OPC Path and the I/O name to construct the fully qualified OPC name passed to an OPC server.

Filter: *

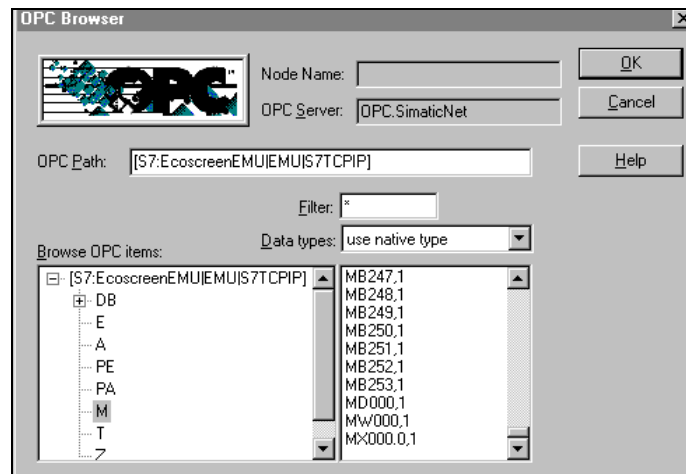
The filter allows you to display only names that match the filter string. Use wildcards for missing characters! For example, for filtering of items beginning with “MB”, type in “MB*”.

Note: Not all OPC servers support filtering.

Data types: use native type

The data types selection allows you to display only names that match the data type.

Note: Not all OPC servers support data type filtering.



This is the tree view of the OPC server’s hierarchical address space with the branches on the left side and the leaves on the right side.

Note: Not all OPC servers have an hierarchical address space and/or support browsing. Browsing of OPC servers from a remote node is supported ONLY on Windows NT and Windows 2000 platforms.

OK

Click **OK** to close the dialog box.

Note: If a filename has not been defined, the OPCLink server prompts you to enter a filename for saving the configuration.

Configuration File Format

The configuration file is an ASCII file that can be edited with any standard text editor. It complies with the Windows profile format having a section for each topic and a general section for server information. However, in order to avoid damage to the internal structure of the configuration file, it is recommended **not** to manually edit this file.

Example of an OPCLINK configuration file:

```
[GENERAL_SETTINGS]
ALLOW_ACTIVE_TOPIC_MODIFICATION=1
[S7400_EMU]
UPDATEINTERVAL=1000
ENABLE_ACCESS_TO_UPDATE_INTERVAL=0
OPC_SERVERNODE=
OPC_SERVERNAME=OPC.SimaticNet
OPC_PATH=[S7:EcoscreenEMU|EMU|S7TCPIP]
POKE_ASYNCHRONOUSLY=1
LIFECHECKTIMEOUT=0
REFRESH_AFTER_POKE=0
POKEMODE=1
```

Saving Multiple Configuration Files

There is no limit to the number of configuration files that you can create. This allows multiple configuration files to be accessed by using special switches ('/d:' and '/f:').

For example, to start OPCLink using OPCLink.cfg as configuration file located in a different directory, click the **Start** button on the taskbar, then choose **Run** and enter the following in the **Open** combo box:

“directory path\OPCLINK”/d:<drive or network path>\directoryname

Also, you can create and access multiple configuration files in separate directories with specific filenames by using another special switch (/f:):

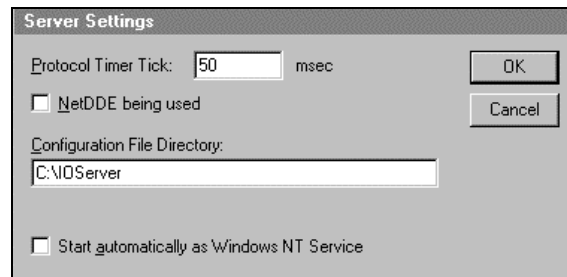
“directory path\OPCLINK”/f:<drive or network path>\directoryname\configurationfilename

Configuring OPCLink (Server) Settings

Use the **OPCLink Settings** option from the Configure menu to change the protocol timer, to enable the network using Wonderware NetDDE, to specify the default configuration file path, or to enable the server to start automatically as a Windows NT service.

Note: When configuring the server on Windows NT, the user must be logged on with system administrator privileges. This will ensure that updates to the system registry may be performed.

When the **Server Settings** option is selected, the **Server Settings** dialog box will appear:



Note: The server described in this section refers to OPCLink.

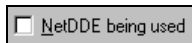
Server Settings



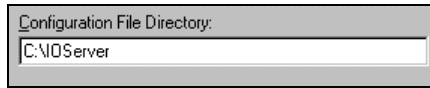
Enter the frequency (in milliseconds) that the server is to check for data to process. This should be approximately two to four times faster than the fastest rate desired to update data from the equipment.

Note: The default protocol timer tick value will vary between servers.

Note: For OPCLink, the protocol timer tick only effects activations, deactivations, and pokes. All updates are done immediately after new data has been received from an OPC server.

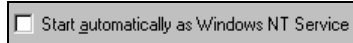


Select this option if you are networking using Wonderware NetDDE.



To create a new default configuration file, enter the complete path for the directory in which the file is to be saved in this field. This new path will automatically be written to the **WIN.INI** file and the server will use this path to load its configuration file the next time it is started.

Note: There is no limit to the number of configuration files created. When using the server with InTouch, we recommend that you save the configuration file in your application directory.



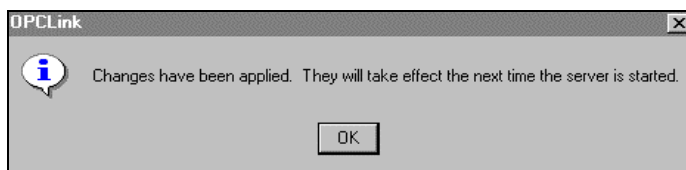
Enabling this option will cause OPCLink to start as a Windows NT service.

Windows NT offers the capability of running applications even when a user is not logged on to the system. This is valuable when systems must operate in an unattended mode. Enabling this option and rebooting the system will cause OPCLink to run as a Windows NT service. However, to view configuration information or to reconfigure the server, the user must log on to the system. Any server related problems that may arise such as missing adapter cards, licensing failures or device drivers not loading will not be visible to the user until a log on is performed. Disabling this option and rebooting the system will cause OPCLink to run as a Windows NT application program once again.

Note: It is highly recommended that both OPCLink and its OPC servers are configured and communicating successfully prior to running them as Windows NT services. Also, both OPCLink and its OPC servers should be running as Windows NT services at the same time, especially when they are remote from each other.

Click **Cancel** to close the dialog box without saving changes.

Click **OK** to accept OPCLink settings. The following message box will appear:

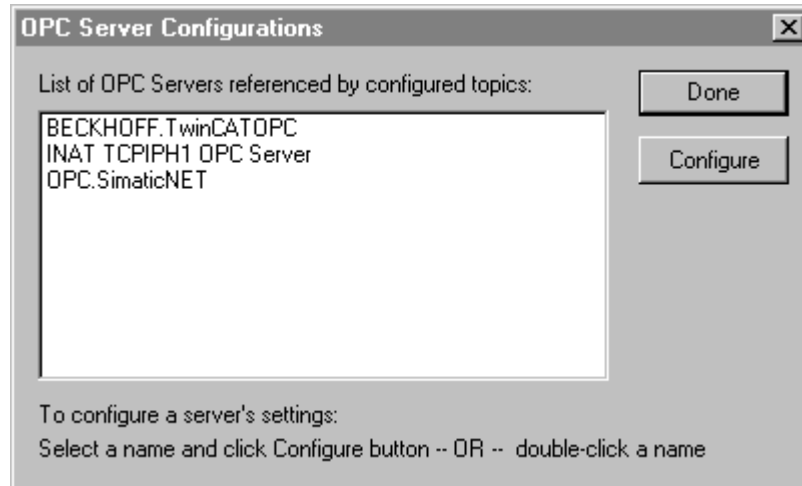


Click **OK** to close the dialog box.

Note: You must restart OPCLink for the changes to take effect.

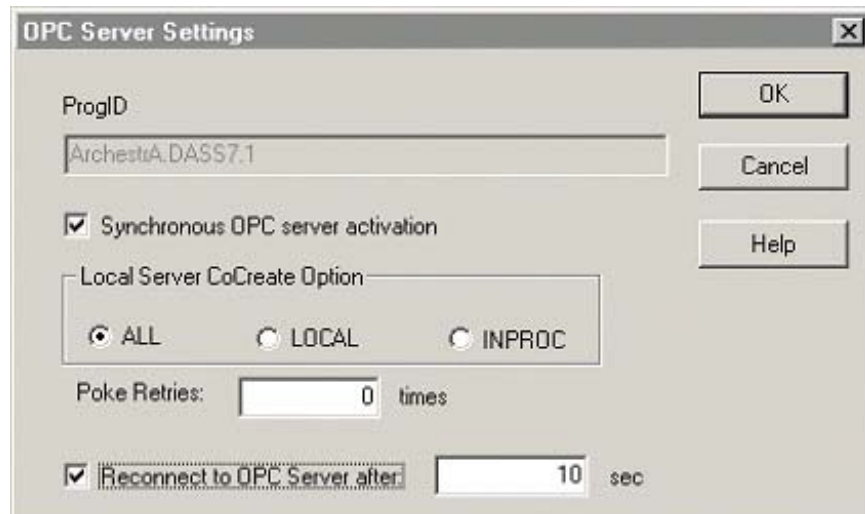
OPC Servers

Use the **OPC Servers** option from the Configure menu to access the OPC Server Configuration window. This window lists all OPC Servers alphabetically, referenced by configured topics.



To configure the settings for a listed server name, double-click the server name, or select the server name, then click Configure.

The OPC Server Settings window appears:



The Synchronous OPC server activation

This controls the OPC server activation mode. Leaving this box unchecked allows OPCLink to perform different tasks (data acquisition etc.), while starting up and activating an OPC server. However, some OPC servers work in a 'Handler' mode, requiring this box to be checked to work correctly. In this mode, the OPC server is activated synchronously making OPCLink wait for the completion of the startup.

Asynchronous operation of OPC settings will enhance performance. This is a desired behavior.

Note: The **Synchronous OPC server activation** option is selected by default.

Local Server CoCreate Option

- ALL:** Instantiate the OPC server in the following precedence:
In-Proc, local out-of-proc, or remote COM server.
- LOCAL:** Instantiate the OPC server as a local out-of-proc COM server
- INPROC:** Instantiate the OPC server as an in-proc COM server

Poke Retries

When there are Poke errors OPCLINK will try single pokes; the number of retries is configurable. A value of "0" in this field disables the retries.

Note: If several pokes have been processes within the same time (example: recipe download) only pokes with errors are retried.

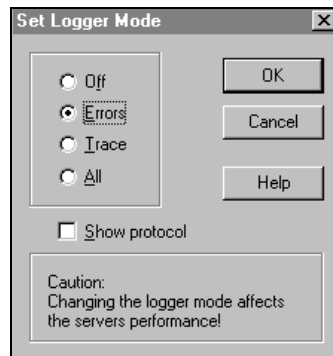
Reconnect to OPC Server after

Check this setting if you want OPCLink to reconnect to an OPC server automatically when a connection failure to the OPC server is detected. You can enter a latency time (in seconds) to wait on reconnecting. Entering "0" sec instructs OPCLink to immediately reconnect after detecting the connection failure.

Note: In normal operation mode, OPCLink should always be configured to reconnect to an OPC server after a connection failure. To allow manual shutdown of OPC servers, disable this setting or configure a timeout that allows the OPC server to complete the shutdown before OPCLink attempts to restart.

Configuring the Logger

This option is used to set the Logger mode of the server.



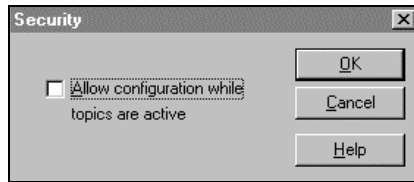
Select **ERRORS** for normal operation message and error logging to the Wonderware Logger. Select **TRACE** to log errors and all activation/deactivation activities including calls to OPCLink. If you select **ALL**, the server will log all activities including single item value updates. Selecting **OFF** disables all message logging. The option **Show protocol**, enables logging of further interactions with connected OPC servers.

Note: When logging activity increases, OPCLink's performance degrades.

Note: The logger mode defaults to **ERRORS** and deselects **Show protocol** whenever OPCLink is started.

Configuring Security

This option is used to lock the servers configuration against changes during runtime.



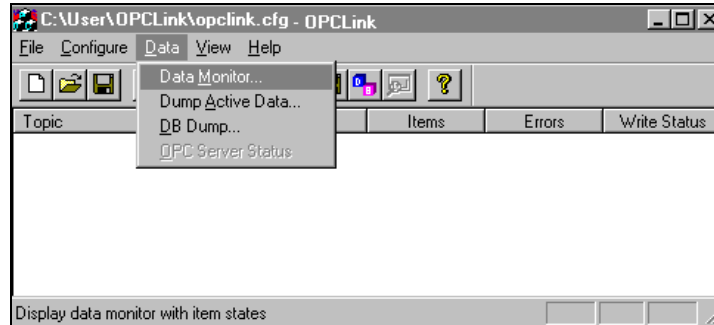
This dialog box shows the default state of this option. All topics are locked against changes while the server is active (a client is connected). You can still view the topic configurations. If you check: 'Allow configuration while topics are active', you can access some elements of the topic configuration while the topics are active. You can change the 'Update Interval', 'Refresh after poke' and 'Enable access to update interval'. You cannot add, delete, rename, or change other properties of a topic.

Click **OK** to close the dialog box.

To close the **Security** dialog box without saving the changes, click **Cancel**.

Data Menu

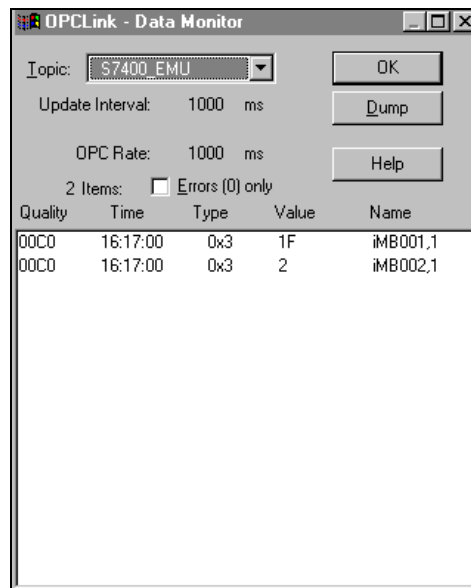
The Data menu offers functionality, **Data Monitor**, **Dump Active Data**, and **OPC Server Status** for diagnostic purposes during runtime (when a client is attached). Also, it provides capability, **DB Dump**, to transfer the internal topic database to InTouch.



The following describes the **Data Monitor** menu command.

Data Monitor

Use the **Data Monitor** option from the data menu to display information about a selected topic with active items or errors. The information is updated as it changes. This is a useful tool for tracking down errors, for determining performance and to validate item values. When the **Data Monitor** option is selected, the **OPCLink Data Monitor** dialog box will appear.



The data monitor dialog box displays information of any topic with active items or errors. The data is updated as it changes.

Topic:

This drop down list box allows you to select any of the active topics. A preceding ‘*’ indicates that at least one item in this topic has an error while a preceding ‘#’ indicates bad status.

Update Interval: ms

This field displays the current configured update interval of the topic. This value changes whenever the value is poked.

OPC Rate: ms

This field displays the current configured update interval of the corresponding group from the OPC server.

2 Items: errors (0) only

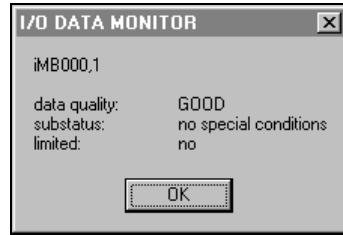
This field displays the number of active items and the number of items with errors (in parenthesis). If you check this box, only items with errors will be displayed in the item data list box.

Quality	Time	Type	Value	Name
0000	16:17:00	0x3	1F	iMB001,1
0000	16:17:00	0x3	2	iMB002,1

The item list box adapts to the size of the dialog box. So you can resize it for your personal needs. The server will save the size and position.

The item data list box displays information about each item. The box is divided into 5 columns. The first column displays the data quality. The second column shows the timestamp of the last update of that item. The third column indicates the type enumerator (VARIANT type). The fourth column shows the item value (represented as Hexadecimal value or as String). The last column contains the item name. For further information of the Quality value and the VARIANT type please refer to the OPC documentation provided by the vendor of your OPC server.

Clicking on an item line generates a dialog box displaying the item properties:



Click this button to dump the contents of the monitor window into an ASCII text file. The resulting file can be loaded into Excel or any standard text editor. All columns are separated by tabs.

After the dump is complete, a message box containing the newly created ASCII text file name is displayed.



The server automatically generates the ASCII text file name and sets the files location using the following pattern:

<current working directory>\datmonXX.txt

Where XX is a two digit number between 00 and 99. OPCLink increments the number each time a file is dumped. When this number reaches 99, it will restart again from 00 in a cyclic fashion.

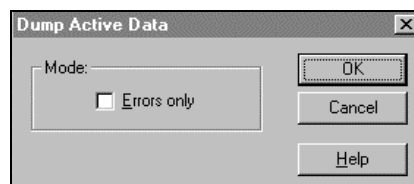
Note: Existing files with the same name are erased.

Click **OK** to return to the **Data Monitor** dialog box.

Note: Monitoring the items may affect OPCLink performance.

Dump Active Data

This command dumps all topics with active items into an ASCII text file.



Select the Errors only option to dump topics with errors. The resulting file can be loaded into Excel or any standard text editor. All columns are separated by tabs.

After the dump is complete, a message box containing the newly created ASCII text file name is displayed:



The server automatically generates the ASCII text file name and sets the files location using the following pattern:

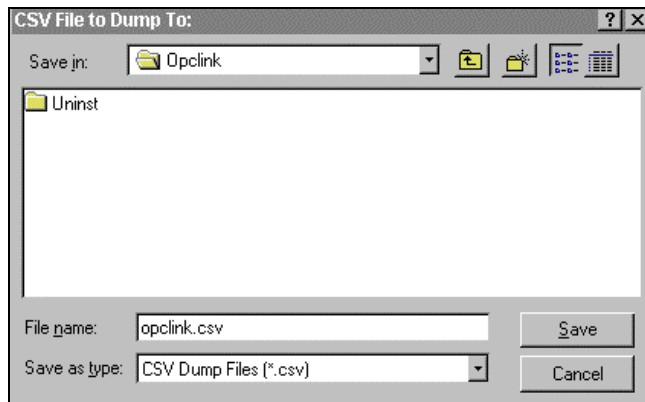
<current working directory>\datdmpXX.txt

Where XX is a two digit number between 00 and 99. OPCLink increments the number each time a file is dumped. When this number reaches 99, it will restart again from 00 in a cyclic fashion.

Note: Existing files with the same name are erased.

DB Dump

This command is used to dump the internal topic database into a CSV file. The resulting file can be loaded into InTouch using Wonderware's DbLoad utility.



OPC Server Status

This option monitors the status of any connected OPC server. An OPC server will provide the time it was started, time of last update, current time, its status (OPCLink adds a string which displays the version of OPC data access used by this OPC server), vendor information, version information, number of connected groups, and bandwidth it is consuming. This dialog box is only available if there are OPC servers connected.

OPC Server Status [X]

OPC.SimaticNET [v]

Done

Time

Start: 06.12.01 15:12:29

Last: 06.12.01 15:12:48

Clock: 06.12.01 15:12:48

Help

Status: OPC server is running. (DA 2.0)

Vendor Info: SIMATIC NET OPC-Server

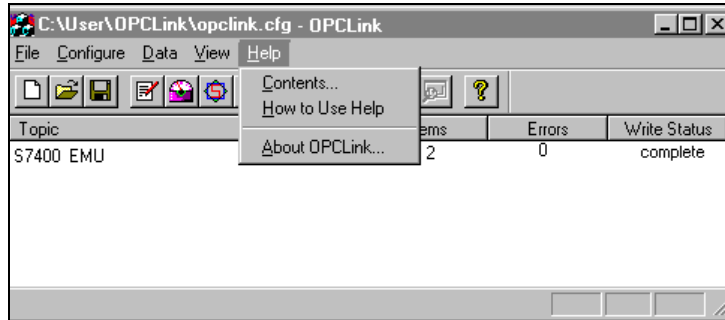
Build Info: Build Number=110.
Version Number: Major=2; Minor=2.

Groups: 1

Bandwidth: n/a

Help Menu

The Help menu contains three options that are used to access help for the server.



The following briefly describes the Help menu options.

Contents

This option is used to display the table of contents for the Help file.

How to Use Help

This option is used to access a list of basic instructions for using the Help file.

About OPCLink

This option provides the product name, software version, copyright, and licensing information.

Item Names

The OPCLink item syntax is based on the following rules.

Each item name must start with a prefix:

- d discrete value
- i integer value
- r real value
- m message value

The item names added to the OPC path of the topic without the prefix must give a fully qualified OPC item name for the OPC server of the topic.

Example:

```
item name:  itemperature
topic:      Slave0
OPC path:   PROFIBoard DP_0.Master0.Slave0.
OPC server: Softing.OPC_DP
results in the OPC item: PROFIBoard DP_0.Master0.Slave0.temperature
```

Here every slave/updateinterval combination has its own topic.

If you choose to have fewer topics, you can also configure:

```
item name:  iSlave0.temperature
topic:      Master0
OPC path:   PROFIBoard DP_0.Master0
OPC server: Softing.OPC_DP
results in the OPC item: PROFIBoard DP_0.Master0.Slave0.temperature
```

Here the slave information is part of the item name and each Master/updateinterval combination has its own topic.

Predefined Item/Point Names

All OPCLink topics have predefined item/point names to monitor and control communication properties.

UPDATEINTERVAL Item

The UPDATEINTERVAL item is used to access the currently set update interval. It is the same value displayed in the monitor window of OPCLink. It indicates the currently requested update interval in milliseconds. If the corresponding topic is configured for "Enable I/O access to update interval", a client can poke new values into this item. In any case the value can always be read through I/O. The range of valid values differs with respect to the topic settings. The value of zero indicates that no items on that topic are updated.

I/O Type: INTEGER
ACCESS: READ/WRITE (if configured)
RANGE: 1..2147483647
0 (topic inactive, no items are updated)

Note: By poking a value of zero into the update interval item, a client can conveniently stop all update activities on the corresponding topic without having to deactivate the items.

STORESETTINGS Item

The STORESETTINGS item is used to make the temporary update interval changes via I/O permanent. If the client pokes a value of 1 into the STORESETTINGS item, the currently set update interval is written to OPCLink's configuration file. If the update interval has been changed via I/O and STORESETTINGS is not poked to 1, OPCLink will use the original update interval for that topic the next time it is started. Reading the item will always give 0.

I/O Type: DISCRETE
ACCESS: READ/WRITE
RANGE: 0, 1

Note: Use this item to conveniently store your optimized configuration via I/O.

ITEMCOUNT Item

The ITEMCOUNT item is used to access the number of active items in the corresponding topic. It is the same value displayed in the data monitor window of OPCLink under "Items". This item is read only.

I/O Type: INTEGER
ACCESS: READ
RANGE: 0..2147483647

Note: Use the ITEMCOUNT item monitor the number of currently active items.

ERRORCOUNT Item

The ERRORCOUNT item is used to access the number of active items that have errors in the corresponding topic. It is the same value displayed in the data monitor window of OPCLink under "errors () only". If the communication status of a topic is bad, all items have errors (item count equals error count). This item is read only.

I/O Type: INTEGER
ACCESS: READ
RANGE: 0..2147483647

Note: Use the ERRORCOUNT item to monitor if there are any badly configured items or invalid item values. This is the case if the topic status is 1 and ERRORCOUNT is not zero.

WRITECOMPLETE Item

The WRITECOMPLETE item is used to access the state of pending write activities on the corresponding topic. If the topic is opened the value of the WRITECOMPLETE item is initially 1 indicating all write activities are complete - no pokes are pending. If values are poked into any items of the topic the value of the WRITECOMPLETE item changes to 0 indicating write activity is currently in progress. If the server has completed all write activities the value of the WRITECOMPLETE item changes to 1 if all pokes were successful or to -1 if at least one poke has failed. If the value of the WRITECOMPLETE item is not zero the client can poke 1 or -1 to it (poke a 1 to clear errors or a -1 to test a client reaction on write errors).

I/O Type: INTEGER
ACCESS: READ/WRITE
RANGE: -1,0,1

Note: Use the WRITECOMPLETE item to serialize a sequence of poke value into a single item or to monitor success or failure of poking into one or more items.

STATUS Item

The **STATUS** item is used to access the state of communication between OPCLink and an OPC server. The discrete item, **STATUS**, is set to **1** when communication is successful or is set to **0** when communication with the OPC server fails.

I/O Type: DISCRETE
ACCESS: READ
RANGE: 0,1

READCOMPLETE Item

The READCOMPLETE item is used to access the state of initial reads on all items in the corresponding topic. If the topic is opened and no item is advised, the value of the READCOMPLETE item is 0. After all activated items have been read once, the READCOMPLETE item changes to 1. If at least one item could not be read the READCOMPLETE item alters to -1 indicating there have been read errors.

If at least one new item is advised now READCOMPLETE changes to 0 and after this/these item(s) has/have been read it changes to 1 or -1 again.

Poking a 0 to this item resets the internal read states of all items in this topic. This resets the READCOMPLETE item to 0. If all items are read again after this poke, the READCOMPLETE item changes back to 1 or -1.

I/O Type: INTEGER
ACCESS: READ/WRITE
RANGE: -1,0,1

POLLNOW Item

Poking a 1 to the predefined POLLNOW item forces all items within this topic to be read immediately. This is useful if you want to get the newest values from the device, regardless of its update interval. You can use this if you want to trigger the reads within a topic by your own.

I/O Type: DISCRETE
ACCESS: READ/WRITE
RANGE: 0,1

Note: You can use POLLNOW and READCOMPLETE for measuring of the longest interval. Create a topic with update interval 0. Poke POLLNOW to 1 and READCOMPLETE to 0 and store the actual time. Then wait until "READCOMPLETE" changes back to 1. The difference between the current time and the stored time equals the update interval.

Monitoring OPCLink Communications to OPC Servers

For each OPCLink defined topic name there is a built-in discrete item used to monitor the status of communications between OPCLink and an OPC server. The discrete item, **Status**, is set to **0** when communication fails and is set to **1** when communication is successful.

Using the Status Item in Excel

The status of communications between OPCLink and an OPC server can be read into Excel by entering the following DDE reference formula in a cell on a spreadsheet:

```
=OPCLINK|S7400_EMU!Status
```

where:

OPCLINK	is the name of the DDE server application.
S7400_EMU	is the exact topic name defined in OPCLink for the OPC server.
Status	is the built-in discrete item used to monitor the status of communications between OPCLink and the OPC server.

Monitoring the Status of an InTouch I/O Conversation

WindowViewer supports built-in topic names called **DDEStatus** and **IOStatus** that can be used to monitor the status of specific I/O conversations. For example, let's assume that **WindowViewer** (VIEW) is communicating with OPCLink about an OPC server defined in OPCLink with **S7400_EMU** as its topic name. The discrete items, **DDEStatus** and **IOStatus** are set to 0 when this I/O conversation has failed and set to 1 when this I/O conversation is successful. For more information on the built-in topic names **DDEStatus** and **IOStatus**, see your online *InTouch User's Guide*.

Using DDEStatus and IOStatus in Excel

The I/O conversation status between InTouch and an OPCLink can be read into Excel by entering the following DDE reference formula in a cell on a spreadsheet:

```
=VIEW|DDEStatus!S7400_EMU
```

or:

```
=VIEW|IOStatus!S7400_EMU
```

where:

VIEW	is the InTouch application name.
[DDE] [IO]Status	are the topic names used to monitor InTouch I/O conversations.
S7400_EMU	is the exact Access Name defined in InTouch for this topic.

Reading Values from OPCLink into Excel

Values may be read directly into Excel spreadsheets from the server by entering a DDE formula into a cell using the following format:

=applicationname|topicname!itemname

Example formula:

=OPCLINK|S7400_EMU!'iMB001,1'

where:

OPCLINK	is the name of the server application.
S7400_EMU	is the exact topic name defined in OPCLink for the OPC server.
iMB001,1	is the actual location in the PLC that contains the data value. This is the item name from the OPC server with the OPCLink item type prefix.

In this example, each time the value of **iMB001,1** changes in the PLC, the OPC server will automatically send the new value to OPCLink which will update the cell containing the formula in Excel.

Note: In this example, each time the value of **iMB001,1** changes in the PLC, the OPC Server will automatically send the new value to OPCLink which will update the cell containing the formula in Excel.

Writing Values to OPCLink from Excel

Values may be written to the server from Microsoft Excel by creating an Excel macro that uses the **POKE** command. The proper command is entered in Excel as follows:

```
channel=INITIATE("applicationname","topicname")
=POKE(channel,"itemname",Data_Reference)
=TERMINATE(channel)
=RETURN()
```

The following describes each of the above **POKE** macro statements:

```
channel=INITIATE("applicationname","topicname")
```

Opens a channel to a specific topic name (defined in the server) in a particular application name (the executable name less the .EXE) and assigns the number of that opened channel to **channel**.

Note: By using the **channel=INITIATE** statement the word **channel** must be used in the **=POKE** statement instead of the actual cell reference. The "**applicationname**" and "**topicname**" portions of the formula must be enclosed in quotation marks.

```
=POKE(channel,"itemname",Data_Reference)
```

POKEs the value contained in the **Data_Reference** to the specified item name (actual location in the PLC) via the **channel** number returned by the previously executed **INITIATE** function. **Data_Reference** is the row/column ID of the cell containing the data value.

```
=TERMINATE(channel)
```

Closes the channel at the end of the macro. Some applications have a limited number of channels, therefore they should be closed when finished. **Channel** is the channel number returned by the previously executed **INITIATE** function.

```
=RETURN()
```

Marks the end of the macro.

Note: Refer to the **.XLM** sample Excel poke macro provided on the server program disk. Also refer to the Microsoft Excel manual for complete details on entering Remote Reference formulas for cells.

Troubleshooting I/O Server Communication Problems

This section provides you with some simple steps that can be taken to ascertain and correct communication problems. The problems described here represent the most probable causes of communication failure.

Note: This is a general troubleshooting guide and for the sake of brevity we cannot cover every possible source of communication problems.

Debugging Communication Between InTouch and an I/O Server

This section explains the most common error situations that can occur when attempting to establish communication between InTouch and a server.

Servers are Window applications that communicate with I/O, PLCs, and/or other data sources. If a server supports either the Microsoft Dynamic Data Exchange (DDE) or the Wonderware SuiteLink protocol, it is capable of communicating with the Wonderware InTouch program.

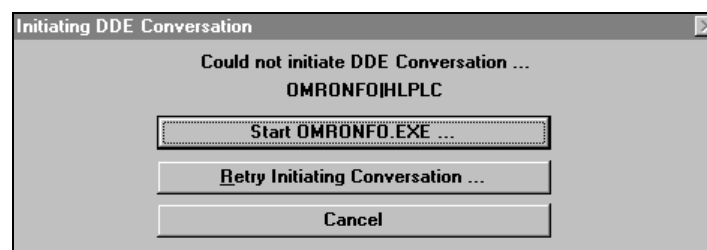
Note: All Wonderware version 7.0 or later servers support both DDE and SuiteLink. However, the SuiteLink protocol is only supported on the Windows NT (version 4.0 or later) operating system.

Servers respond to data requests made by other applications. Requesting applications are called clients. When WindowViewer acts as a client and requires the value of an item, it contacts the server and requests the item's value. The server will report the value and update WindowViewer only if a change occurs. All WindowViewer data requests provide information relating an item to a register, coil number, or I/O data point understood by the server. The server uses the information to automatically handle all messages to and from I/O, hardware devices (PLC), and/or other data sources.

Note: We highly recommend starting all the servers required by the InTouch application before starting WindowViewer. InTouch (versions prior to 7.0) will display the **Initiating DDE Conversation** message box for each uninitiated conversation.

For example:

If you start up WindowViewer and cannot successfully establish a conversation with a server, the following Initiating DDE Conversation dialog box will appear:



The information in the second line indicates that you have at least one I/O type tagname defined in your Tagname Dictionary that is associated with an Access Name that defines OMRONFO as the Application Name, and HLPLC as the Topic Name. Make note of exactly how the application and topic names are spelled.

☞ This example only applies when using a version of InTouch prior to InTouch 7.0.

To troubleshoot communication problems between WindowViewer and the server, perform the following steps as listed below.

➤ **Verify the I/O Server is running.**

1. Start the server program.
2. Verify the server is running by checking to see if it is in the Windows Task List.

On Windows NT, click the right mouse button on the Windows taskbar and select Task Manager from the menu. Click the Applications tab to view all currently running applications. Or press the CTRL+SHIFT+ESC keys.

On Windows 95, press the ALT+TAB keys while holding down the ALT key.

On Windows 3.1 or Windows for Workgroups, press the CTRL+ESC keys.

➤ **If the I/O Server is running, verify the I/O Server's program name is correct in all WindowMaker Access Name definitions.**

1. Switch to (or start) WindowMaker. Select Access Names from the Special Menu, the Access Name Definitions dialog box appears listing all Access Names defined in the WindowMaker.
2. In the Access Names list, select the Access Name referencing the server and click Modify. The Modify Access Name dialog box will appear.
3. Verify the server's program name in the Application Name box is correct. If it is wrong then correct it and click OK, else click Cancel.

☞ The server's exact "executable name" must be typed in the Application Name box in all Access Name definitions. The ".exe" extension is not used.

☞ If you are debugging a remote tagname reference, also verify that the node name for the remote computer in the Node Name box is correct.

4. Repeat steps 2 & 3 and verify the server program name is correct in all Access Names that use it.

➤ **If you still cannot establish a conversation, verify the exact topic name used in the WindowMaker Access Name definitions are defined in the I/O Server program.**

1. Close WindowViewer if it is running. The server cannot be configured if WindowViewer is running.
2. Start the server program.
3. From the server's Configure menu select Topic Definition. The Topic Definition dialog box appears listing all topic names defined in the server.
4. Verify that the topic name exists and is spelled exactly the same (including spaces) as the topic name referenced in the WindowMaker Access Name definition.

☞ Blank spaces cannot follow the topic name in either the server's Topic Definition or the Access Name definition.

5. If the topic name is different, either correct it in the server or switch to WindowMaker and correct it in the Access Name definition.
6. Once you performed the above procedure, restart WindowViewer and switch to the server program. Data should now appear in the server's program window to indicate that WindowViewer and the server are communicating.
 - ☞ The data in the server's program window indicates the read and write messages the server is sending to and receiving from the PLC. These are not error messages; only status messages are written to the server's program window.
7. If no data appears in the server's program window, switch to the Wonderware Logger to check for error messages. For example, a common error message is:

"Error for DDE: OMRONFO|HLPLC!<null>("item") Advise failed"

This message appears when the item defined in one or more tagnames is invalid for the server.

 - ☞ InTouch tagnames use specific naming conventions when accessing data from a server. The valid item names for all Wonderware servers are documented in their respective user's guides. Typically, the item naming conventions used by each server are consistent with the names used by the equipment manufacturer.
 - ☞ For more information on the Wonderware Logger, see your online *FactorySuite System Administrator's Guide*.

- **If you are still experiencing problems, continue with the following troubleshooting section.**

Debugging Communication Between SuiteLink and an I/O Server

If you have successfully applied the debug techniques listed in the previous section and are still experiencing communication problems to a server that is attempting to communicate using the SuiteLink protocol, perform the following steps as listed below:

- **Verify the I/O Server supports the Wonderware SuiteLink protocol, that is, the I/O Server is version 7.0 or above.**
- **Try communicating to the I/O Server using the DDE protocol. If this is not possible, then proceed to the next troubleshooting section otherwise continue with the following steps:**
 1. Verify Microsoft's TCP/IP stack is installed and configured properly.
 - ☞ SuiteLink uses the Microsoft TCP/IP stack for its communications even if the client application and the server reside on the same node.
 2. If you do not have an Ethernet card to bind to the TCP/IP stack, install the Microsoft Loop Back Adapter.
 3. Install the Microsoft TCP/IP stack.

Debugging Communication Between an I/O Server and a PLC

This section provides you with simple steps to diagnose and correct server to PLC communication problems. The debug techniques listed below address both serial and board servers. Disregard any information that is not applicable to the server type that you are using.

When attempting to establish communication between a server and a PLC, if no data appears in the server's program window and the data items are not updating in WindowViewer, switch to the Wonderware Logger and check for error messages.

☞ For more information on the Wonderware Logger, see your online *FactorySuite System Administrator's Guide*.

For example, some of the most common errors that may appear in the Wonderware Logger for serial servers are:

Response Timeout
WCRET = -2
WakeUp = -2
Receive Overrun
Framing Errors

Note: Unless specified otherwise, most serial communication based servers are full duplex. If you require a server for half duplex (one that monitors the CTS and RTS lines) or if you are not sure whether the PLC's protocol is full or half duplex, call your PLC supplier.

Also, during in-house server testing, we have found that the communication cards that use the National 16450 and 16550 UARTs seem to be less susceptible to level and timing problems. Cards based on other chips may work, but we recommend using the National cards. Some of the highly integrated UART chips (most notably, Winbond and UMC) have a tendency for their transmitters to hang, requiring re-initialization of the UART. If this occurs, you may have to restart the server or execute the Reinitialize I/O command from the Special menu in WindowViewer.

➤ **Check your cabling to the PLC.**

Is it wired correctly? Check for shorts, loose wires, broken wires, crossed wires, and so on.

☞ A continuity tester can be helpful here.

➤ **Verify the I/O Server's serial configuration settings (Parity, Stop Bits, Baud Rate, Handshaking and so on) against the settings in the hardware device.**

➤ **Verify the communication port is working properly in Windows.**

1. Close the server program.

☞ If you are using a server that requires a TSR, you will not be able to verify that the port is functioning properly while the TSR is running. Stop all TSRs then continue with this procedure. If you confirm that the port functions properly without the TSR running, change your software interrupt (IRQ) to another number, for example, change 60 to 62.

- ☞ Also, if you are using an AT type computer, two devices cannot share interrupts. Verify that the communication port you are using has a unique interrupt setting.
- 2. On Windows 3.1 or Windows for Workgroups, start the Terminal program. On Windows 95 or Windows NT, start the HyperTerminal program.
- 3. Configure the Terminal (or HyperTerminal) program to use the same communication port with the same settings (baud rate, parity, stop bits and so on) as the hardware device.
- 4. Connect a null modem cable to a second computer's port.
- 5. On the second computer, start and configure the Terminal (or HyperTerminal) program with the same settings as the first computer.
- 6. Verify that you can send data between the two computers.
 - ☞ If you do not have two computers and the computer you are using has another port, start two instances of the Terminal (or HyperTerminal) program with each configured to their own port. Then try communicating between them.
 - ☞ If you have an external modem, connect the modem to the communication port that you are testing to see if you can dial out.
- 7. If the communication port does not appear to be functioning properly, check your environment files (AUTOEXE.BAT, **CONFIG.SYS**, **SYSTEM.INI**, and **WIN.INI**). Look for suspicious programs or drivers that might be taking control of the port or its interrupt before the server is loaded. Always keep your environment files as clean as possible. If you are in doubt about an entry, comment it out.
- 8. If the previous step was unsuccessful, try another communication port or another computer.

Note A common misconception of connecting to a PLC with a DOS program and the same communication port will work in Windows is not the case! Windows is an entirely different environment than DOS.

➤ **What type of UART is on the COM port?**

If it's not a 16550, then you must lower your baud rate to 9600 or slower. Only the 16550 UART can sustain continuous, error free communications at speeds higher than 9600 baud. Other UARTs may work at speeds faster than 9600 baud, but errors may be written to the Wonderware Logger. For example, "Receive Overruns." To determine which UART you have, enter MSD at a DOS prompt, then choose COM Ports from the MSD menu.

➤ **If you are running Windows for Workgroups, verify the following:**

1. Add these lines to the [386Enh] section of your **SYSTEM.INI** file which is located in your \Windows directory:

```
EMMEXCLUDE=A000-EFFF
COMxFIFO=0
```

where x specifies the COM port number. You need to add a separate COMxFIFO line for each serial port using a 16550 UART chip.

2. If you are running Windows for Workgroups (version 3.1 or later), download SERIAL.386 (this fixes a Microsoft bug) from the Wonderware Bulletin Board system (949-727-0726) or from the Wonderware WEB site at: <http://wondertech.wonderware.com>.
 - ☞ The size of this file is 10620 bytes, has a date of 2/17/94, and is a self-extracting zip file called wg1001.exe. It's located in the "PatchFix Library" on the Wonderware BBS in the File Transfers / Patchfix area.
 - ☞ You must be registered to access the Wonderware Web site.
3. If the above numbers, 1. and 2., do not work, verify the value of the ComBoostTime parameter in your **SYSTEM.INI** file. This parameter represents the number of milliseconds a virtual machine processes a COM interrupt. (The default value is 2.) It is not recommended that you change this setting. However, if you are receiving errors such as "Receive Overruns" or "WCRE=-2", try increasing the value to 20.

➤ **Verify the parameters for WWCMTSR.**

Do not specify a receive and/or transmit buffer size of 8!

For example:

Correct:

COM1:0 COM2:1,2048,2048

COM1:0 COM2:1

Incorrect:

COM1:0 COM2:1,8,8

➤ **Does your computer crash when trying to communicate through the COM port?**

If so, verify that each TSR has a unique software interrupt.

- ☞ A utility, ShowSoft, is available on the Knowledge Base CD that can assist in determining the available software interrupts.

➤ **Does your computer lock up?**

Verify the COM port's IRQ's do not conflict with each other or with other communication boards in the computer.

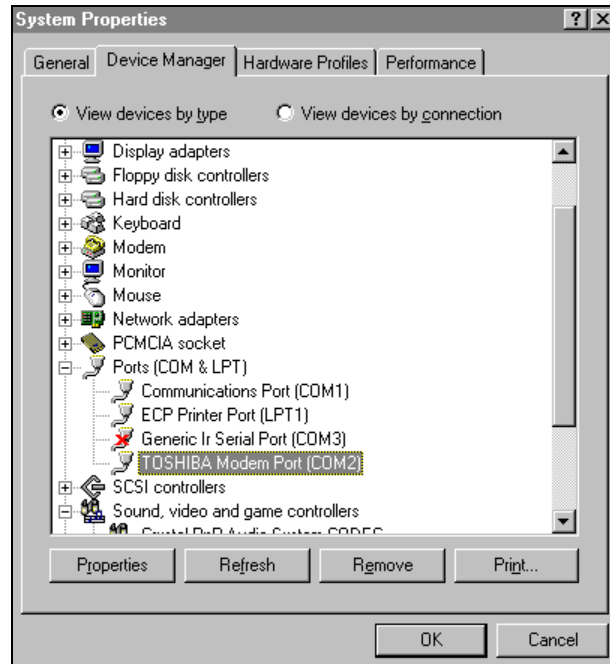
➤ **If the PLC or field device has more than one COM port, verify the connection to the correct port.**

The COM port on your computer uses the RS-232 hardware communication standard and connects the cable from the COM port to an RS-232 compliant device.

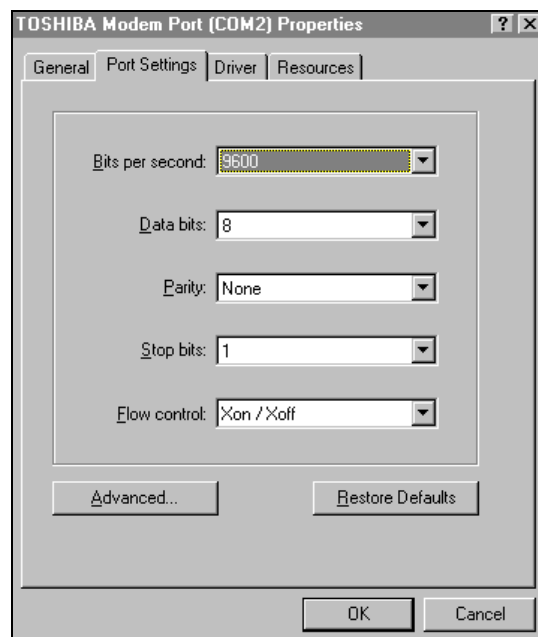
Note: To connect to an RS-422 or RS-485 port on the PLC, you need an RS-232 to RS-422/485 conversion device.

If possible, use an external converter instead of a board-based converter that plugs into a slot in the computer. A board-based converter is difficult to get working for inexperienced users. If a board-based converter is not set up properly, it can conflict with other communication boards in the computer such as, internal modems.

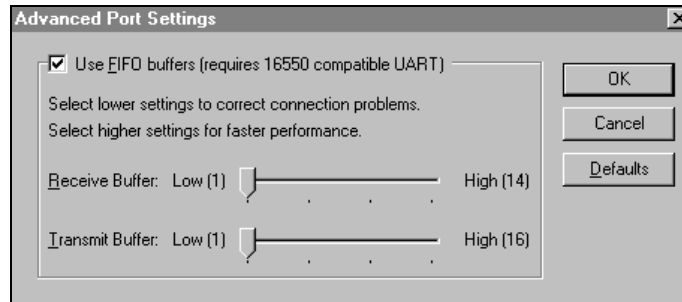
- **If you are using the Windows 95 or Windows 98 operating system, verify the following:**
1. Click Start on the Windows taskbar. Point to Settings, then click Control Panel in the menu. The Control Panel dialog box will appear.
 2. Double-click the System icon. The System Properties dialog box will appear. Click the Device Manager tab and select the COM port that you are using for the server. For example:



3. Click Properties. The Properties dialog box will appear. Click the Port Settings tab.



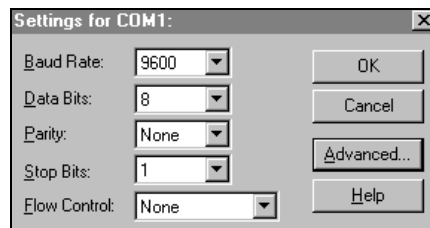
- Click Advanced. The Advanced Port Settings dialog box appears:



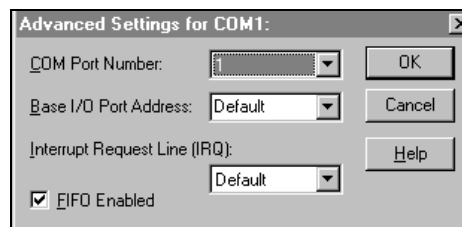
- Lowering the default Receive Buffer and Transmit Buffer settings to their minimum may solve I/O communication problems for portable computers (notebook or laptops) and framing errors for standard computers.
- If using a 16550 UART chip, select the Use FIFO buffers (requires 16550 compatible UART) option. If you are not using a UART chip, make sure this option is not selected.

➤ **If you are using the Windows NT operating system, verify the following:**

- Click Start on the Windows taskbar. Point to Settings, then click Control Panel in the menu. The Control Panel dialog box will appear.
- Double-click the Ports icon, the Ports dialog box will appear.
- Select a port and click the Settings button. The Settings for COMx dialog box appears:

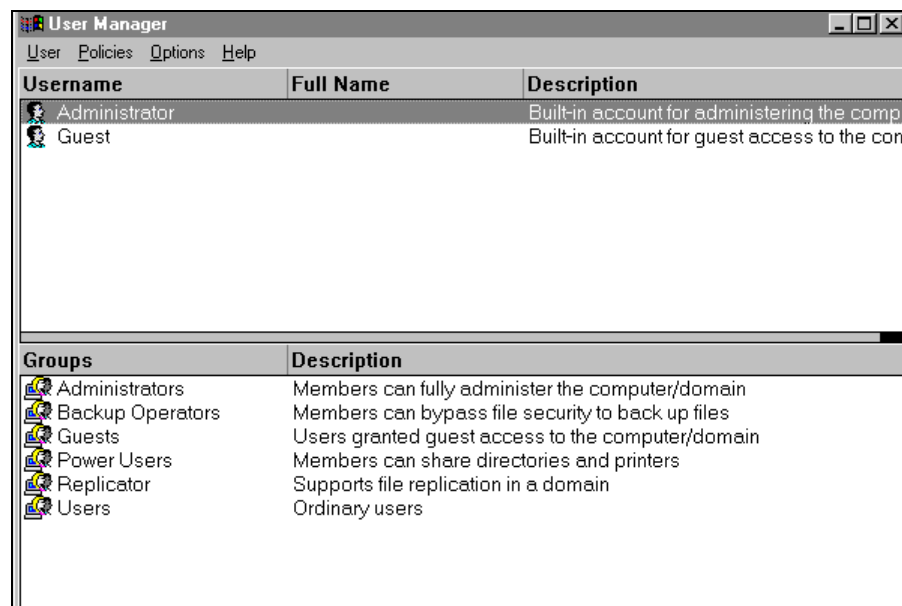


- Click Advanced. The Advanced Settings for COMx dialog box appears:



- Lowering the setting for the Interrupt Request Line (IRQ) value to the minimum may solve I/O communication problems for portable computers (notebook or laptops) and framing errors for standard computers.
- If you are using a 16550 UART chip, select the FIFO Enabled option. If you are not using a UART chip, make sure this option is not selected.

- **How long is your RS-232 cable?**
Fifteen meters (fifty feet) is the maximum practical length for the RS-232 standard.
 - **Try using a different COM port for the I/O Server.**
 - **If you are installing an I/O Server or configuring a board-based I/O Server on a computer running on the Windows NT operating system, log on with Administrator privileges.**
 - ☞ Without Administrator privileges, the server and Server Install program cannot make the necessary edits to the Windows NT Registry during installation or board configuration of the server.
1. Click Start on the Windows taskbar. Point to Programs, then to Administrative Tools (Common), and click User Manager in the menu. The User Manager dialog box will appear:



2. Double-click the Username you typed in during log on.
 3. If the User Properties dialog box does not appear, you do not have Administrator privileges.
 4. If the User Properties dialog box does appear, click on the Groups button and verify "Administrators" is in the "Member of" list.
- **If you experience occasional or random communication errors in the Wonderware Logger, such as "Response Timeouts," check for noise.**
Do the physical cables run past any known noise sources such as photocopiers, fluorescent lamps, fans, pumps, motors or generators? Are the cables properly shielded from its environment? With radio modems and satellite link ups, occasional communications errors in the Wonderware Logger are normal and to be expected as long as they do not adversely impact the flow of data.

- **Increase the Reply Timeout setting in the I/O Server to a value between 5 and 10 seconds.**

Not allowing the PLC or field device enough time to respond to the server's request for data may result in communication errors.

- **Verify the PLC is properly configured and the cable is good by using the programming software for the PLC.**
 1. Connect to the PLC with the programming software. The connection must be through the same port and cable. Go into the programming software configuration and write down what the communications parameters are (baud rates, routes, node number, error checking, etc.).
 2. Close the programming software. Open the I/O Server and verify the communications settings are the same.
 3. Poke data into the PLC with InTouch or WWClient.
 4. Shut down the server and use the programming software to verify that the values were correctly poked.

☞ Performance of this test depends upon the type of PLC you are using.

- **Reinstall the I/O Server and verify that you are using the latest version.**

Wonderware is continually improving our servers and using the latest version will guarantee the best results.

☞ New versions of the Wonderware I/O Servers are released regularly on the Knowledge Base CD. These are available to Comprehensive Support customers on the Wonderware Bulletin Board System (949-727-0726) or from the Wonderware WEB site at: <http://wondertech.wonderware.com>.

- **Move the I/O Server's configuration file to another location on the computer's hard drive. This will clear all configuration for the I/O Server, then reconfigure the I/O Server.**

☞ Wonderware server configuration files are typically the exact same name as the server's executable name with the .cfg extension. For example, **OMRONFO.cfg**. Refer to the Configuration File section of the specific server user's guide for the exact name of the configuration file.

- **If possible, reinstall the Windows operating system.**

Files installed earlier on your computer or the NT registry may have been corrupted or accidentally modified.

- **If these troubleshooting suggestions do not solve your problem, there may be a problem with your computer. There are many subtle differences between the various computer hardware brands. Using a computer that is a different brand and meets the following criteria:**

Select a different PC manufacturer and if this is not possible, try a different PC model from the same manufacturer.

The computer can not use an OEM (Original Equipment Manufacturer) version of Microsoft Windows. We highly recommend using only a Microsoft Windows product. Contact your vendor to determine if installing an off-the-shelf copy of Microsoft Windows will cause any problems.

- **If you feel you have tested all possible situations that may be causing your failed I/O communications, contact your local Wonderware distributor for technical support.**

↪ For more information on obtaining technical support, see your online *FactorySuite System Administrator's Guide*.

Special Wonderware Logger Messages

The following messages may appear in the Wonderware Logger. They can be very useful in debugging communication problems.

➤ **The following logger messages can occur in error situations if log errors (default) is selected in the logger settings.**

- **ERROR: poking item: <item name>(<topic name>)**

Writing a value to an item failed.

- **POKE: invalid value, clamp at high limit for item: <item name> on <topic name>**

- **POKE: invalid value, clamp at low limit for item: <item name> on <topic name>**

An invalid value has been specified for poking data. Value is clamped and transferred to the OPC server.

- **POKE ERROR: invalid value, cannot convert for item : <item name> on <topic name>**

An invalid value has been specified for poking data. Value cannot be converted and write operation is aborted. After all write operations are completed the "writecomplete" item will go to -1.

- **Error invalid topic name: <topic name>**

Topic name does not exist. Change topic name or configure correct topic.

- **Invalid item name: <item name> (<topic name>)**

Item name does not exist. Check for correct item name.

- **Write complete (ERROR) – item: <item name> on <topic name>**

A write operation has failed on item with <item name>.

- **New status (<topic name>): 0 (ERROR)**

Connection to the OPC server has gone bad.

- **New status (<topic name>): 1 (GOOD)**

Connection to the OPC server has gone good again.

- **ERROR: (hr=<function result>) cannot create OPC server instance of <OPC server>.**

OPCLink could not create an instance of the OPC server. Check DCOM settings if the OPC server is on a remote machine and check whether the OPC server was installed properly. Lookup the system message for <function result> to get a detailed error message.

- **ERROR: (hr=<function result>) attaching the OPC server failed. You may not have registered the Proxy dll of <OPC server>**

Something went wrong in calling the needed interfaces for an OPC connection. Check the installation of the OPC server and whether the opcproxy.dll was registered correctly.

- **ERROR: (hr=<function result>) adding OPC group <topic name>**

The OPC server could not add the OPC group <topic name>. Some OPC servers don't allow non-alphanumeric chars in their groupnames. Choose another topic name for this topic.

- **ERROR: cannot add OPC item: <OPC item name> to group on topic: <topic name>**

The OPC server cannot advise the item <OPC item name> within the connected OPC server. This message is followed by a more detailed error message. One reason could be that the specified <OPC item name> is not known in the OPC servers address space.

➤ **The following logger messages can occur in normal operation mode if log trace is selected in the logger settings. These messages show the general activity of OPCLink. Note: it can slow down the performance of OPCLink.**

- **TRACE: Topic allocated: <topic name>**

A topic has been created (client has connected on this topic).

- **TRACE: Free topic: <topic name>**

A topic has been freed (client has disconnected on this topic).

- **TRACE: Item <item name> created on <topic name>**

An item has been created in OPCLink.

- **Activating item <item name> on <topic name>**

An item has been activated in OPCLink. A client has advised/requested data.

- **Deactivating item <item name> on <topic name>**

An item has been deactivated in OPCLink. A client has unadvised data.

- **New value for device - item: <item name> on: <topic name>**

A client performed a write operation on an item.

- **Write complete (OK) - item: <item name> on: <topic name>**

The write operation was completed successfully on the protocol. The OPC server has confirmed the reception of data.

- **TRACE: successfully connected to OPC server: <OPC server>**

The sequence for connecting OPC server <OPC server> was successful.

➤ **The following logger messages can occur in normal operation mode if log all is selected in the logger settings. These messages show all of OPCLinks activities. Note: it can slow down the performance of OPCLink.**

- **New <data type> value: <value> for item: <item name> on <topic name>**

An item was updated with a new value from the OPC server.

- **Write complete item: 0**

A client has initiated write operation and OPCLink signals the processing of poke values by setting the "write complete" item to zero

- **Write complete item: 1**

All requested pokes from clients have been processed and there were no errors. OPCLink signals the successful processing of poke values by setting the "write complete" item to one.

- **Write complete item: -1**

All requested pokes from clients have been processed and there were errors. OPCLink signals that there were at least one item that could not be poked by setting the "write complete" item to minus one.

- **The following logger messages can occur in normal operation mode if "show protocol" is selected in the logger settings. These messages show the sent and received messages.**

Note: it can slow down the performance of OPCLink.

OPC Data Access 2.0:

- **TRACE: OPCCallback::OnDataChange called from "<OPC server>" (dwTransactionID=<transaction id> grphandle=<grouphandle> masterquality=<masterquality> mastererror=<mastererror> count=<itemcount>)**

The OPC server sent changed data to OPCLink.

- **TRACE: processing datetime (DA 2.0) from "<OPC server>" (dwTransactionID=<transaction id> grphandle=<grouphandle> masterquality=<masterquality> mastererror=<mastererror> count=<itemcount>)**

The received data from an OPC server was processed (used to update all served items).

- **TRACE: OPCCallback: OnWriteComplete called from "<OPC server>" (dwTransactionID=<transaction id> grphandle=<grouphandle> mastererror=<mastererror> count=<itemcount>)**

The OPC server sent a writecomplete notification to OPCLink.

- **TRACE: processing writecomplete (DA 2.0) from "<OPC server>" (dwTransactionID=<transaction id> grphandle=<grouphandle> mastererror=<mastererror> count=<itemcount>)**

A received writecomplete notification from an OPC server was processed.

OPC Data Access 1.0a:

- **TRACE: datetime (DA 1.0a) received from "<OPC server>" (dwSize=<input stream size> dwItemCount=<itemcount> hClientGroup=<client group handle> dwTransactionID=<transaction id> hrStatus=<function result>)**

The OPC server sent changed data to OPCLink.

- **TRACE: processing datetime (DA 1.0a) from "<OPC server>"**
(dwSize=<input stream size> dwItemCount=<itemcount>
hClientGroup=<client group handle> dwTransactionID=<transaction id>
hrStatus=<function result>)

The received data from an OPC server was processed (used to update all served items).

- **TRACE: writecomplete (DA 1.0a) received from "<OPC server>"**
(dwItemCount=<itemcount> hClientGroup=<client group handle>
dwTransactionID=<transaction id> hrStatus=<function result>)

The OPC server sent a writecomplete notification to OPCLink.

- **TRACE: processing writecomplete (DA 1.0a) from "<OPC server>"**
(dwItemCount=<itemcount> hClientGroup=<client group handle>
dwTransactionID=<transaction id> hrStatus=<function result>)

A received writecomplete notification from an OPC server was processed.

- **confirmation received for poking item: <OPC item name> (<topicname>)**

A writecomplete notification for item <OPC item name> has been received.

Declarations:

<OPC item name> is the full name of an OPC item (including the OPC path).

<item name> is the name of the item which could be advised in a topic (including the prefix but not the OPC path).