

ArchestrA Protocols Guide

Invensys Systems, Inc.

Revision A
Last Revision: 6/29/07



Copyright

© 2007 Invensys Systems, Inc. All Rights Reserved.

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 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 the 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.

Invensys Systems, Inc.
26561 Rancho Parkway South
Lake Forest, CA 92630 U.S.A.
(949) 727-3200

<http://www.wonderware.com>

For comments or suggestions about the product documentation, send an e-mail message to productdocs@wonderware.com.

Trademarks

All terms mentioned in this documentation 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 documentation should not be regarded as affecting the validity of any trademark or service mark.

Alarm Logger, ActiveFactory, ArcestrA, Avantis, DBDump, DBLoad, DT Analyst, FactoryFocus, FactoryOffice, FactorySuite, FactorySuite A², InBatch, InControl, IndustrialRAD, IndustrialSQL Server, InTouch, 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

Welcome.....	5
Documentation Conventions.....	5
Technical Support	6
Chapter 1 SuiteLink	7
SuiteLink Features	7
Hardware and Software Requirements.....	8
Time Stamping	8
Data Quality	8
Quality Bit Field.....	9
Substatus Bit Field.....	10
Limit Bit Field	12
Wonderware I/O Server Quality Reporting	13
Good.....	13
Clamped High.....	13
Clamped Low	14
Cannot Convert	14
Cannot Access Point.....	15
Communications Failed	15

Chapter 2 DDE, FastDDE, and NetDDE	17
DDE.....	17
FastDDE	18
NetDDE	18
Microsoft Windows NT Operating System and NetDDE.....	18
Windows NT Networking Support	19
Included Extensions	19
System Requirements	20
Interface Configuration.....	20
Configuring DDE Share Security	21
DDE Shares	21
Security Configurations	23
Changing the Default DDE Share Security	24
Customized DDE Shares.....	26
Configuring the WinSock Interface.....	28
Installation Requirements for WinSock.....	28
Accessing the WinSock Interface Configurations.....	29
Configuring WWSOCK	29
WinSock Error Messages	32
NetDDE Helper Service	40
Services as NetDDE Clients	40
Index	41

Welcome

This guide provides background information on the primary communication protocols used between components of Wonderware products.

A protocol is a set of rules and standards for enabling computers to connect and exchange data over a network.

This guide includes information on setting up and using some of these protocols.

You can view this document online or you can print it, in part or whole, by using the Adobe Acrobat Reader's print facility.

Documentation Conventions

This documentation uses the following conventions:

Convention	Used for
Initial Capitals	Paths and file names.
Bold	Menus, commands, dialog box names, and dialog box options.
Monospace	Code samples and display text.

Technical Support

Wonderware Technical Support offers a variety of support options to answer any questions on Wonderware products and their implementation.

Before you contact Technical Support, refer to the relevant section(s) in this documentation for a possible solution to the problem. If you need to contact technical support for help, have the following information ready:

- The type and version of the operating system you are using. For example, Microsoft Windows XP, SP1.
- Details of how to recreate the problem.
- The exact wording of the error messages you saw.
- Any relevant output listing from the Log Viewer or any other diagnostic applications.
- Details of what you did to try to solve the problem(s) and your results.
- If known, the Wonderware Technical Support case number assigned to your problem, if this is an ongoing problem.

Chapter 1

SuiteLink

Wonderware SuiteLink uses a TCP/IP based communication protocol. SuiteLink 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 or later.

SuiteLink is not a replacement for DDE, FastDDE, or NetDDE. Each connection between a client and a server depends on your network situation.

SuiteLink Features

SuiteLink is designed specifically for high speed industrial applications 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, including server loading, computer resource consumption, and network transport, are made accessible through the Microsoft Windows NT operating system performance monitor. This feature is critical for the maintenance of distributed industrial networks.
- Consistent high data volumes can be maintained between applications when 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. You do not have to create shares for SuiteLink I/O Servers.

Hardware and Software Requirements

You must have the following installed to use SuiteLink for data communications.

- Windows NT 4.0 or later
- TCP/IP installed and configured

Time Stamping

SuiteLink allows for the passing of time stamping information with process data. The SuiteLink time stamp is a 64-bit data structure representing the number of 100-nanosecond intervals since January 1, 1901 in Greenwich Mean Time. This matches the Microsoft FILETIME specification. Conversion to and from local time is the responsibility of the application layer. All time stamps carried in the SuiteLink protocol are in GMT.

Data Quality

When a data value is acquired by a Wonderware I/O Server using SuiteLink, a 2-byte quality flag is assigned to the value. This flag represents the quality state for an item's data value. The lower eight bits of the quality flag consists of three bit fields: Quality (Q), Substatus (S), and Limit (L). (The high eight bits are undefined.)

These three bit fields are arranged as follows:

QQSSSSL

The bit assignments in these two bytes complies fully with the OPC specification for data quality. Each of these bit fields is described in the following table:

Bit Field	Description
Quality	<p>Determines the status of the data value. Data values can be bad (0), uncertain (1), or good (3).</p> <p>A server that does not support quality information must return 3 (Good). It is also acceptable for a server to simply return Bad or Good (0 or 3) and to always return 0 for substatus and limit.</p>
Substatus	<p>Used to further describe the overall quality of the value. For example, if the quality for a particular value is bad, then the Substatus field carries a number associated with a reason that the value was bad, such as a device failure or a configuration error.</p> <p>Servers that do not support substatus should return 0. Note that an 'old' value may be returned with the Quality set to BAD (0) and the substatus set to 5. This is for consistency with the Fieldbus Specification.</p>
Limit	<p>Returns information on the limits of the value. For example: Is it clamped at a high limit? The Limit bit field is valid regardless of the Quality and Substatus. In some cases such as Sensor Failure it can provide useful diagnostic information.</p>

Quality Bit Field

The following table describes the data quality values for the Quality bit field.

Value	Quality	Description
0	Bad	Value is not useful for reasons indicated by the substatus.
1	Uncertain	The quality of the value is uncertain for reasons indicated by the substatus.
2	N/A	Not used by OPC.
3	Good	The Quality of the value is Good.

Substatus Bit Field

The layout of this field depends on the value of the Quality bit field. The following table describes the “Bad” quality values for the substatus bit field.

Value	Quality	Description
0	Non-specific	The value is bad but no specific reason is known.
1	Configuration Error	There is some server specific problem with the configuration. For example the item is question has been deleted from the configuration.
2	Not Connected	The input is required to be logically connected to something but is not.
3	Device Failure	A device failure has been detected.
4	Sensor Failure	A sensor failure had been detected (the 'Limits' field can provide additional diagnostic information in some situations.)
5	Last Known Value	Communications have failed. However, the last known value is available. Note that the 'age' of the value can be determined from the OPCITEMSTATE.
6	Comm Failure	Communications have failed. There is no last known value is available.
7	Out of Service	The block is off scan or otherwise locked (e.g. by a configuration builder).
8-15	N/A	Not used by WW/OPC.

The following table describes the "Uncertain" quality values for the substatus bit field.

Value	Quality	Description
0	Non-specific	There is no specific reason why the value is uncertain.
1	Last Usable Value	Whatever was writing this value has stopped doing so. The returned value should be regarded as 'stale'. Note that this differs from a BAD value with substatus 5 (Last Known Value). That status is associated specifically with a detectable communications error on a 'fetched' value. This error is associated with the failure of some external source to 'put' something into the value within an acceptable period of time. Note that the 'age' of the value can be determined from the OPCITEMSTATE.
2-3	N/A	Not used by OPC.
4	Sensor Not Accurate	Either the value has 'pegged' at one of the sensor limits (in which case the limit field should be set to 1 or 2) or the sensor is otherwise known to be out of calibration via some form of internal diagnostics (in which case the limit field should be 0).
5	Engineering Units Exceeded	The returned value is outside the limits defined for this parameter. Note that in this case (per the Fieldbus Specification) the 'Limits' field indicates which limit has been exceeded but does NOT necessarily imply that the value can not move farther out of range.
6	Sub-Normal	The value is derived from multiple sources and has less than the required number of Good sources.
7-15	N/A	Not used by WW/OPC.

The following table describes the "Good" quality values for the substatus bit field.

Value	Quality	Description
0	Non-specific	The value is good. There are no special conditions.
1-5	N/A	Not used by WW/OPC.
6	Local Override	The value has been Overridden. Typically this means the input has been disconnected and a manually entered value has been 'forced'.
7-15	N/A	Not used by WW/OPC.

Limit Bit Field

The following table describes the data quality values for the Limit bit field.

Note The Limit bit field is valid regardless of the Quality and Substatus. In some cases, such as Sensor Failure, it can provide useful diagnostic information.

Value	Quality	Description
0	Not Limited	The value is free to move up or down.
1	Low Limited	The value has "clamped" at some lower limit.
2	High Limited	The value has "clamped" at some high limit.
3	Constant	The value is a constant and can not move.

Wonderware I/O Server Quality Reporting

Wonderware I/O Servers can report six mutually exclusive states of data quality back to their clients. Their data quality states are:

- 1 Good
- 2 Clamped High
- 3 Clamped Low
- 4 Cannot Convert
- 5 Cannot Access Point
- 6 Communications Failed

Good

In order for the "Good" quality state to be reported, the following must conditions must be met:

- The communications link was verified.
- The PLC understood the poll request and returned a valid response packet.
- If a write occurred, there were no errors during the write process.
- There were no conversion problems with the data contained in the response packet.

The I/O Server returns a value of 0x00C0 for the Good quality state.

Clamped High

The "Clamped High" quality state is reported if it was necessary to clamp the intended value to a limit because the value was larger than the maximum allowed.

The communications link was verified and the PLC understood the poll request and returned a valid response packet. The register was read or written without error.

Note In the case of a string, it is truncated.

The I/O Server returns a value of 0x0056 for the Clamped High quality state.

Clamped Low

The "Clamped Low" quality state is reported if it was necessary to clamp the intended value to a limit because the value was smaller than the minimum allowed.

The communications link was verified and the PLC understood the poll request and returned a valid response packet. The register was read or written without error.

The I/O Server returns a value of 0x0055 for the Clamped Low quality state.

Cannot Convert

The "Cannot Convert" quality state is reported if a conversion error occurs. The communications link was verified and the PLC understood the poll request and returned a valid response packet. Causes for conversion errors include, but are not limited to:

- The data from the PLC could not be converted into the desired format.
- The server may return a constant in place of the data, or return quality information alone.
- The data is not usable.
- It is not known whether the value is too large or too small.
- The data returned from the PLC is of the incorrect data type.
- A floating-point number is returned, but is not a value (that is, it is not a number).

The I/O Server returns a value of 0x0040 for the Cannot Convert quality state.

Cannot Access Point

The "Cannot Access Point" quality state occurs if the PLC reported that it could not access the requested point or that the data is not usable. The communications link was verified and the PLC understood the poll request and returned a valid response packet. Possibilities for lack of accessibility include, but are not limited to:

- The item does not exist in PLC memory.
- The item is not currently available (locked in some way due to resource contention).
- The item is not of the correct format/data type.
- A write attempt was made, but the item is read-only.

In most cases, a group of items is affected when one item is invalid. This is due to the block-polling scheme used by servers. For example, if one item in a block of 10 is invalid, then the entire block is marked invalid by the PLC. The server reports invalid quality for all items in the block.

The I/O Server returns a value of 0x0004 for the Cannot Access Point quality state.

Communications Failed

The "Communications Failed" quality state is reported if any one of the following occurs:

- Data communications are down.
- The topic is in slow poll (or equivalent) mode.
- There have been no link validating messages.
- Lack of resources in the server (for example, a TSR or driver can not allocate memory).
- Lack of resources in the communications link.
- The communications link is offline.
- All communications channels are in use.
- The network is unable to route the message to the PLC.

The I/O Server returns a value of 0x0018 for the Communications Failed quality state.

Chapter 2

DDE, FastDDE, and NetDDE

The DDE protocols used by Wonderware products are DDE, FastDDE, and NetDDE.

DDE

Dynamic Data Exchange (DDE) is a Microsoft communications protocol that lets applications in the Windows environment send/receive data and instructions to/from each other. DDE implements a client-server relationship between two concurrently running applications. The server application provides 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.

Note InBatch does not support DDE/NetDDE connections to the I/O Servers, including InControl. For those connections, SuiteLink must be used.

Requests for data can be one of two types: one-time requests or permanent data links. With one-time requests, the client program requests a "snapshot" of the desired data from the server application. An example of a one-time request is a program, such as Excel, running a report-generating macro. The macro opens a channel to another application, requests specific data, closes the channel, and uses the data to generate the report.

Permanent data links are called "hot links." When a client application sets up a hot link to another application it requests the server application to notify the client whenever a specific item's data value changes. Permanent data links remain active until either the client or server program terminates the link or the conversation. Permanent data links are a very efficient means of exchanging data because, once the link has been established, no communication occurs until the specified data value changes. ArcestrA components can use DDE to communicate with I/O device drivers and other DDE application programs.

FastDDE

FastDDE provides a means of packing many Wonderware DDE messages into a single Microsoft DDE message. Message packing improves efficiency and performance by reducing the number of DDE transactions required between client and server.

NetDDE

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 PCs connected via LAN or modem and DDE-aware applications running on non-PC based platforms under operating environments such as VMS and UNIX.

NetDDE is not supported by the Windows Vista operating system.

Microsoft Windows NT Operating System and NetDDE

Microsoft's version of NetDDE is included in the Microsoft Windows NT operating system. To install Microsoft Windows NT operating system on the local node, refer to your Microsoft Windows NT System Guide. If you are a new Microsoft Windows NT operating system user, it is recommended that you read this guide to familiarize yourself with Microsoft Windows NT operating system and NetDDE's role in providing DDE connectivity between various operating environments.

Windows NT Networking Support

NetDDE, included with the Microsoft Windows NT operating system, runs transparently to the user and expands the standard Microsoft Windows DDE (Dynamic Data Exchange) functionality to include communication over various networks. To use it, two or more IBM compatible PCs running Microsoft Windows NT operating system is required. Microsoft Windows NT operating system must be installed on all network nodes between which DDE data is to be exchanged.

The Microsoft Windows NT operating system includes built-in networking support, a component of which is NetDDE with the NetBIOS interface. The networking software chosen for installation on the local node depends on what other PCs and workstations you intend to connect with Microsoft Windows NT operating system.

The WinSock network interface configuration extension allows easy configuration of the WinSock interface without having to access the Windows NT Registry Editor.

No networking software package is required for stand-alone remote PCs that dial in to a network system.

Included Extensions

NetDDE Extensions for Windows NT operating system includes the DDE Share Security extension and the WinSock network interface extension.

The DDE Share Security extension lets you configure a DDE Share Security policy and administer the configured DDE Share Security policy when remote workstations attempt to gain access to DDE data available at the local node. With this extension, configurations can be made easily without having to use the Windows NT operating system standard security dialogs.

For more information, see [Configuring DDE Share Security](#) on page 21.

The WinSock network interface extension lets you easily configure the WinSock interface without having to access the Windows NT Registry Editor.

For more information, see [Configuring the WinSock Interface](#) on page 28.

System Requirements

To install NetDDE Extensions for Windows NT operating system, the following minimum system requirements must be met:

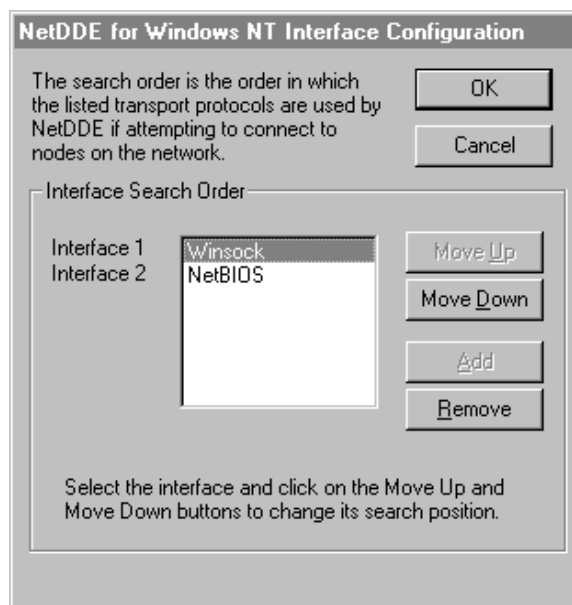
- Microsoft Windows NT operating system 3.51 or later. If running InTouch you must have Microsoft Windows NT 4.0 operating system installed.
- TCP/IP Network Protocol installed.

Interface Configuration

When using NetDDE for Windows NT operating system, the interface search order needs to be configured.

To configure the interface:

- 1 On the **Configure** menu, click **Interfaces**. The **NetDDE for Windows NT Interface Configuration** dialog box appears.



- 2 Click the **Move Down** button to switch the search order for the interface. For example, Interface 1, which is the first search order listed is Winsock. When the **Move Down** button is clicked the order changes. Interface 1 is now listed as NetBIOS and will be the first interface searched.
- 3 Click **OK** to accept the new search order and close the dialog box.

Configuring DDE Share Security

Because the Microsoft Windows NT operating system allows access by remote workstations to DDE data stored on the local node, a security policy is created to prevent unauthorized access. With a DDE Share Security policy in place, access must be explicitly granted to shared DDE data available at the local node. Likewise, a remote workstation that wants access to secured DDE data must be able to respond appropriately to the requirements exacted by the DDE Share Security subsystem in Microsoft Windows NT operating system.

The DDE Share Security extension lets you configure a DDE Share Security policy and administer the configured DDE Share Security policy when remote workstations attempt to gain access to DDE data available at the local node. Using NetDDE Extensions makes it easier to configure the security issues than using Microsoft Windows NT operating system standard security dialogs.

DDE Shares

DDE shares correspond to DDE data maintained by DDE-aware server applications. Some applications, such as Wonderware InTouch and Microsoft Excel, can be both DDE clients and DDE servers on the local node. DDE shares are defined in the DDE shares database maintained by the operating system on each node. The DDE shares database stores the name of each application and topic pair that can be referred by a remote node in a Microsoft Windows NT operating system conversation. The DDE shares database also identifies the security permission levels for each DDE share that defines the access nodes available to that share.

A DDE share can be created for each DDE topic supported by a DDE-aware application. Or, a "wild card" DDE share, specifying "*" as the topic name, can be defined to enable access through Microsoft Windows NT operating system to all topics supported by the given DDE-aware application.

DDE Share Permission Levels

A DDE share representing an application and topic pair that has been explicitly defined in the DDE shares database is referred to as a custom DDE share. Each custom DDE share defined in the DDE shares database has a specific permission level assigned to it. The permission level assigned to the DDE share determines what type of access is granted by Microsoft Windows NT operating system to remote workstations.

The following permission levels can be assigned to a DDE share:

Permission Level	Description
Full Access	Allows access to the specified application and topic from all remote workstations without any restrictions.
No Access	Allows no access to the specified application and topic from any remote workstation.
Read-Only	Allows only DDE Request and DDE Advise access to the specified application and topic from any remote workstation. No DDE Poke or DDE Execute access is allowed.
Permissions	Allows only DDE clients with the correct permission level to access data at the selected node. When selected the Permissions button will appear.

For more information, see Security Configurations on page 23.

Default DDE Share Security

Default DDE Share Security is applied to all application and topic pairs that are not explicitly itemized in the DDE shares database. When Microsoft Windows NT operating system receives an initiate to a specific application and topic, it first interrogates the DDE shares database to see if specific security permission levels have been assigned. If a share for the specific application and topic pair has not been defined, Microsoft Windows NT operating system uses the security permission levels assigned to the default DDE share and apply them to the initiated DDE conversation. The default DDE share can be assigned the same security permission levels as custom DDE shares.

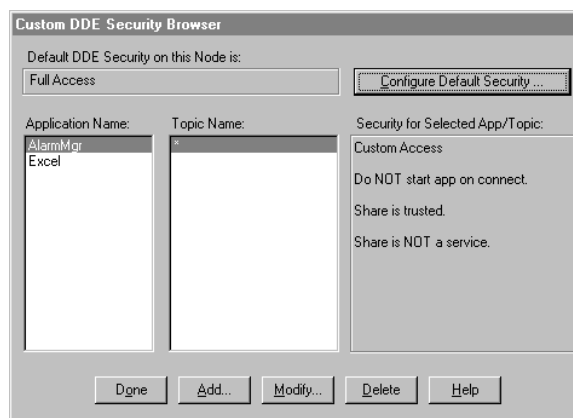
Security Configurations

To access DDE Share Security configuration

- 1 Execute **NetDDE Extensions**. The **Wonderware NetDDE Extensions** dialog box appears.



- 2 On the **Configure** menu, click **Security**. The **Custom DDE Security Browser** dialog box appears.



- 3 In the **Default DDE Security on this Node** box, the default of **Full Access** appears. The default security level can be changed by configuring a new default setting.

Changing the Default DDE Share Security

The default DDE share can be defined and modified from the **Configure Default Security** browser.

To modify the default security level:

- 1 On the **Custom DDE Security Browser** screen, click the **Configure Default Security** button. The **Default DDE Security** dialog box appears.

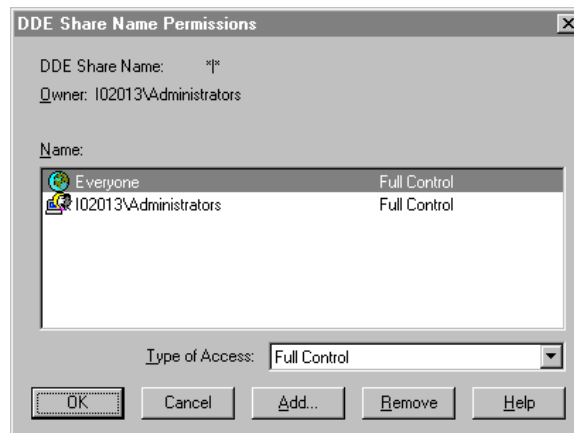


- 2 Select a **Default Access** option for the node. Descriptions for each option appear in the field to the right of the option when selected.

By default, the Microsoft Windows NT operating system assumes **Full Access** for the node and creates a corresponding default DDE share when it is activated (unless a DDE share already exists in the DDE shares database). This ensures a smooth transition from the Microsoft Windows NT operating system environment, letting you become gradually accustomed to securing DDE data in this manner.

For more information, see **DDE Share Permission Levels** on page 22.

- 3 If the **Permissions** Default Access has been selected, the **Permissions** button appears. Select this button to view or change directory permissions. The **DDE Share Name Permissions** dialog box appears.



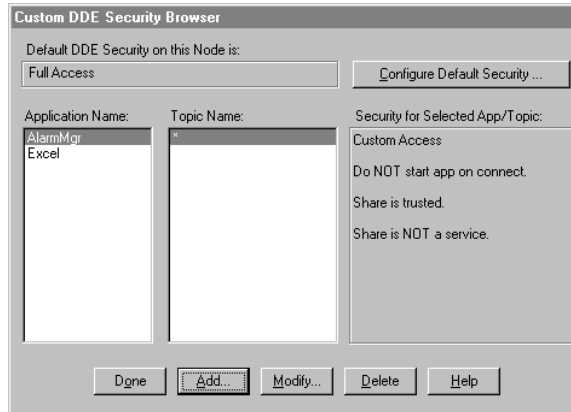
- 4 In the **Type of Access** box, select the access level for the DDE Share Name Permission.
- 5 Click **OK** to update the permission and close the **DDE Share Name Permissions** dialog box.
For complete details on Access Types, see the Microsoft Windows NT documentation.
- 6 Select one of the following options.
- **Start on Connect**
To start an application that is not already running on connection from the remote node. This option is disabled if the **Service** option is selected.
 - **Trust Share**
Lets other users access the share. Otherwise, only local applications can be accessed. This option can be used to remove all access to the share without having to delete the share.
 - **Service**
Select when the share is an installed Microsoft Windows NT operating system service and is started at system boot time. This option is disabled if the **Start on Connect** option is selected.
- 7 Click **OK** to accept changes and return to the **Custom DDE Security Browser** dialog box.

Customized DDE Shares

The DDE shares database can be edited using the DDE Share Security extension. New custom DDE shares can be added and existing shares modified. The **Custom DDE Security Browser** lets you view existing custom DDE shares and initiate actions to add, modify, or delete selected DDE shares.

To add custom DDE shares

- 1 On the **Configure** menu, click **Security**. The **Custom DDE Security Browser** dialog box appears.

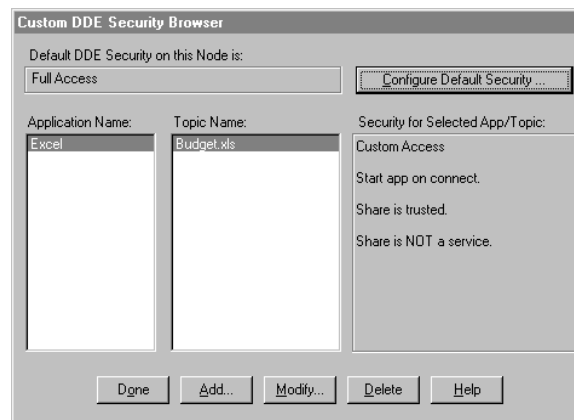


- 2 Click **Add**. The **Custom DDE Security Configuration** dialog box appears.



- 3 In the **Application** box, type the name of the application; for example, Excel.
- 4 Select **Topic Name** and type the name of the application in the box. A security level can now be assigned to this topic. For example, Budget.xls is the topic name of the application Excel and now requires a password to read and write to the topic.
- 5 Select **Any Topic** to allow access to all topics in your application.

- 6 In the **Select Required Access Security** group, select the security permission level to assign to the custom DDE share. Descriptions for each option will appear in the field to the right of the option when selected.
For more information, see DDE Share Permission Levels on page 22.
- 7 In the **Application Options** group, select from the following options:
 - **Start on Connect**
To start an application that is not already running on connection from the remote node. This option is disabled if the **Service** option is selected.
 - **Trust Share**
Lets other users access the share. Otherwise, only local applications can be accessed. This option can be used to remove all access to the share without having to delete the share.
 - **Service**
Select when the share is an installed Microsoft Windows NT operating system service and is started at system boot time. This option is disabled if the **Start on Connect** option is selected.
- 8 Click **Cancel** to close the dialog box, without saving changes.
- 9 Click **OK** to accept changes and return to the **Custom DDE Security Browser** dialog box, which appears with the added Application Name and Topic Name.



- 10 To view each node's security permission level assigned to a custom DDE share, select the application from the **Application Name** field, and then select the topic from the **Topic Name** field.

All topics in this list are associated with the selected application. The security level appears in the **Security for Selected App/Topic** field.

- 11 Click **Done** to close the dialog box and save security changes.

To modify a DDE Share

- 1 On the **Custom DDE Security Browser** dialog, select the Application Name and Topic Name you need to modify. The **Custom DDE Security Configuration** screen appears.
- 2 Make the necessary modifications to the security access and application options.
- 3 Click **OK** to return to the **Custom DDE Security Browser** dialog box.

To delete a DDE Share

- 1 On the **Custom DDE Security Browser** dialog, select the Application Name and Topic Name you want to delete. A message box will appear confirming your delete request.
- 2 Click **OK**. The share is removed from the DDE shares database and the displays in the **Custom DDE Security Browser** dialog are updated.

Configuring the WinSock Interface

NetDDE Extensions for Microsoft Windows NT operating system allows viewing or configuring of the WinSock interface parameters without having to access the Windows NT Registry Editor.

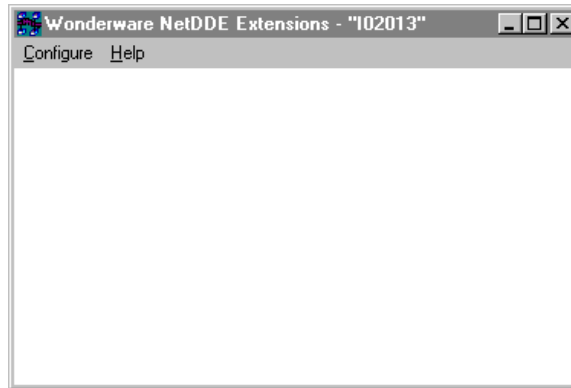
Installation Requirements for WinSock

Prior to installing Windows NT operating system and enabling its TCP/IP interface, a TCP/IP stack that conforms to the WinSock 1.1 standard must be installed.

To establish conversations between nodes, a network name and address must be defined for the local node and for each remote node.

Accessing the WinSock Interface Configurations

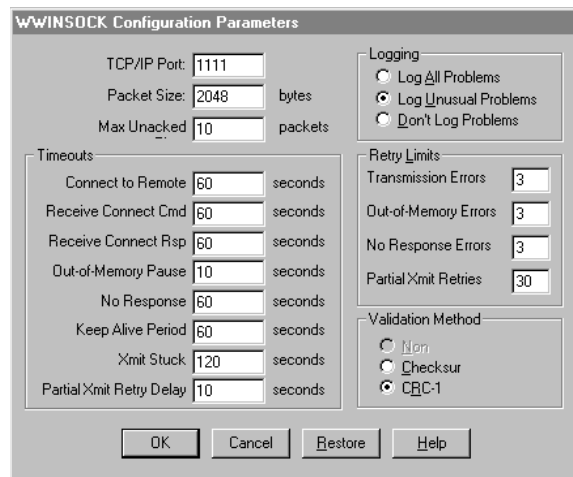
To access the WinSock network interface configurations, execute **NetDDE Extensions**. The **Wonderware NetDDE Extensions** dialog box appears.



Configuring WWINSOCK

To configure WWINSOCK

- 1 On the **Configure** menu, double-click **WWINSOCK**. The **WWINSOCK Configuration Parameters** dialog box appears.



- 2 In the **TCP/IP Port** box, type the local port number used by TCP/IP to connect to a host. All hosts communicating with each other must use the same port number.
- 3 In the **Packet Size** box, type the size of network packets to be used over the WinSock network. The default value of 2048 bytes is optimal for most configurations.

- 4 In the **Max Unacked Pkts** box, type the number of unacknowledged packets the WinSock network interface will allow.

During normal operation, the WinSock interface lets several unacknowledged network packets remain outstanding at the interface before pausing to wait for acknowledgment. If the values for this parameter are different at two connecting nodes, the minimum value is used by the WinSock interface.

- 5 In the **Timeouts** group, type the time-out values (measured in seconds) for the WinSock interface:

- **Connect to Remote**

Type the number of seconds for the WinSock network interface to wait before timing out on the connection.

- **Receive Connect Cmd**

Type the number of seconds for the WinSock network interface to wait from the time of the initial connect to the time it receives an initial connect packet from the remote node.

- **Receive Connect Rsp**

Type the number of seconds for the WinSock network interface to wait from the time it sends an initial connect packet to the time it receives an initial connect response packet from the remote node.

- **Out-of-Memory Pause**

Type the number of seconds for the WinSock network interface to wait to re-transmit a message to the remote node after receiving notification the remote node is out of memory.

- **No Response**

Type the amount of time for the WinSock network interface to wait for a transmitted packet to be acknowledged by the remote node before attempting to re-transmit the unacknowledged packet.

- **Keep Alive Period**

Enter the amount of time between keep-alive packets that are exchanged between connected Windows NT programs. Keep-alive packets are used as positive acknowledgment the connection is still functional in the absence of normal DDE message activity.

- **Xmit Stuck**

Enter the number of seconds for the WinSock network interface to wait for permission from the network interface to transmit an outbound packet before timing out and closing the connection.
 - **Partial Xmit Retry Delay**

Enter the number of seconds for the WinSock network interface to wait before re-transmitting an unsuccessfully transmitted packet.
- 6 Select the **Logging** type that you want to use. These options provide the ability to control the amount of information the specific network interface logs to the WWINSOCK.LOG file located in the WINNT/SYSTEM32 directory. The following mutually exclusive options are available:
- **Log All Problems**

Log all problems detected at the network interface.
 - **Log Unusual Problems**

Only log problems that are unusual for the network interface. This is the default option.
 - **Don't Log Problems**

Disable problem logging.
- 7 In the **Retry Limits** group, type the retry limits enforced by the WinSock network interface after the associated timeout expires:
- **Connect to Remote**

Type the number of times the WinSock network interface will retry transmission of a specific packet to a remote node after that packet has been rejected by the remote node.
 - **Out-of-Memory Errors**

Type the number of times the WinSock network interface will retry transmission of a specific packet to a remote node after that node has requested that WinSock "back off" (due to low memory conditions).
 - **No Response Errors**

Type the number of times the WinSock network interface will retry transmission of a specific packet to a remote node without receiving any response from the remote node for that packet.

- **Partial Xmit Retries**
Type the number of times the WinSock network interface will try to re-transmit an unsuccessfully transmitted packet before closing the connection.
- 8 Select the **Validation Method** type that will provide the ability to control data authentication performed on message packets.
- **None**
Is not available in the WinSock interface configuration extension.
 - **Checksum**
This method uses a checksum (summing of message contents) to verify data integrity and is the default.
 - **CRC-16**
This method uses a 16-bit cyclic redundancy check to verify data integrity.
- 9 To restore the originally installed default values for all parameters, click **Restore**. Otherwise, click **OK** or **Cancel**.

WinSock Error Messages

Error messages for the WinSock network interface are logged to the WWSOCK.LOG file located in the WINNT/SYSTEM32 directory. To view error messages from this file, open the file in a text editor, such as Notepad. Possible error messages include:

Error Messages	Descriptions
AsyncWindowProc: WSOCK_EVENT error WSAERRORCODE on socket SOCKET_NUMBER	A WinSock message was received indicating an error has occurred for a specific asynchronous event.
Changing the TCP/IP Port will require you to change on this every node in your system! Are you sure you want to change this?	This warning message states that changing the port number on the local host requires that all hosts which will connect to the local host will need to have matching port numbers to establish a connection.
ConnectToHost: connect() failed, WSAERRORCODE	The connect call to the specified host failed with the error specified by WSAERRORCODE.

Error Messages	Descriptions
Copyright (c) 1993 Wonderware Software Development Corp. All rights reserved.	Informational copyright message.
Local host HOST_NAME is not in the host table. Please add HOST_NAME to host table.	The local host name was not found in the host table. HOST_NAME is the NetDDE node name and must be entered in the host table for the WinSock interface to initialize properly.
Maximum Sockets supported: <i>NNNNN</i>	Maximum number of sockets supported by the TCP/IP vendor's WinSock.
NDDEAddConnection: bind() failed, error: ERROR_CODE	Unable to bind a socket. The error code specifies the reason.
NDDEAddConnection: connect() failed, error: ERROR_CODE	Attempt to connect failed. The error code specifies the reason.
NDDEAddConnection: socket() failed, error: WSAERRORCODE	Unable to create a socket. The error code specifies the reason.
Local host HOST_NAME is not in the host table. Please add HOST_NAME to host table.	The local host name was not found in the host table. HOST_NAME is the NetDDE node name and must be entered in the host table for the WinSock interface to initialize properly.
Maximum Sockets supported: <i>NNNNN</i>	Maximum number of sockets supported by the TCP/IP vendor's WinSock.
NDDEAddConnection: bind() failed, error: ERROR_CODE	Unable to bind a socket. The error code specifies the reason.
NDDEAddConnection: connect() failed, error: ERROR_CODE	Attempt to connect failed. The error code specifies the reason.

Error Messages	Descriptions
NDDEAddConnection: socket() failed, error: WSAERRORCODE	Unable to create a socket. The error code specifies the reason.
NDDEAddConnection: Unknown host HOST_NAME. error: WSAERRORCODE	Host name and address were not in the host table. Enter the host name and Internet address into the host table.
NDDEShutDown: No listen was outstanding at shutdown.	No listen socket existed at shutdown. This is an internal anomaly which indicates the listen socket was destroyed before NetDDE shutdown.
NODE_NAME not in host table. Please configure host table properly.	Specified node name was not found in the host table. Enter the host name and Internet address into the host table.
ReceiveAllData: Receive Error = WSAERRORCODE, Socket = NNNNN, BufferSize = NNNNN	A receive error occurred while trying to read data. The most common occurrence of this message is for a WSAEWOULDBLOCK. In this case, there is either inadequate buffer space or no data pending to be read. If the buffer is less than the NetDDE buffer size, then the buffer space for WinSock should be increased.
SendData: Too many partial Tx retries on same packet: NNN/NNN.NNN	Too many attempts were made to transmit the same packet. The connection will be closed.
SetAsyncEvents: socket NN, hwnd NNNN	A bad socket identifier or Async window handle was identified while setting asynchronous socket attributes. Internal application error.
SetupListen: bind() failed. WSAERRORCODE	Unable to bind the listen socket. The creation of listen socket failed during binding. The WSAERRORCODE specifies the WinSock error.

Error Messages	Descriptions
SetupListen: listen() failed. ERROR_CODE	Unable to create the listen socket. The creation of listen socket failed during the initialization. The WSAERRORCODE specifies the WinSock error.
SetupListen: socket() failed. WSAERRORCODE	Unable to create the listen socket. The creation of listen socket failed during the establishment of the socket. The WSAERRORCODE specifies the WinSock error.
Unable to resolve address for host HOST_NAME. error: WSAERRORCODE	WinSock was unable to resolve the hostname. Verify the host name is in the host table or if a DNS is being used, the DNS is reachable, and the host name exists.
WinSock initialization error: ERROR_CODE	WinSock initialization error in WSStartup. WinSock internal error. WWinSock initialization will fail.
WSAAsyncGetHostByName failed: WSAERRORCODE	WinSock was unable to resolve the host name because the function that retrieves the host name reported an error. Verify the host name is in the host table. Or, if a DNS is being used, verify the DNS is reachable and the host name exists.
WWINSOCK vN.NN... Node NODE_NAME	Informational message providing WinSock version number and node name.

If the LogAll Option is Selected

Error Messages	Description
NDDTimeSlice: Closing Connection to host HOST_NAME on socket NNNN	Informational message stating that WinSock is closing a connection.
SendData: Connection closed while trying to send	WinSock received a close indication while trying to send data. The connection will be closed.
SendData: <i>NN</i> partial Tx retries on same packet: <i>NN/NN.NN NN</i>	Informational message stating that a packet has been partially transmitted <i>N</i> number of times.
WINSOCK_VENDOR_TEXT	WinSock vendor provided text. WinSock receives this text as part of its WinSock initialization procedure.

If the LogUnusual Option is Selected

Error Messages	Description
AcceptConnection: accept() failed, error: ERROR_CODE	An attempt to accept a connection from another host failed. The error code specifies the reason.
Changes take effect next time NetDDE is run	For the WinSock configuration parameters that were changed to take effect, NetDDE will have to be closed and reopened.
CreateAsyncWindow: CreateAsyncWindow failed	WinSock was unable to create its Async window. WinSock initialization will fail.
CreateAsyncWindow: Register failed.	WinSock was unable to register its window class. WinSock initialization will fail.
NODE_NAME: Verify Error, closing connection	A validation error on a packet occurred in the message header or message data. The connection will be closed.

Error Messages	Description
SendData: Retxmt required. WSAERRORCODE	An unusual error occurred which requires retransmission of a packet. The error code specifies the reason.
SendData: send() failed, error: WSAERRORCODE	A packet was unsuccessfully sent, with the error code specifying the reason. The packet will be resent.
SetAsyncEvents() Failed	WinSock was unable to properly initialize the new socket with asynchronous attributes.

Low-Level Interface Logging

The following ERROR_CODES are returned by the low-level WinSock interface in response to various commands.

Error Messages	Description
WSAEACCES	Permission denied.
WSAEADDRINUSE	The specified address is already in use. (See the SO_REUSEADDR socket option under setsockopt().)
WSAEADDRNOTAVAIL	The specified address is not available from the local machine.
WSAEAFNOSUPPORT	The specified address family is not supported by this protocol.
WSAEBADF	Bad file number.
WSAECONNABORTED	The virtual circuit was aborted due to timeout or other failure.
WSAECONNREFUSED	The attempt to connect was forcefully rejected.
WSAECONNRESET	The virtual circuit was reset by the remote side.
WSAEDESTADDRREQ	A destination address is required.

Error Messages	Description
WSAEFAULT	The <i>addr</i> len argument is too small (less than the size of a struct sockaddr).
WSAEHOSTDOWN	The host is down.
WSAEHOSTUNREACH	Unable to connect to specified host.
WSAEINPROGRESS	A blocking Windows Sockets call is in progress.
WSAEINTR	The (blocking) call was canceled via WSACancelBlockingCall().
WSAEINVAL	listen() was not invoked before an accept().
WSAEISCONN	The socket is already connected.
WSAELOOP	An illegal loopback operation.
WSAEMFILE	The queue is empty upon entry to accept() and there are no descriptors available.
WSAEMSGSIZE	The datagram was too large to fit into the specified buffer and was truncated.
WSAENAMETOOLONG	The specified name is too long.
WSAENETDOWN	The Windows Sockets implementation has detected the network subsystem has failed.
WSAENETRESET	The connection must be reset because the Windows Sockets implementation dropped it.
WSAENETUNREACH	The network can't be reached from this host at this time.
WSAENOBUFS	No buffer space is available.

Error Messages	Description
WSAENOPROTOOPT	The option is unknown or unsupported. In particular, SO_BROADCAST is not supported on sockets of type SOCK_STREAM, while SO_ACCEPTCONN, SO_DONTLINGER, SO_KEEPAIVE, SO_LINGER and SO_OOINLINE are not supported on sockets of type SOCK_DGRAM.
WSAENOTCONN	The socket is not connected (SOCK_STREAM only).
WSAENOTSOCK	The descriptor is not a socket.
WSAEOPNOTSUPP	The referenced socket is not a type that supports connection-oriented service.
WSAEPFNOSUPPORT	Protocol format not available.
WSAEPROTONOSUPPORT	Protocol not supported.
WSAEPROTOTYPE	The specified protocol is the wrong type for this socket.
WSAESHUTDOWN	The socket has been shutdown; it is not possible to sendto() on a socket after shutdown() has been invoked with how set to 1 or 2.
WSAESOCKTNOSUPPORT	Socket type not supported.
WSAETIMEDOUT	The attempt to connect timed out without establishing a connection.
WSAETOOMANYREFS	Too many references.
WSAEWOULDBLOCK	The socket is marked as non-blocking and no connections are present to be accepted.

NetDDE Helper Service

The NetDDE Helper service `WVNETDDE.EXE`, maintains connectivity between NetDDE conversations by performing two main functions:

- Ensuring that shares remain available as different users log on and off.
- Hooking the DDE agent so that client-side DDE conversations do not end when a user logs off.

The NetDDE Helper service shares the NetDDE shares so that remote computers can access them. These shares are established under the authentication of the "master" ArchestrA user account that is specified during common component installation. Details for the ArchestrA user account are encrypted and stored in the Windows NT Registry.

The Wonderware NetDDE Helper service is set up to use the System account and be interactive. This account does not have permissions to establish network shares. Therefore, when the service starts up, it uses the ArchestrA user account to establish the shares.

Services as NetDDE Clients

Any service that can function as a NetDDE client must be configured to:

- Start using the System user account and interact with the desktop. To configure this, use the Services program in Control Panel.
- Impersonate a user using the ArchestrA user account before starting the DDE conversation.

Index

C

copyright 2

D

DDE Share Security extension 19

DDE shares

- configuring security 22

- creating custom 26

- overview 21

- permissions 22

- security 21, 24

DDE, See Dynamic Data Exchange

documentation conventions 5

Dynamic Data Exchange

- overview 17

F

FastDDE

- overview 18

I

I/O Servers

- quality reporting 13

N

NetDDE

- Extensions 19

- for Windows NT 19

- overview 18

NetDDE for Windows NT

- interface configuration 20

NetDDE Helper service

- overview 40

P

permissions

- for DDE shares 22

Q

quality 7

- bit fields 8

- from I/O Servers 13

S

services

- NetDDE Helper 40

system requirements

- NetDDE Extensions 20

T

TCP/IP 7, 20
time-stamping 7, 8
topic 21

W

WinSock 7
 configuring 29
 configuring interface 28
 error messages 32
 installation requirements 28
WinSock network interface extension 19