Wonderware® FactorySuite®

Protocols Guide

Revision A

Last Revision: 11/19/02

Invensys Systems, Inc.

Invensys Systems, Inc.
33 Commercial Street
Foxboro, MA 02035
(949) 727-3200
http://www.wonderware.com

Trademarks

Contents

Before You Begin

About This Book

This guide provides background information on the main protocols used between components of Wonderware products. A protocol is the set of rules and standards for enabling computers to connect and exchange data over the network. This guide also includes information on setting up and using some of these protocols.

C H A P T E R 1

SuiteLink

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

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

Contents

SuiteLink Features

SuiteLink was 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 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.

Note You do not have to create shares for SuiteLink I/O Servers.

Hardware and Software Requirements

In order use SuiteLink for data communications for a computer, you must have the following installed on that computer:

- TCP/IP installed and configured.
- Windows NT operating system; Version 4.x or higher.

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, 1601 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:

QQSSSSLL

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	Determines the status of the data value. Data values can be bad (0), uncertain (1), or good (3). A server that supports no 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.

Bit Field	Description
Substatus	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. 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.
Limit	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.

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.)

Value	Quality	Description
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 cannot 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 is 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 cannot move.

Wonderware I/O Server Quality Reporting

Wonderware I/O Servers can report six mutually exclusive states of quality of data being sent back to their clients. They are as follows:

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

The conditions under which each of these quality states will be reported are described in the following sections.

Good

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

- 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 will be reported if it was necessary to clamp the intended value to a limit because the value was larger that 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 will be reported if it was necessary to clamp the intended value to a limit because the value was smaller that 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 will be 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 will be 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 will report 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 will be 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 cannot allocate memory).
- Lack of resources in the communications link.
- The communications link is off-line.
- 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.

C H A P T E R 2

DDE, FastDDE, and NetDDE

This chapter describes the DDE protocols used by Wonderware products.

Contents

DDE

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*.

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 would be a program (such as Excel) running a report-generating macro. The macro would open a channel to another application, request specific data, close the channel and use 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 will 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. Components of the FactorySuite can use DDE to communicate with I/O device drivers and other DDE application programs.

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

FastDDE

FastDDE provides a means of packing many 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 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.

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

Microsoft Windows NT Operating System and NetDDE

Microsoft's version of NetDDE is included in the Microsoft Windows NT operating system product. 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.

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 will depend on what other PCs and workstations you intend to connect to 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 allows you to configure a DDE Share Security policy and to 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 18.

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

For more information, see "Configuring the WinSock Interface" on page 26.

System Requirements

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

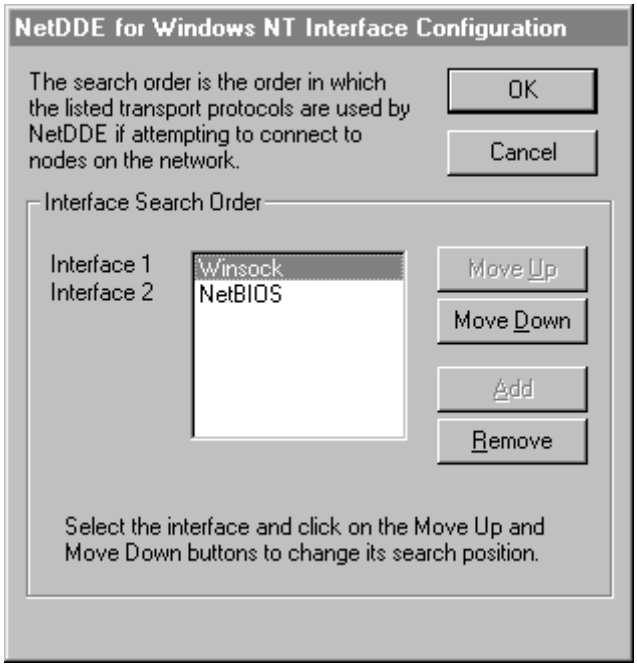
- Microsoft Windows NT operating system 3.51 or later. If running InTouch you must have Microsoft Windows NT 4.0 operating system installed.
- TCP/IP Network Protocol installed.

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

Since 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 allows you to configure a DDE Share Security policy and to 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 then 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 will be 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 20.

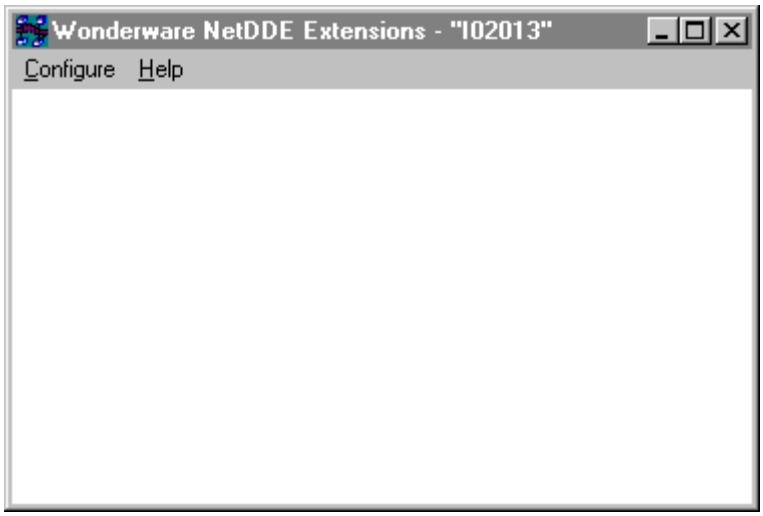
Default DDE Share Security

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 will use 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.

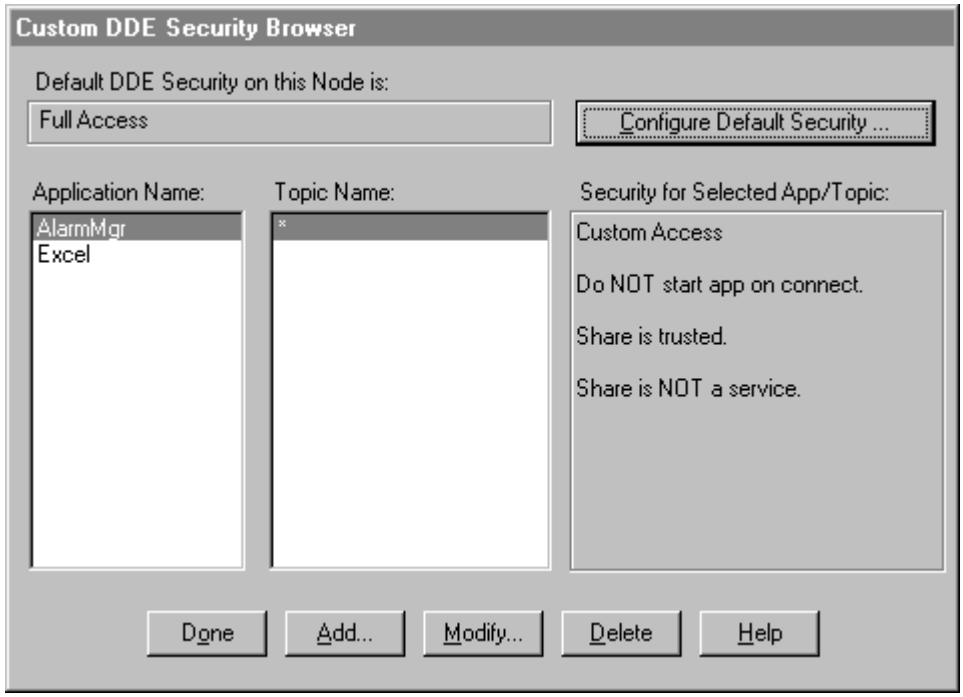
Security Configurations

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**.

To modify the default security level:

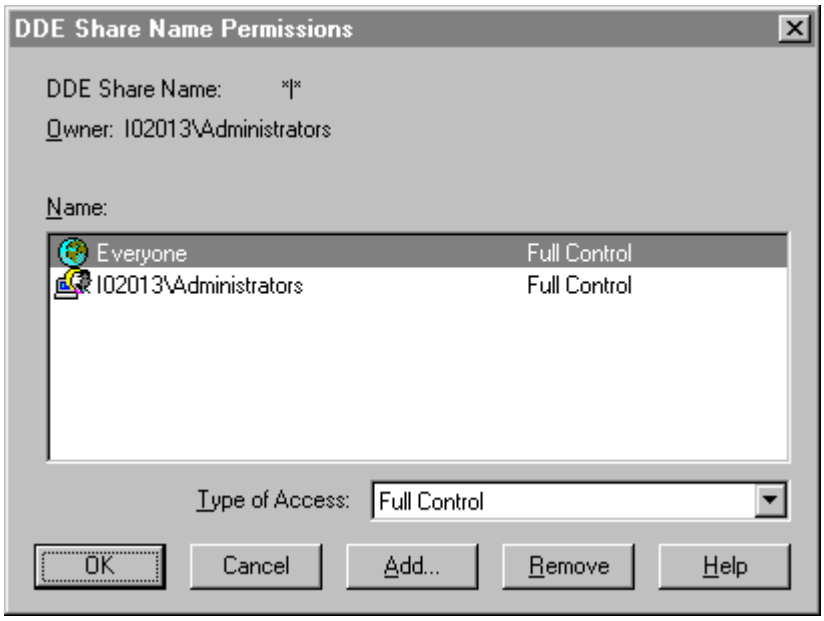
1. On the **Custom DDE Security Browser** screen, click the **Configure Default Security** button. The **Default DDE Security** dialog box appears.

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

   By default, 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, allowing you to become gradually accustomed to securing DDE data in this manner.

   For more information, see "DDE Share Permission Levels" on page 19.

3. If the **Permissions** Default Access has been selected the **Permissions** button will appear. 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, refer to the *Microsoft Windows NT System Guide*.
6. Select the following **Default 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
   Allows other users to 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 **Cancel** to close the dialog box, without saving changes.
8. Click **OK** to accept changes and return to the **Custom DDE Security Browser** dialog box.

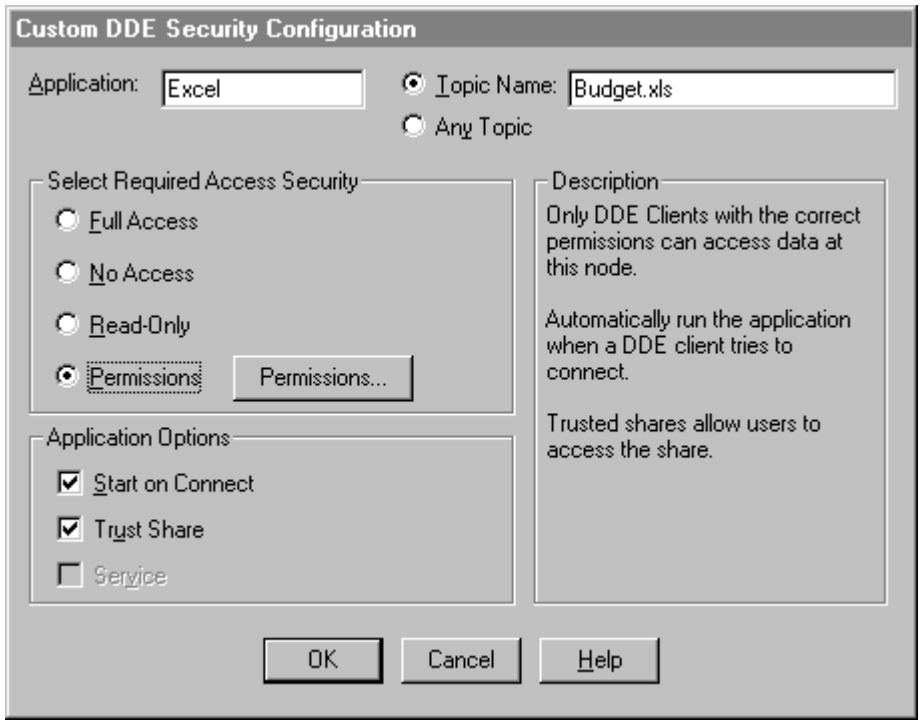
Customized DDE Shares

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** allows you to 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 that now has a password required 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 desired security permission level to be assigned to the custom DDE share. Descriptions for each option will appear in the field to the right of the option when selected.

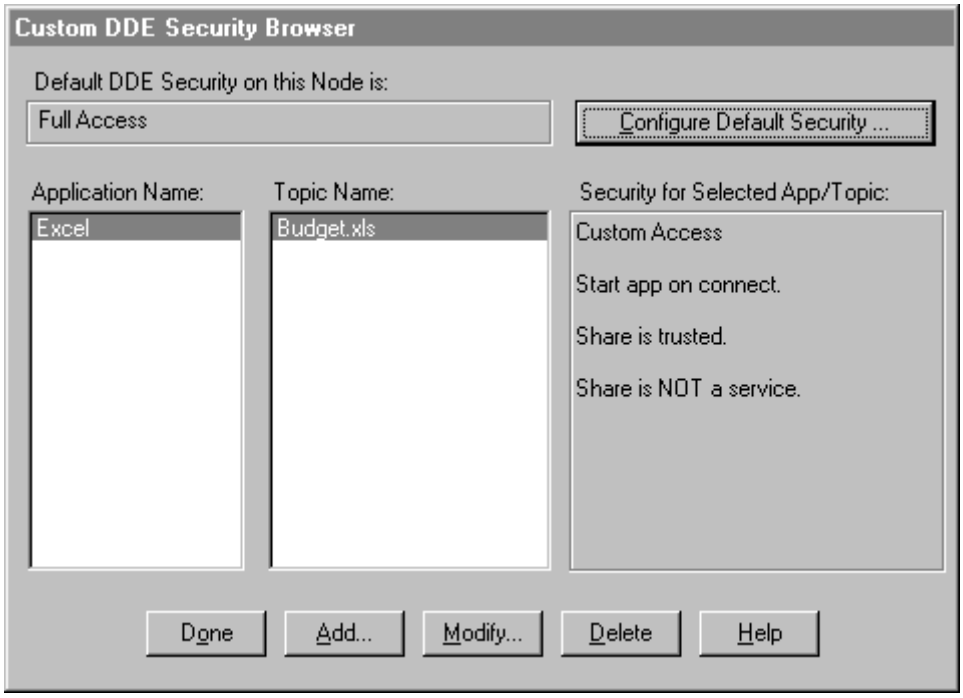
   For more information, see "DDE Share Permission Levels" on page 19.
7. In the **Application Options** group, select the from the following options:

   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
   Allows other users to 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 will appear with the added Application Name and Topic Name.



10. To view each nodes 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 will appear 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 will be removed from the DDE shares database and the displays in the **Custom DDE Security Browser** dialog will be 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.

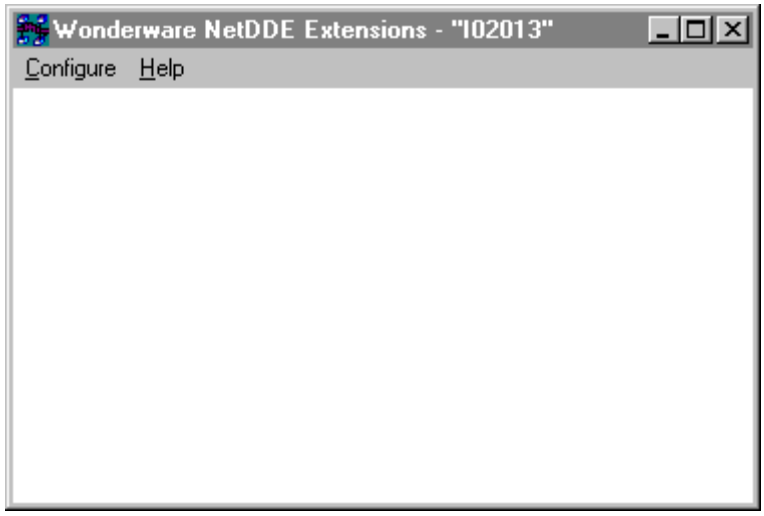
Installation Requirements for WinSock

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(s).

Accessing the WinSock Interface Configurations

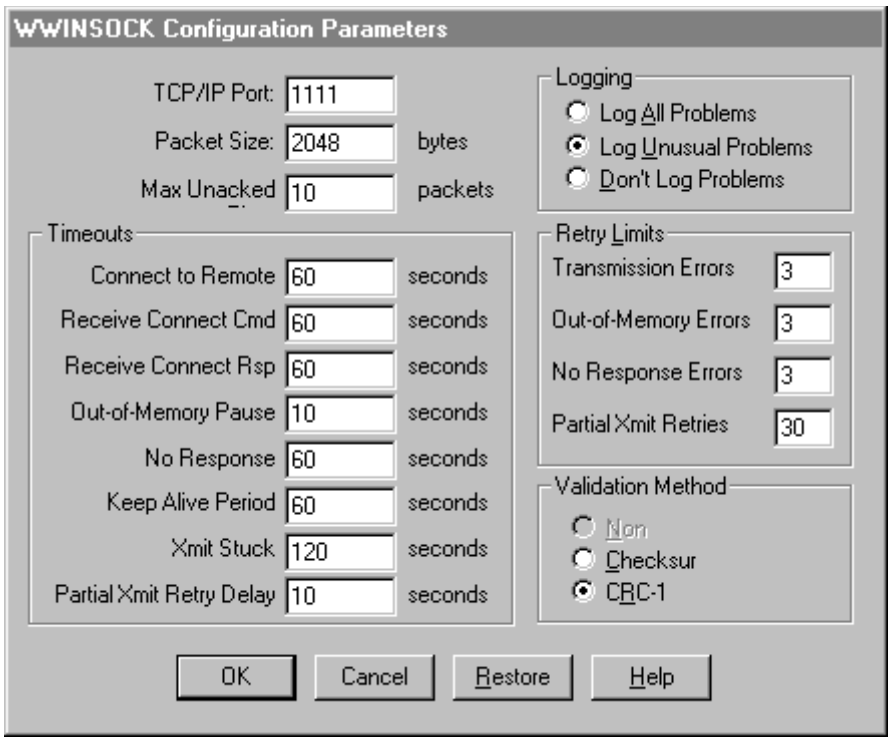
To access the WinSock network interface configurations, execute **NetDDE Extensions**. The **Wonderware NetDDE Extensions** dialog box appears.



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 when attempting to connect to a host. All hosts communicating with each other must have 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 allows several unacknowledged network packets to be 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 will be used by the WinSock interface.

5. In the **Timeouts** group, type the timeout values (measured in seconds) for the WinSock interface:

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

Receive Connect Cmd
Type the number of seconds the WinSock network interface will 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 the WinSock network interface will 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 the WinSock network interface will 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 the WinSock network interface will 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 the WinSock network interface will 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 the WinSock network interface will 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 will log 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 WWINSOCK.LOG file located in the WINNT/SYSTEM32 directory. To view error messages from this file, open the file in a text editor, e.g., Notepad. Possible error messages include:

"AsyncWindowProc: WINSOCK_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.

"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: *NNNNNN*"
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.

"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.

"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 WSAStartup. 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 v*N.NN*... Node NODE_NAME"
Informational message providing WinSock version number and node name.

If the 'LogAll' Option is Selected:

"NDDETimeSlice: Closing Connection to host HOST_NAME on socket *NNNNN*"
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: *NN* partial Tx retries on same packet: *NN/NN.NN NN*"
Informational message stating that a packet has been partially transmitted *N* 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:

"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.

"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.

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.

WSAEFAULT
The *addrlen* 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.

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_KEEPALIVE, SO_LINGER and SO_OOBINLINE 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 (WWNETDDE.EXE) is designed to maintain connectivity between NetDDE conversations by performing two main functions:

- Ensures that the shares remain available as different users log on and off.
- Hooks the DDE agent so that client-side DDE conversations are not terminated 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" FactorySuite user account that is specified during common component installation. Details for the FactorySuite 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 FactorySuite user account to establish the shares.

Services as NetDDE Clients

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

- Start up 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 FactorySuite user account before starting the DDE conversation.

Index