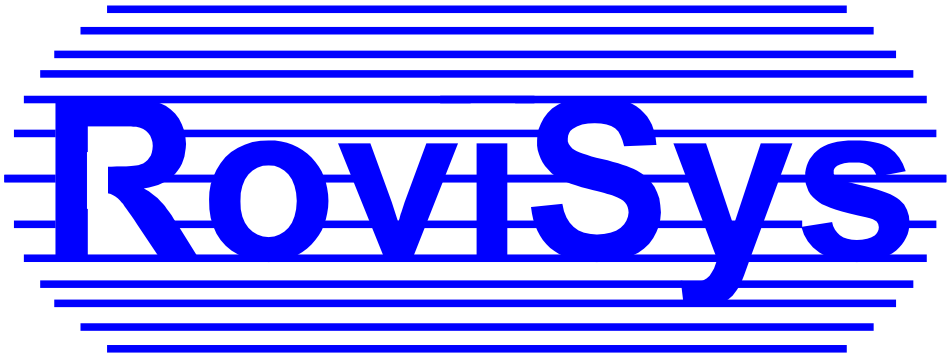


Excellence in Process Control

PIDDE Server

for

OSI PI Historian

Table of Contents

1. Introduction

The RoviSys Company PIDDE server enables client DDE applications access to the OSI (formerly known as Oil Systems Incorporated) PI Historian. The PIDDE server can communicate with multiple PI 2 and PI 3 home nodes providing access to both the snapshot and archive values. System integrity, functionality, and data throughput is maintained by utilizing the built in PI and DDE exception reporting techniques. PIDDE server can be run on an NT based PI-API node PC or on a PI3 Home node PC.

PIDDE Server supports Wonderware InTouch, Rockwell Software RSView32, Intellution FIX 32 (via their DDE client driver) third party Man Machine Interfaces and other MMI packages. It also supports desktop DDE client applications such as Microsoft Excel, Access and Word.

A version intended for InTouch is provided that was developed using the Wonderware DDE development toolkit. It supports the Wonderware “fast” DDE data exchange mechanism and Suitelink. Another version intended for RSView32 was developed using the Rockwell Software DDE development toolkit. When used with RSView32 it supports the Rockwell Software “advanced” DDE data exchange mechanism and Wonderware “fast” DDE when used with InTouch. Both versions are installed by setup. Either works with any DDE client.

This document is intended for individuals who are familiar with OSI PI Historian configuration principals in terms of how to add points and enable access permissions. An understanding of how to use the client DDE application is also required.

2. Terms

Throughout this document the RoviSys designed OSI PI Historian DDE Server will be referenced as PIDDE server or server. The term *PI* is defined to include both PI 2 and PI 3 systems.

DDE is an acronym for Dynamic Data Exchange. DDE is a communication protocol designed by Microsoft to allow applications in the Windows environment to send/receive data and instructions to/from each other. This protocol is implemented as a client-server relationship between two or more concurrently running applications. In this relationship, the server is the provider of data and accepts requests from all other applications called clients which are interested in its data. Some applications like InTouch, RSView and Microsoft Excel are designed to operate simultaneously as a client and server. The PIDDE server is designed to always operate as a server.

3. Functional Description

To exchange data with the PIDDE server the client opens a channel to it by specifying two things. The first is the server's application name, which for the PIDDE server is always *PIDDE*. The second is the topic name. A topic name is one that the user defines using the PIDDE server program. A topic name must be defined for each PI home node from which data is to be exchanged.

Once a channel is opened using the application name and topic, data exchange can begin for any PI tag name within that topic (PI home node). PIDDE server items are PI tag names appended with subscripts which specify a particular PI attribute. An example item might be SINUSOID;SNPVR which specifies the latest snapshot value of the PI tag as a real number or SINUSOID;DESC which specifies the descriptor for the PI tag.

Application name, topic, and item are defined as part of third party man-machine interface (MMI) tag names when DDE point types are selected. The number of pieces of information (tags) that can be exchanged with the PIDDE server is the lesser of the maximum number of allowable MMI tag names or maximum number of tags supported by the PI Historian. For example a 128 tag InTouch system interfaced to a 10,000 tag PI Historian home node limits the data exchange to 128 DDE tags. Whereas an unlimited tag version of InTouch interfaced to this same device has a tag limit that exceeds the 10,000 PI Historian home node tag limit. This is due to the fact that each tag within the PI Historian has multiple pieces of information made available within InTouch as individual tag name / items.

4. Installation

Attach the PIDDE protection key to the LPT1 port. This key is intended to be connected in series with any other third party MMI protection key. (Demo versions of PIDDE are not provided with this key.) PIDDE Server software is comprised of two disks.

To install PIDDE insert disk one into the floppy drive and run the “SETUP.EXE” program by double clicking on its icon from within file manager or explorer. PIDDE Server setup uses the industry standard install shield.

Two versions of PIDDE Server are automatically installed. Both are designed to be used with Windows 95/98/NT 4.0 or later (Intel based PCs). The first version is developed with the Wonderware DDE toolkit which supports Wonderware “fast DDE”. It is installed in the “PIDDE\WW” directory. This version should be used with Wonderware InTouch versions 6.x and higher or other non MMI 32 bit DDE clients. The other version is developed with the Rockwell Software DDE toolkit which supports Rockwell “Advanced DDE” and Wonderware “fast DDE”. It is installed in the “PIDDE\RS” directory. This version should be used with Rockwell Software RSView32 or other MMI or non-MMI DDE clients.

Setup will install a few sample files such as an Excel spreadsheet. The Excel spreadsheet file BROWSE.XLS is used by RoviSys to validate the browse data items supported by the PIDDE server. This file provide a good test example to show item syntax, validate communication with the PI Historian and provide general purpose tag browsing capabilities from within Excel. This sample Excel version 7.0 file requires definition of a PIDDE server topic named “BROWSE”.

In some cases the PIDDE Server version based on the Wonderware toolkit might need some of the DLLs installed in the "Program File\PIDDE Server\FactorySuite\Common" subdirectory. Wonderware installations already have these DLLs installed. For non-Wonderware installations if you receive a missing DLL error when running PIDDE Server, the missing DLL(s) can be copied from the afore mentioned directory to the Windows System32 directory to resolve the missing DLL condition.

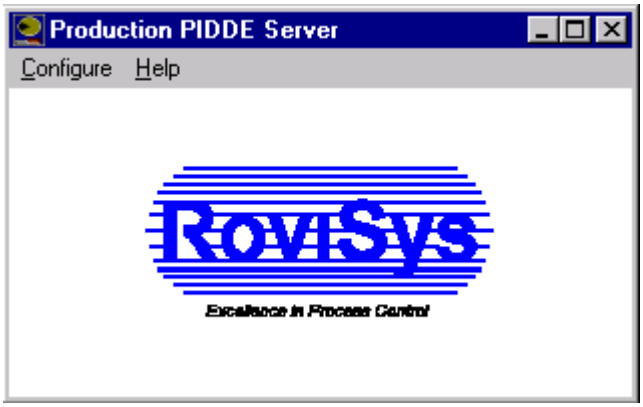
Installation of the server is complete. If this is not a demo copy you should make a backup copy of the server distribution disks. Also non-demo users need to update their OSI licensing agreement to include a PI API license. Contact the local OSI representative about pricing. Make sure that you register the server using the enclosed registration form. Registration entitles you to three months of free updates, server technical support and upgrade notifications. We truly welcome your comments and suggestions on how we can further improve the functionality of our server to better fit your needs.

5. Getting Started

This section is intended to "bridge the gap" between the configuration information provided in this manual and the third party MMI documentation. At this point, the PI home node should already be running. The MMI software and PIDDE server has been installed on the target machine, and you are ready to begin defining PI related DDE points that the MMI is to acquire. The following sub-sections demonstrate running the PIDDE server, defining a topic and defining server operational parameters. Thereafter, the MMI document should be consulted for other details of tag name generation and usage of tags within graphical displays.

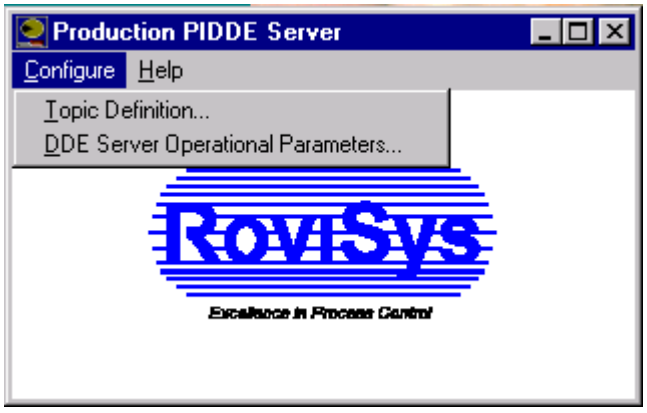
5.1. Executing the PIDDE Server Program

To execute the PIDDE Server program double click the appropriate  icon, in the RoviSys PIDDE Server group (or other group you may have added it to). The following screen will appear:



If you are reviewing a demo copy of the server, the program menu bar will read "Demo PIDDE Server (minutes left: 60)". *Please, DO NOT begin to rely on the demo copy to monitor production data within your facility. Its automatic shutdown after 1 hour of operation will freeze the data on your MMI which could result in operations making incorrect control decisions based on old data.*

This screen is used to configure the PIDDE server. To select configure activities, type ALT C or single click on the Configure menu item which will display the following screen:



Type ALT T or single click on Topic Definition to configure server topic names and associated data.

Type ALT D or single click on DDE Server Operational Parameters to configure overall server performance associated with DDE data exchange.

5.2. Defining Topics

The Topic Definition sub-menu selection allows definition of PIDDE server topics. A topic is a name which represents a specific PI home from which data is to be exchanged. For example, a topic named *NT_HOME* could describe data exchange with a PI 3 home node running on a NT PC whereas *VAX_HOME* might describe a topic data exchanged with a PI 2 home node running on a VAX VMS computer. Topic names can be selected that make it clear where the data is being exchanged. Multiple topics to the same PI home node are also permitted to support sets of data being exchanged at a rate configured by the topic. The Topic Definition sub-menu selection displays the following:

This area shows names of existing topics. Currently this example shows the definition of three topics called Browse, NT_HOME and VAX_HOME. The definition of a particular topic can be modified by double clicking on it or the modify button.

New Button:
Click on this button to define a new topic.

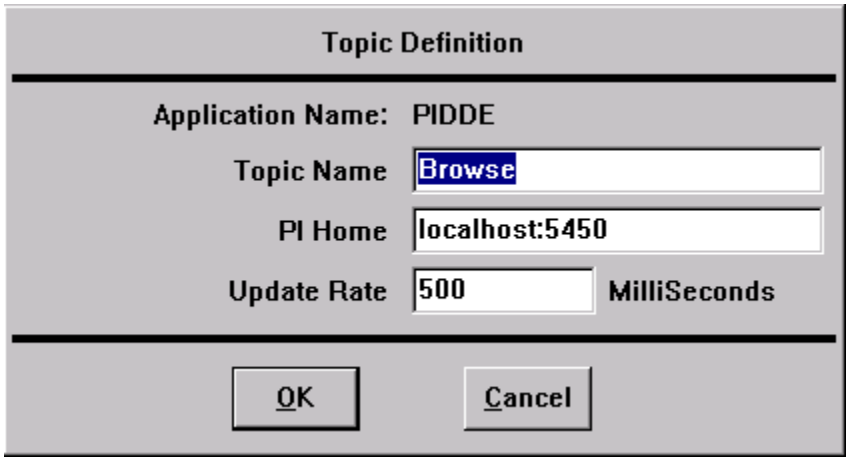
Modify Button:
Click on this button to bring up the topic definition dialog for the selected existing topic.

Delete Button:
Click on this button to delete the currently selected topic.

Cancel Button:
Click on this button to cancel topic definition.

OK Button:
Click on this button to accept current topic name selection and modify its topic definition.

When adding a new topic or modifying its existing definition, the following dialog is displayed:



Application Name:
is always fixed at *PIDDE .*

Topic Name:
is a topic name such as *Browse* defined by this example.

PI Home:
is the name of the PI home node to which this topic is associated. If PIDDE server is running on a different machine than the PI system, use PI home set equal to *PI3NAME*:5450 or *PI2NAME*:545, where *PI3NAME* is the name of the host machine running the PI 3 server and *PI2NAME* is the name of the host machine running the PI 2 server. If PIDDE server is running on the same machine as the PI 3 system for which data is to be exchanged, set host name equal to *localhost*:5450. Note that you ***must*** identify the type of PI home by appending ***:545*** to the PI Home name if it is a PI 2 system and ***:5450*** if it is a PI 3 system.

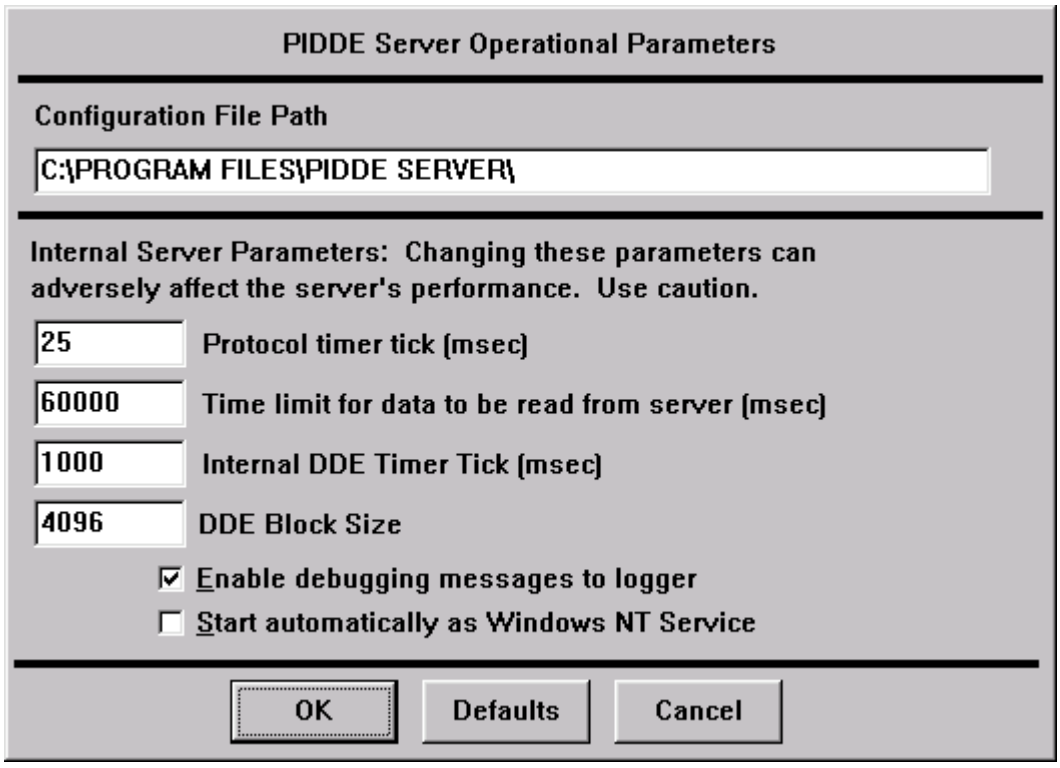
Update Interval:
frequency in milliseconds at which data is to be read from the defined PI Home node. Settings in the range of 1000 to 3000 milliseconds are common.

OK Button:
Click on this button to accept current topic definition.

Cancel Button:
Click on this button to cancel topic definition.

5.3. Defining Operational Parameters

The DDE Server Operational Parameters sub-menu selection allows definition of PIDDE server operational parameters. However, the defaults settings for these parameters provide good performance and in most cases, modifications are not required. Nevertheless, they can be changed to fine-tune the server for a specific environment and therefore, upon its selection it displays the following dialog:



Configuration File Path:
is used to specify the path (disk driver and directory) where the server will save its configuration file. The server will use this path to load its configuration file the next time it is started. Note that this entry is the disk and path name only. The server will always save its configuration in a file called *PIDDE.CFG*.

Protocol Timer Tick:
is the frequency (in milliseconds) at which the server checks for work to do and should be evenly divisible into the update intervals defined for each topic.

Time limit for data to be read from device:
is a spare parameter that currently has no functionality within the PIDDE server.

Internal DDE Timer Tick:
is the frequency (in milliseconds) in which the server performs various internal timing functions associated with error reporting.

DDE Block Size:
is the size in which data is packed when communicating to the MMI. This field is used to adjust this block size but has no effect on the operation of the server with other non-MMI DDE clients.

Enable Debugging Messages to Logger :
is used to enable user selectable debugging messages to the WW logger.

Start Automatically As a NT Service :
is used to enable PIDDE to start automatically as a NT service. This feature is only supported for Windows NT / 2000 based systems. Selecting it when running on Windows 95/98/ME has no effect.

Default Button:
Click on this button to set the server operational parameters to default settings.

Cancel Button:
Click on this button to cancel changes to the server operational parameters.

OK Button:
Click on this button to accept changes to server operational parameters.

6. PI Tag Data Items

PIDDE server PI tag data items provide access to detailed information about PI tags. Data concerning the setup of the tag is available along with its current snapshot value which can be read and written. Tag archived values are also available. PI tag data items utilize the PI tag name with a subscript appended to it which selects a particular attribute to be read. The convention for using the PIDDE Server PI tag data items is:

PI_TAGNAME;attribute

where:

Item field	Purpose
PI_TAGNAME	actual PI tag name for which data is to be exchanged
attribute	attribute for the indicated tag name that is to be exchanged

The "attribute" field of the item selects the specific data to be accessed. The following table lists the supported attributes.

Attribute	Purpose	DDE Type	Access
;tagname	name of the current tag	message	read
;pointNum	point number assigned to the tag	integer	read
;type	point type (0 = real, 1 = integer, 2 = digital)	integer	read
;typeText	point type as a text string	message	read
;descriptor	descriptor	message	read
;exDesc	extended descriptor	message	read
;zero	zero of the point	real	read
;span	span of the point	real	read
;engUnits	engineering units	message	read
;excDev	exception report deviation in engineering units	real	read
;excMax	exception report maximum time in seconds	integer	read
;excMin	exception report minimum time in seconds	integer	read
;compDev	compression deviation in engineering units	real	read
;compMax	compression maximum time in seconds	integer	read
;compMin	compression minimum time in seconds	integer	read
;scan	scan flag (0 = off, 1 = on)	boolean	read
;archiving	archiving flag (0 = off, 1 = on)	boolean	read
;compressing	compressing flag (0 = off, 1 = on)	boolean	read
;filter	filter setting	message	read
;source	point source	message	read
;loc1	location code 1	integer	read
;loc2	location code 2	integer	read
;loc3	location code 3	integer	read
;loc4	location code 4	integer	read
;loc5	location code 5	integer	read
;typical	typical point value	real	read
;resCode	point resolution code	integer	read
;sqRoot	point square root code	integer	read
;totalizer	totalizer code (0-6)	integer	read
;convFactor	conversion factor	real	read
;createName	name of login account used to create the point	message	read
;createDate	date when the point was created	message	read

;modifyName	name of login account used to last modify the point	message	read
;modifyDate	date when the point was last modified	message	read
;snpQ	quality of snapshot read (0 = good, 1 = bad)	boolean	read
;snpVR	snapshot value as a real	real	read/write
;snpVI	snapshot value as an integer	integer	read/write
;snpVText	snapshot value as text	message	read/write
;snpWriteStat	snapshot write status	integer	read/write
;snpReadStat	snapshot read status	integer	read
;snpTime	snapshot time	integer	read
;snpTimeText	snapshot time in text format	message	read
;arcRead	read the archive request flag (0 = done, 1 = request it)	boolean	read/write
;arcReadText	result of last archive read request	message	read
;arcReadMode	mode of the archive read (1 = before, 2 = after, 3 = interpolated, 4 = interpolated or before) for given time stamp	integer	read/write
;arcQ	quality of value from the archive (0 = good, 1 = bad)	boolean	read
;arcVR	archive value at requested time as a real data type	real	read
;arcVI	archive value at requested time as an integer data type	integer	read
arcVText	archive value at requested time as message data type	message	read
;arcReadStat	status from the archive at requested time	integer	read
;arcTime	requested archive time (absolute	integer	read/write
;arcTimeText	requested archive time in text format	message	read/write

The majority of the PI tag item attributes are self explanatory. There are two groups that are intended for accessing the snapshot and archive. The others provide tag setup information and generally are not required by DDE client applications but only provided for the sake of completeness.

6.1. PI Tag Snapshot Values

PI tag snapshot values are accessed using the ;snp* attributes. Snapshot values that may be accessed are its quality *;snpQ*, value as a real *;snpVr*, value as an integer *;snpVi* value as a text string *;snpVText*, its PI read status *;snpReadStat*, PI time of snapshot *;snpTime*, and PI time of snapshot as a text string *;snpTimeText*. The snapshot values may also be written. The default write status is zero but may be defined using the *;snpWriteStat* attribute.

Note that the *;snpQ* attribute is a discrete value managed by PIDDE Server. Its value is zero when the snapshot value can be read successfully and one when it is bad due to a read failure.

When accessing discrete PI tags the *;snpVr*, *;snpVi*, or *;snpVText* attributes can be used. Discrete values should be written as 0, 1, 2, etc. PIDDE Server will automatically translate them to the equivalent internal PI digital state code. When discrete values are access via the *;snpVText* attribute, there value will be set to the corresponding digital state table string. They may be written as 0, 1, 2, etc or written with the actual digital state strings like “Off” or “On” etc. PIDDE Server will accept either format.

6.2. PI Tag Archive Values

PI tag archive values are accessed using the ;arc* attributes. Archive values that may be accessed are its quality *;arcQ*, value as a real *;arcVr*, value as an integer *;arcVi* value as a text string *;arcVText*, its PI read status *;arcReadStat*, PI time of archive *;arcTime*, and PI time of archive as a text string *;arcTimeText*.

The archive value is read when the discrete *;arcRead* attribute is set. Prior to requesting an archive value to be read the desired read mode must be written to the *;arcReadMode* attribute and desired archive time to *;arcTime* or *;arcTimeText* attributes.

The available selections for *;arcReadMode* are:

1 = selects the archive value before the current timestamp,
2 = selects the archive value after the current timestamp,
3 = selects an interpolated value for the current timestamp,
4 = selects the archive value before the current time stamp if the PI tag's resolution code is not equal to four. Otherwise it selects an interpolated value for the current timestamp.

The archive time to be read selected by writing a PI timestamp value to *;arcTime* as an integer or *;arcTimeText* as a string. The string can be written as an absolute time stamp expressed as a time string like "12-Jun-98 16:34:45". Or "TODAY", "YESTERDAY", "SUNDAY", "MONDAY", etc. Relative time can also be written like +|- n d|h|m|s (for example -2d for two days ago). An asterisk can be used for current time or a combination time can be specified using a combination of absolute times and a relative time. See the PI Data Archive Manual for more information on the time string format. When either *;arcTime* or *;arcTimeText* is written the other attribute is updated with the corresponding time. In other words, if a timestamp is written to *;arcTime*, PIDDE server will update *;arcTimeText* with a time string equivalent to what was written to ;arcTime. Likewise if *;arcTimeText* is written with a time string, PIDDE server will update *;arcTime* with the PI timestamp equivalent to what was written to *;arcTimeText*.

Note that the *;arcQ* attribute is a discrete value managed by PIDDE Server. Its value is zero when the archive value is read successfully and one when it is bad due to a read failure or a read request has not been issued yet.

When accessing discrete archive values the *;arcVr*, *;arcVi*, or *;arcVText* attributes can be used. When discrete values are access via the *;arcVText* attribute, there value will be set to the corresponding digital state table string.

7. PI Browser Items

PIDDE server PI browser items allow browsing through the PI tag database. Browsing returns data related to the setup of a browsed tag along with the current snapshot value when the browse was launched. PI tag browsing items utilize the name “?BROWSE” with subscripts appended to it which selects a particular attribute to be read. The convention for using the PIDDE Server PI tag browse items is:

?BROWSE;attribute

The "attribute" field of the ?BROWSE item selects the specific data to be accessed. The following table lists the supported attributes.

Attribute	Purpose	DDE Type	Access
;host	PI home node host name to be browsed (defaults to PI Home defined by the topic)	message	read/write
;name	name (supports wildcards) of tag to be browsed	message	read
;get	flag that requests the tag name to be browsed	boolean	read/write
;next	flag that requests the next tag name to be browsed	boolean	read/write
;prev	flag that requests the previous tag name to be browsed	boolean	read/write
;count	count of tags found for current ;name definition	integer	read
;tagname	name of the current tag	message	read
;pointNum	point number assigned to the tag	integer	read
;type	point type (0 = real, 1 = integer, 2 = digital)	integer	read
;typeText	point type as a text string	message	read
;descriptor	descriptor	message	read
;exDesc	extended descriptor	message	read
;zero	zero of the point	real	read
;span	span of the point	real	read
;engUnits	engineering units	message	read
;excDev	exception report deviation in engineering units	real	read
;excMax	exception report maximum time in seconds	integer	read
;excMin	exception report minimum time in seconds	integer	read
;compDev	compression deviation in engineering units	real	read
;compMax	compression maximum time in seconds	integer	read
;compMin	compression minimum time in seconds	integer	read
;scan	scan flag (0 = off, 1 = on)	boolean	read
;archiving	archiving flag (0 = off, 1 = on)	boolean	read
;compressing	compressing flag (0 = off, 1 = on)	boolean	read
;filter	filter setting	message	read
;source	point source	message	read
;loc1	location code 1	integer	read
;loc2	location code 2	integer	read
;loc3	location code 3	integer	read
;loc4	location code 4	integer	read
;loc5	location code 5	integer	read
;typical	typical point value	real	read
;resCode	point resolution code	integer	read
;sqRoot	point square root code	integer	read
;totalizer	totalizer code (0-6)	integer	read
;convFactor	conversion factor	real	read
;createName	name of login account used to create the point	message	read
;createDate	date when the point was created	message	read
;modifyName	name of login account used to last modify the point	message	read

;modifyDate	date when the point was last modified	message	read
;snpReadText	result of last snapshot read request	message	read
;snpQ	quality of snapshot value (0 = good, 1 = bad)	boolean	read
;snpVR	snapshot value as a real	real	read/write
;snpVI	snapshot value as an integer	integer	read/write
;snpVText	snapshot value as text	message	read/write
;snpWriteStat	snapshot write status	integer	read/write
;snpReadStat	snapshot read status	integer	read
;snpTime	snapshot time	integer	read
;snpTimeText	snapshot time in text format	message	read
;arcRead	read the archive request flag (0 = done, 1 = request it)	boolean	read/write
;arcReadText	result of last archive read request	message	read
;arcReadMode	mode of the archive read (1 = before, 2 = after, 3 = interpolated, 4 = interpolated or before) for given time stamp	integer	read/write
;arcQ	quality of value from the archive (0 = good, 1 = bad)	boolean	read
;arcVR	archive value at requested time as a real data type	real	read
;arcVI	archive value at requested time as an integer data type	integer	read
arcVText	archive value at requested time as message data type	message	read
;arcReadStat	status from the archive at requested time	integer	read
;arcTime	requested archive time	integer	read/write
;arcTimeText	requested archive time in text format	message	read/write

7.1. Browsing PI Database

The PI browser items allows general browsing of any PI home node tag database. The PI home node that will be browsed is indicated by the *;host* attribute. It is defaulted to the PI home node associated with the current topic but can be written to select any PI home node currently online. Note that when selecting a new PI home node don't forget to include the *:5450* or *:545* with the node name indicating its a PI3 or PI2 type node. Note also that PIDDE server will attempt to connect to the new PI home node when the *;get* item is set to one.

The *;name* PI browser item allows selection of a specific known PI tag name or wildcard selection of a group of PI tag names. The wildcard mask supports '?' as replacement of single characters and '*' to replace all characters to the end of a sub field. Sub fields are the first 2, the next 6, and the last 2 characters. If a sub field is blank, an asterisks from the previous sub field carries over. Sub fields do not apply to long tag names. The tags are found in alphabetical order. If a match if found to a long tag name and a tag name, the long tag name is returned.

The *;get* item is used to request PIDDE server to read the PI home node indicated by *;node* for tag names matching the wildcard mask indicated by *;name*. PIDDE server indicates completion of the get request by resetting the *;get* item. After the initial get request the next tag can be browsed by setting the *;next* item. PIDDE server indicates completion of the next request by resetting the *;next* item. After the initial get request the previous tag can be browsed by setting the *;prev* item. PIDDE server indicates completion of the previous request by resetting the *;prev* item.

7.2. Browser Snapshot Values

PI browse snapshot values are accessed using the ;snp* attributes. Snapshot values that may be accessed are its quality *;snpQ*, value as a real *;snpVr*, value as an integer *;snpVi* value as a text string *;snpVText*, its PI read status *;snpReadStat*, PI time of snapshot *;snpTime*, PI time of snapshot as a text string *;snpTimeText*, and result of last snapshot read request stored in *;snpReadText* as a message. The snapshot values may also be written. The default write status is zero but may be defined using the *;snpWriteStat* attribute.

The PI browse snapshot values are **not** continuously updated. They are updated only after each time the *;get*, *;next* or *;prev* attribute is set and PIDDE server indicates the requests has been serviced by resetting it and the result yielded a valid PI tag.

Note that the *;snpQ* attribute is a discrete value managed by PIDDE Server. Its value is zero when the snapshot value can be read successfully and one when it is bad due to a read failure or the get, next or previous tag request yielded and invalid tag.

When accessing discrete PI tags the *;snpVr*, *;snpVi*, or *;snpVText* attributes can be used. Discrete values should be written as 0, 1, 2, etc. PIDDE Server will automatically translate them to the equivalent internal PI digital state code. When discrete values are access via the *;snpVText* attribute, there value will be set to the corresponding digital state table string. They may be written as 0, 1, 2, etc or written with the actual digital state strings like “Off” or “On” etc. PIDDE Server will accept either format.

7.3. Browser Archive Values

PI tag archive values are accessed using the ;arc* attributes. Archive values that may be accessed are its quality *;arcQ*, value as a real *;arcVr*, value as an integer *;arcVi* value as a text string *;arcVText*, its PI read status *;arcReadStat*, PI time of archive *;arcTime*, PI time of archive as a text string *;arcTimeText* and result of last archive read request stored in *;arcReadText* as a message..

The archive value is read when the discrete *;arcRead* attribute is set. Prior to requesting an archive value to be read the desired read mode must be written to the *;arcReadMode* attribute and desired archive time to *;arcTime* or *;arcTimeText* attributes.

The available selections for *;arcReadMode* are:

1 = selects the archive value before the current timestamp,
2 = selects the archive value after the current timestamp,
3 = selects an interpolated value for the current timestamp,
4 = selects the archive value before the current time stamp if the PI tag's resolution code is not equal to four. Otherwise it selects an interpolated value for the current timestamp.

The archive time to be read selected by writing a PI timestamp value to *;arcTime* as an integer or *;arcTimeText* as a string. The string can be written as an absolute time stamp expressed as a time string like "12-Jun-98 16:34:45". Or "TODAY", "YESTERDAY", "SUNDAY", "MONDAY", etc. Relative time can also be written like +|- n d|h|m|s (for example -2d for two days ago). An asterisk can be used for current time or a combination time can be specified using a combination of absolute times and a relative time. See the PI Data Archive Manual for more information on the time string format. When either *;arcTime* or *;arcTimeText* is written the other attribute is updated with the corresponding time. In other words, if a timestamp is written to *;arcTime*, PIDDE server will update *;arcTimeText* with a time string equivalent to what was written to ;arcTime. Likewise if *;arcTimeText* is written with a time string, PIDDE server will update *;arcTime* with the PI timestamp equivalent to what was written to *;arcTimeText*.

Note that the *;arcQ* attribute is a discrete value managed by PIDDE Server. Its value is zero when the archive value can be read successfully and one when it is bad due to a read failure or the get, next or previous tag request yielded and invalid tag.

When accessing discrete archive values the *;arcVr*, *;arcVi*, or *;arcVText* attributes can be used. When discrete values are access via the *;arcVText* attribute, there value will be set to the corresponding digital state table string.

8. Trouble Shooting Hints

This section is provided to help the user identify and correct problems that may arise as a result of incorrectly setting up PIDDE server or tag name database. It is provided as a general guide to allow the user to decipher normal and abnormal operation. If this does not help the server can be enabled to post additional error messages to the WWLogger. This feature can be enabled using the "DDE Server Operational Parameters" dialog available under the "Configure" PIDDE Server menu selection. Use this feature to track down tough problems and discover hard to find configuration errors. Afterwards remember to remove it so as to not overload the WWLogger log file.

After defining topics with PIDDE server and DDE points within the MMI tag name database, the server will begin communicating with PI when the MMI is executed.

8.1. Could Not Initiate DDE Conversation or Remote Data Not Accessible

If while starting up the DDE MMI Client application, a dialog appears indicating that a particular DDE conversation could not be initiated or from Excel, remote data is not accessible, this signifies a topic has been referenced that has not been defined using PIDDE or the PI home node defined by the topic does not exist, is offline or just not responding.

1.) Verify the topic exists and it properly defines the PI home node name.

2.) Verify that the PI home node Proxy Database has been updated to allow access by the PC running PIDDE.

8.2. Appears To Be Communicating But No Data is Being Received

Make sure that the item tag names match tan names defined on the PI home node and they are on scan.

8.3. Not All PIDDE Server DDE Tags Are Receiving Data

1.) Verify point tags indicated by item name actually exists on the PI home node.

2.) Review the item spelling for accuracy. A common mistake is to mix the number zero and letter O. Also verify that the item subscript is valid for the given item.

3.) Make sure that all of the PC memory has not been used.

8.4. Cannot Write Data to the Snapshot or Archive

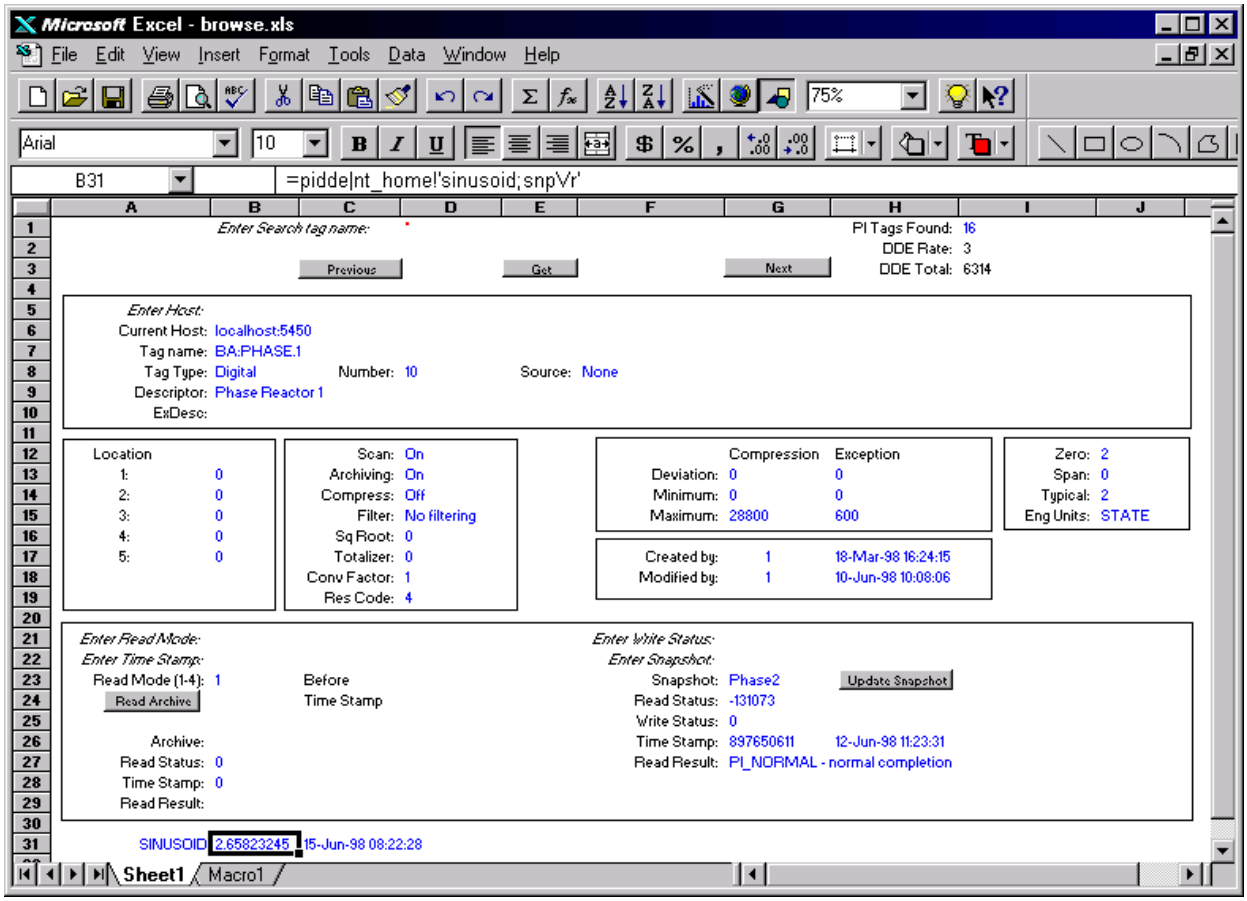
1.) Verify that the PI user associated with the problem topic has write privilege.

8.5. Data Values Exchanged Between PI and Server Are Different

1.) Verify that the tag point Min EU matches the PI zero and Min Raw is set equal to Min EU. Also verify that the tag point Max EU matches value set within PI and Max Raw is set equal to Max EU. Note that the EU Max is calculated as the PI zero plus span. Make sure you calculated the MMI settings correctly.

9. Reading DDE Values into Excel

PIDDE Server DDE values can be read into Excel by entering the appropriately formatted DDE remote reference formulas into the cells into which PIDDE data values are to be read. The following example shows the definition of one such formula:



In this example the cell at B31 references a PIDDE tag value from a topic named “nt_home” reading the current snapshot value of the “sinusoid” PI tag. This Excel 7.0 spreadsheet is included with PIDDE and is installed by setup under the SAMPLES subdirectory with a file name of “BROWSE.XLS”. This spread sheet is an example of using PIDDE to browse any PI home node database. It requires a topic named browse to be defined by PIDDE server. The browse spreadsheet is provided with PIDDE server free of charge with no expressed or implied warrantees.

The following table identifies the user interactions supported by the browse spreadsheet.

Cell	Purpose
D2	Enter tag search mask. Wildcard searches using the astericks ‘*’ character and placeholder using the question mark ‘?’ are supported. For example, a search tag of ‘S*’ will find all tags starting with the letter

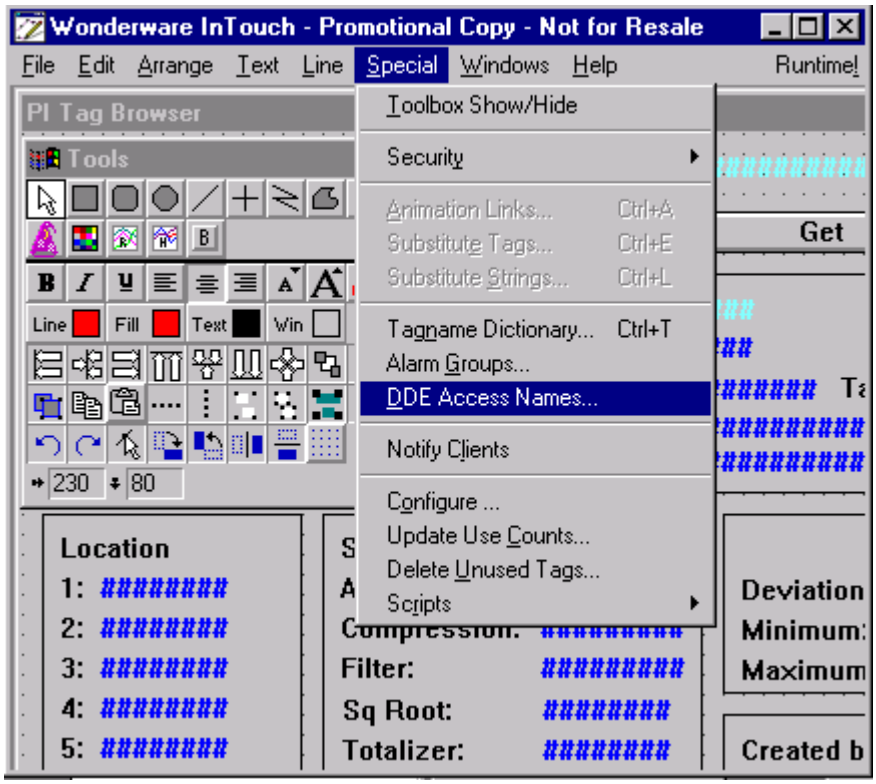
	'S'.
C3	Previous search push button from current search tag name. (Note that the get search push button must be pushed first.)
E3	Get search push button from the current search tag name.
G3	Next search push button from current search tag name. (Note that the get search push button must be pushed first.)
B5	Enter new PI home node host to search in different PI host than the current one defined for the topic named browse. The new PI home node host is selected after pushing the get push button.
A21	Enter the desire archive read mode. Choices are 1 for before the time stamp, 2 for after the time stamp, 3 for interpolated value for given time stamp, and 4 which is dependant on the "Resolution Code" for the given point. If the point resolution code is 4 the archive is read before the current time stamp. Otherwise, an interpolated value for the given time stamp is read.
B22	Time stamp to use when reading the archive value expressed in local PI time format.
C22	Absolute time stamp to use when reading the archive value expressed as a time string like "12-Jun-98 16:34:45". Or "TODAY", "YESTERDAY", "SUNDAY", "MONDAY", etc. Relative time can also be entered like +|- n d|h|m|s (for example -2d for two days ago). An asterisk can be used for current time or a combination time can be specified using a combination of absolute times and a relative time. See the PI Data Archive Manual for more information on the time string format.
A24	Read archive button used to request the archive to be read for the current or newly entered read mode and time stamp.
F21	Write status to use when writing a new snapshot value.
F22	New snapshot value to be written
H23	Update snapshot button used to request the value in F22 to be written to the PI snapshot using the write status in F21.

10. Using PIDDE Server With InTouch

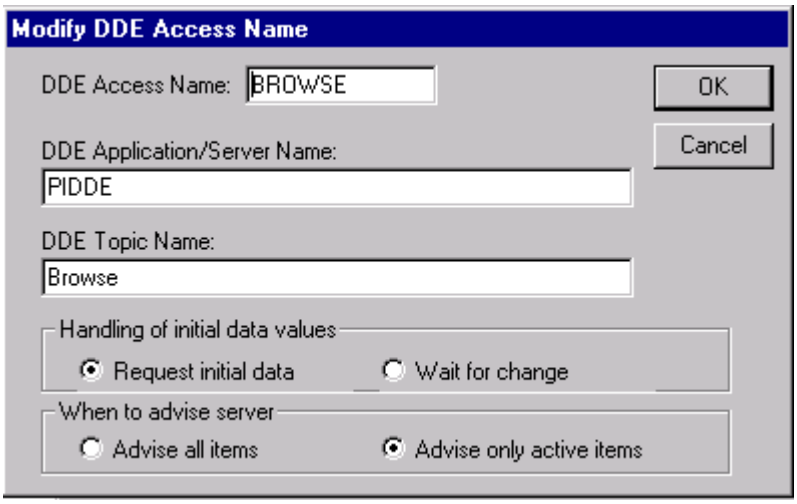
This section presents information related to using PIDDE Server with Wonderware Software Corporation's InTouch third party man-machine interface software. It assumes InTouch has already been installed on the target machine along with the version of PIDDE Server developed with the Wonderware DDE Server toolkit. Note that the screen snapshots presented in this section were taken from Wonderware Version 5.x. The configuration principles remain the same for the Factory Suite 2000 series software.

10.1. Definition of PIDDE Server DDE Points

InTouch WindowMaker is used to configure DDE points to be exchanged with the PIDDE server. Prior to configuring a DDE point, a DDE access name must be defined. A DDE access name associates a name to a DDE server and topic defined within that server. To define a DDE access name, select the DDE Access Names... sub-menu under the InTouch - WindowMaker Special menu. This selection is shown by the following screen.



After selecting DDE Access Names and new under the subsequent dialog, the following dialog will be displayed:



DDE Access Name:
Access name to be assigned to the DDE point which associates DDE server name and topic defined within the server.

DDE Application/Server Name:
Application name of the server which should always be set to PIDDE.

DDE Topic Name:
Topic name defined within the PIDDE server. In this example a topic name of *browse* had been assigned.

Request Initial Data Radio Button:
Selected **only** if the server program is other than a Wonderware DDE server such as PIDDE **and** the server program **does not** return data values immediately when a window is displayed.

Wait for Change Radio Button:
Selected when the DDE application is a Wonderware DDE server such as PIDDE.

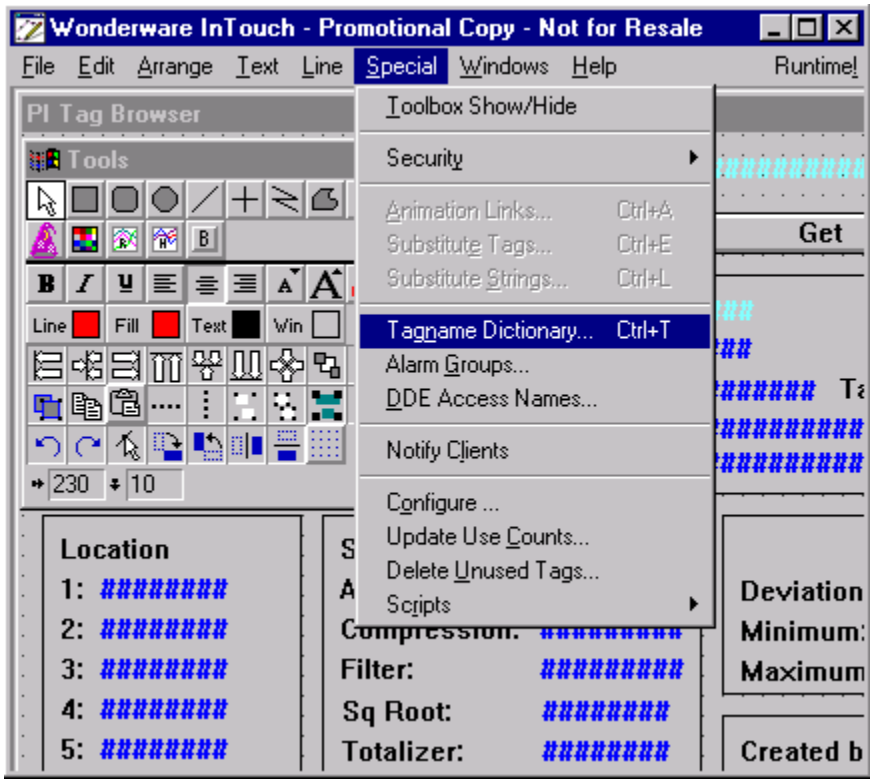
Advise All Items Radio Button:
Selected if the server program is to poll for all data whether or not it is in visible windows, alarmed, logged, trended or used in a script. If data throughput becomes sluggish for large PIDDE server applications and this option is selected, de-selecting it will improve throughput performance.

Advise Only Active Items Radio Button:
Selected to cause the server program to return only points in visible windows and points that are alarmed, logged, trended or used in any script.

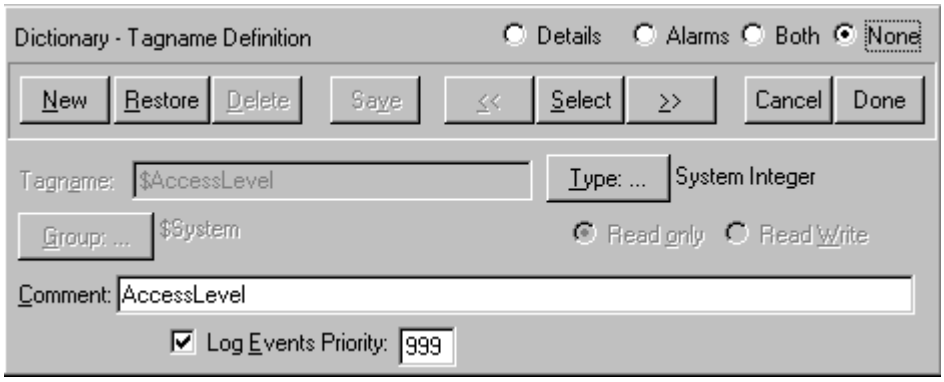
Cancel Button:
Click on this button to cancel changes to the definition of the DDE access name.

OK Button:
Click on this button to accept changes to the definition of the DDE access name.

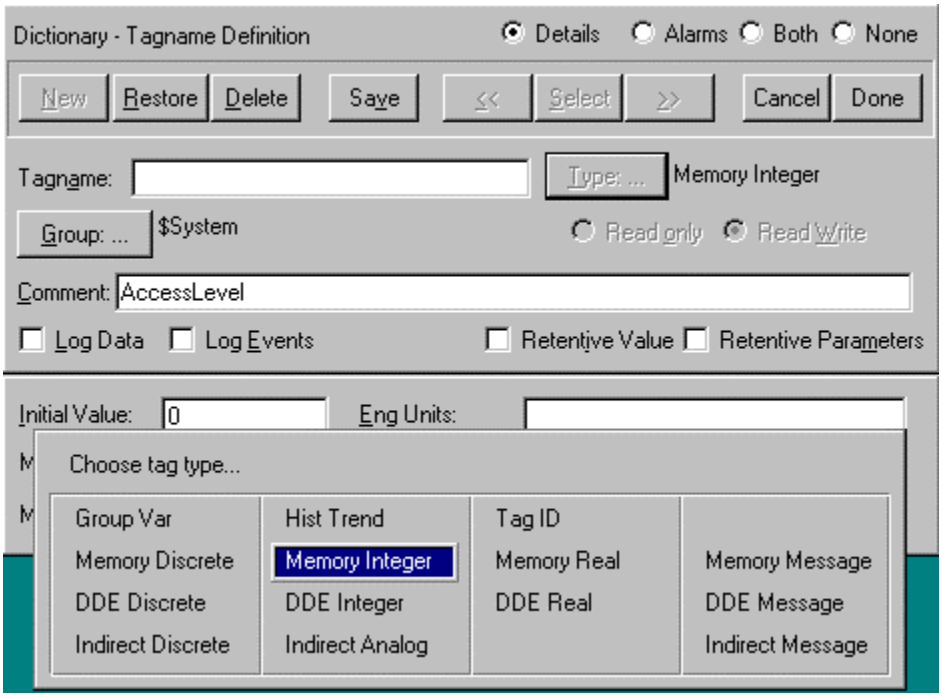
Once a DDE access name has been defined, tag names of DDE type can be defined. To define a tag name, the Tag Name Dictionary... sub-menu under the InTouch - WindowMaker Special menu. This selection is shown by the following screen.



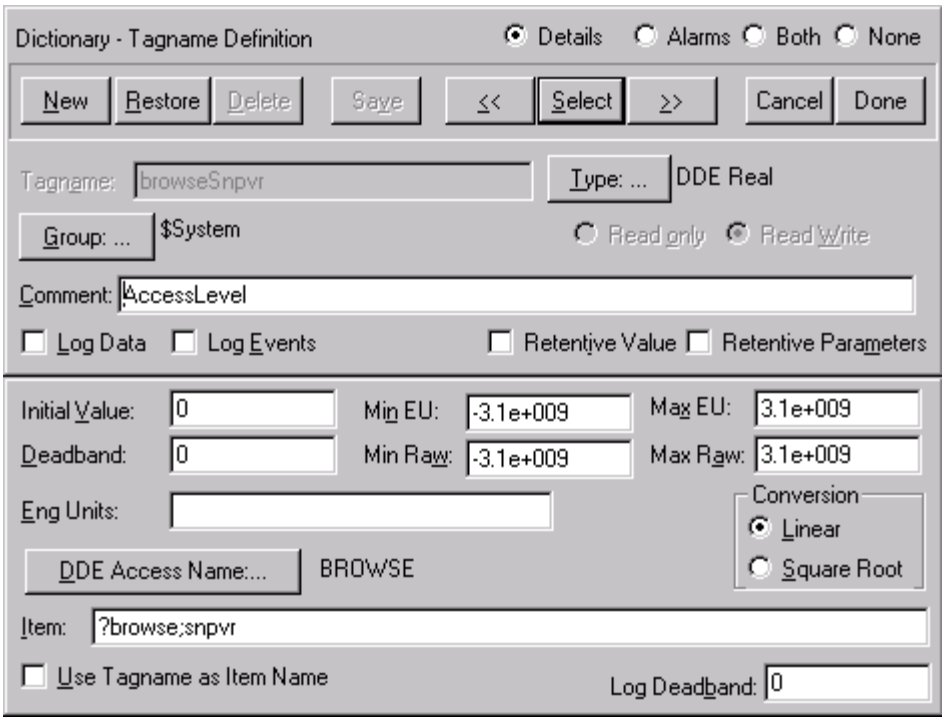
After selecting Tag Name Dictionary... the following dialog will be displayed:



First click on the *Details* radio selection second the New button and third the Type button. This will cause the following dialog to be displayed:



Tag types associated with the PIDDE server can be DDE Discrete, DDE Integer and DDE Real. As the names imply, the type you must choose is dependent on the data type of the point being exchanged with the server. The type corresponds to the server item name and subscript which must be completed as part of the tag name definition. For this example after clicking on DDE Real the following dialog will be displayed:



Note that the data within this dialog has been completed to give an example of reading a DDE real point. It will receive the real snapshot value of the browse item. This has been defined by the item named ?browse;snpVr. Note that the PIDDE server exchanges data in engineering units. Therefore, InTouch conversion of raw data to engineering units **must** be disabled by **always** setting the Min EU and Min Raw values equal along with Max EU and Max Raw values equal.

Initial Value:
Value stored in the tag name when the runtime database is first loaded.

Deadband:
Amount a tag name (expressed in engineering units not percent) must change before the database is updated.

Min EU:
Minimum engineering units value of the tag name when the minimum (or less) raw count value is received.

Max EU:
Maximum engineering units value of the tag name when the maximum (or greater) raw count value is received.

Min Raw:
Minimum value of the low clamp on the DDE value. For the PIDDE server the Min Raw value must always be set equal to the Min EU value.

Max Raw:
Maximum value of the high clamp on the DDE value. For the PIDDE server the Max Raw value must always be set equal to the Max EU value.

Eng Units:
Optional engineering units label to be used for the tag name.

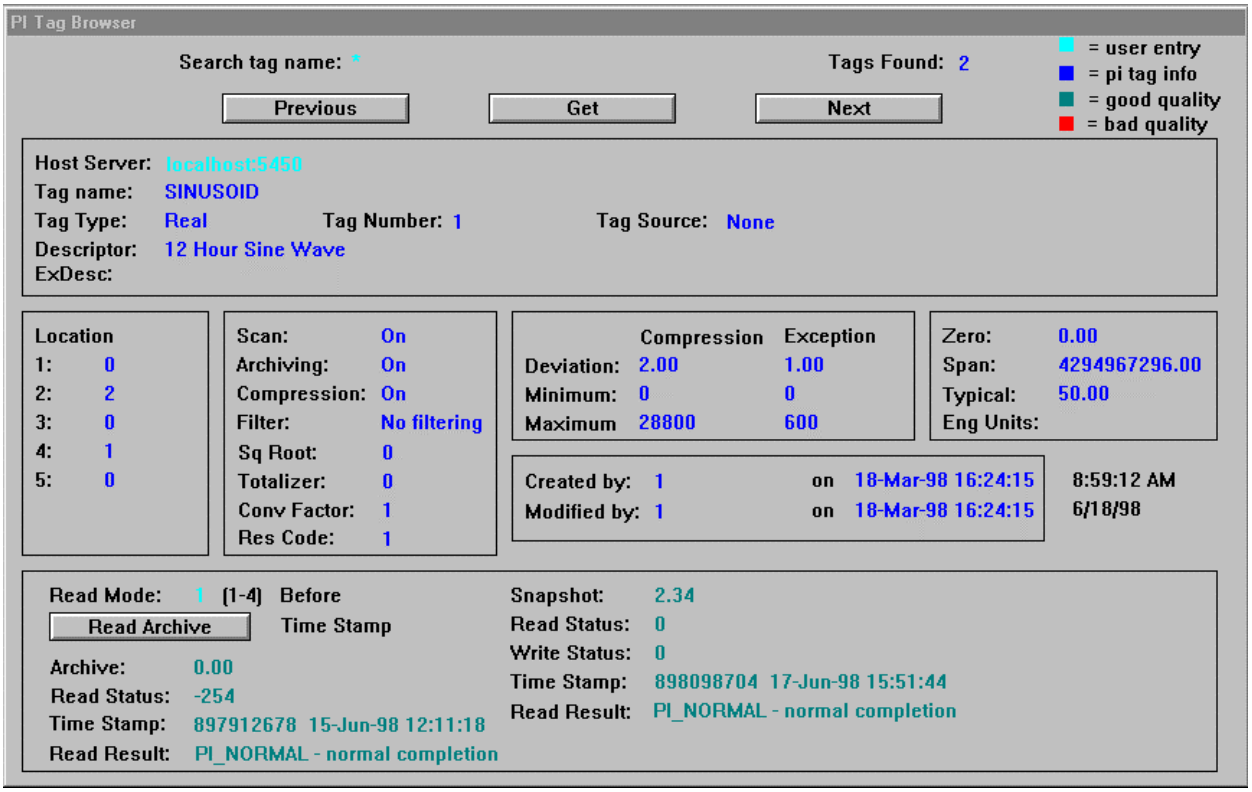
Conversion:
Not applicable to the PIDDE server since raw to engineering units conversion must always be disabled.

DDE Access Name Button:
Used to define or select the DDE Access Name to be associated with the tag name. The name that appears to the right of this button will be assigned to the tag.

Item:
Name of the PIDDE server point type and attribute.

10.2. Browse Application

The Browse InTouch application has been designed and developed by RoviSys. Its purpose is to support browsing and snapshot updating of any PI home node database. PIDDE server must be version 1.0 or later and configured with a topic named “Browse”. Browse is optionally loaded by setup. Browse is provided with PIDDE server free of charge with no expressed or implied warrantees. When InTouch View runs the Browse application, the following window is displayed:



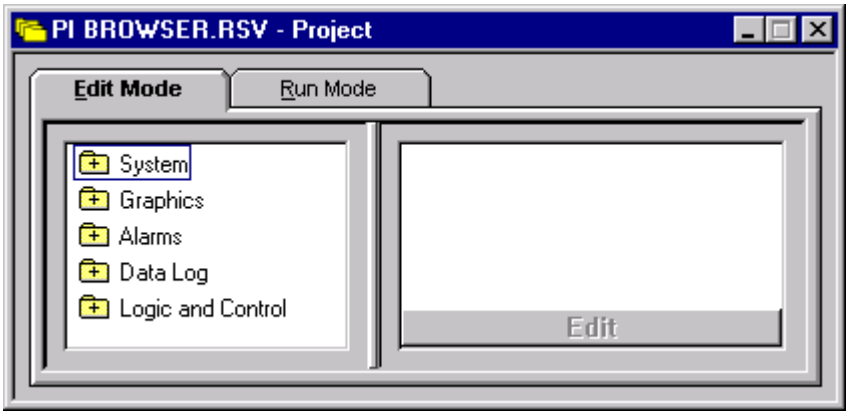
User inputs for this screen include the search tag name, host server, archive read mode, archive time stamp, snapshot value and write status. Buttons are Get (get set of tags for current search tag name), Next (next tag in list), Previous (previous tag in list), and Read Archive (read archive with current read mode and time stamp setting).

11. Using PIDDE Server With RSView32

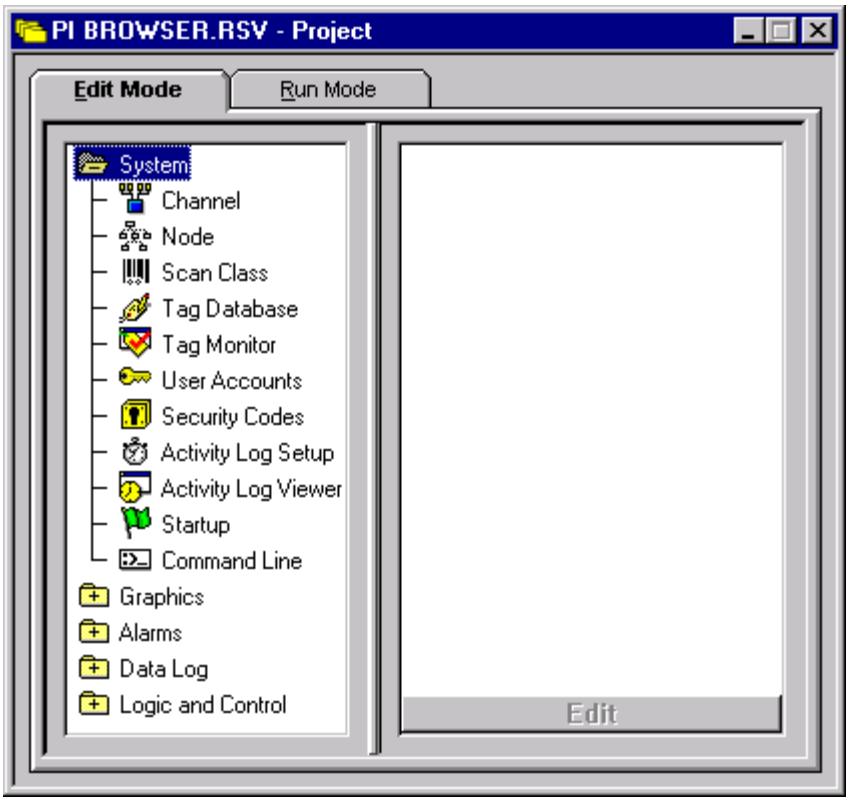
This section presents information related to using PIDDE Server with Rockwell Software's RSView32 third party man-machine interface software. It assumes RSView32 has already been installed on the target machine along with the version of PIDDE Server developed with the Rockwell Software DDE Server toolkit.

11.1. Definition of PIDDE Server DDE Points

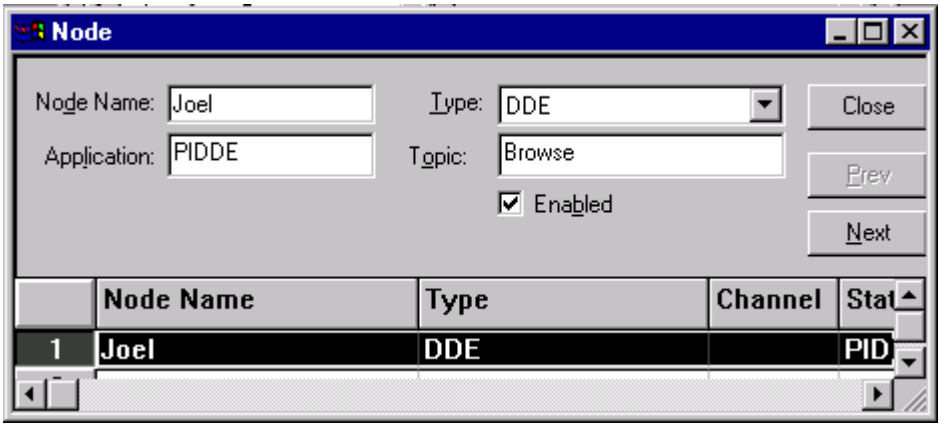
Defined a project using RSView32 Works. The following dialog is an example of a RSView32 PI Browser project:



Double click on the "System" folder to expand it shown as follows:

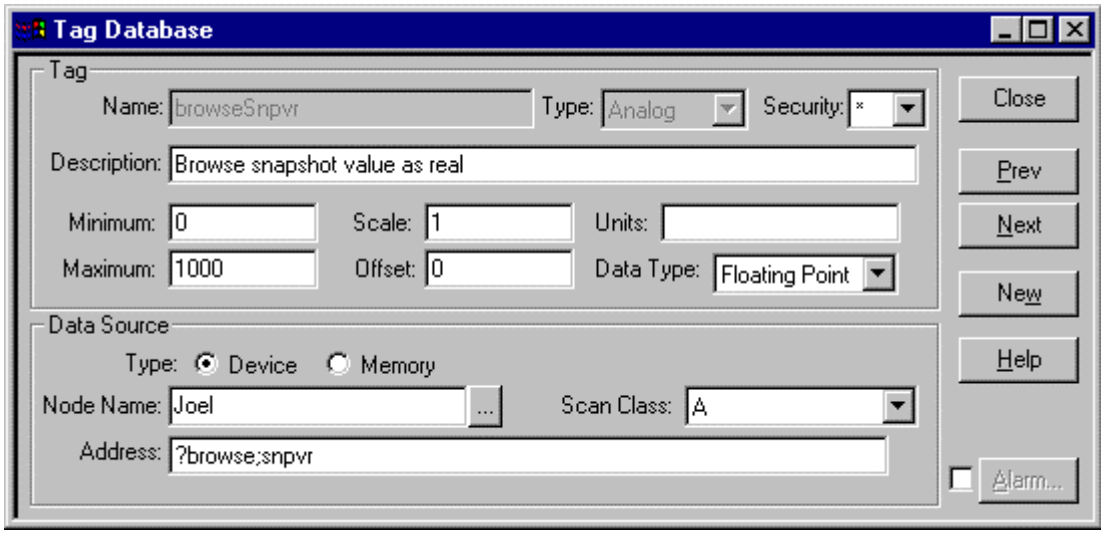


Double click on “Node” to configure a DDE node which will be used to create DDE tags under. This will display the following node definition dialog:



Fill in the node name where PIDDE is executed, select DDE type, set the application to PIDDE and assign a topic to match one defined within the PIDDE server. In this example a topic name of *browse* had been assigned.

Once the node has been defined double click on “Tag Database” to configure DDE points to be exchanged with the PIDDE server. This will display the following point definition dialog:



This dialog allows definition of any type of RSView tag. Assign the tag name and tag type. Note that the dialog fields change based on the tag type selected. For analog types the following fields are displayed:

Name:
Tag name assigned to this point.

Description:
Verbose description of the purpose for this point.

Type:
General type assigned too this tag. Valid choices are analog (real and integer numbers), digital (on or off values) and string.

Security:
Write security level assigned to this tag.

Minimum:
Minimum value allowed for this tag.

Maximum:
Maximum value allowed for this tag.

Scale:
Scaling factor to be applied against the value received for this tag.

Offset:
Offset value to be applied against the value received for this tag.

Units:
Engineering units description associated with this tag.

Data Type:
This selection is based on the DDE point item associated with this tag and the selected type field. Valid selections when the type field is “analog” can be real or long integer. Consult the specific DDE point item to determine which selection is applicable.

Data Source Type:
Declares point to be an internal RSView 32 point (memory) or coming from a driver, OPC Server or DDE Server (device).

Node Name:
Name of the PC running PIDDE server.

Address:
Name of the PIDDE server point type and attribute.

Scan Class:
RSView32 scan class for this point.

Close Button:
Closes this dialog after point definition has been completed.

Previous Button:
Go back on tag definition.

Next Button:
Advance to the next tag definition.

New Button:
Request definition of a new point.

Help Button:
Request help on tag definition dialog.

Alarm Checkbox:
Requests that alarming be enabled for this point.

Alarm Button:
If alarming has been enabled for this point click on this button to setup the alarm criteria.

Corporate Headquarters:
The RoviSys Company
1455 Danner Drive
Aurora, Ohio 44202
(330) 562-8600 (330) 562-8688 fax
http://www.rovisys.com