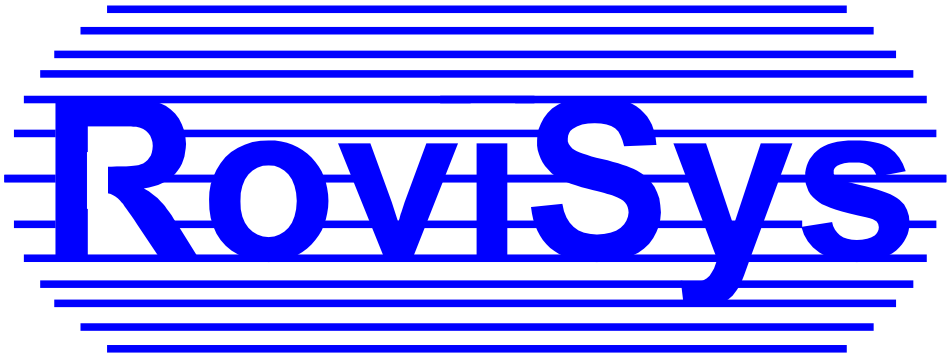


Excellence in Process Control

N90DDE Server

for Bailey

Command Series® / NETWORK 90® / INFI 90®

Table of Contents

1. Introduction

The RoviSys Company N90DDE server enables client DDE applications access to the Bailey Controls Command Series, NETWORK 90 and INFI 90 Distributed Control Systems. The N90DDE server can communicate at the network level via the appropriate Computer Interface Unit (NCIU0X), Plant Loop to Computer Interface (INPCI0X), Infi-net to Computer Interface (INICI01, INICI03, INICI12) or, within a single Process Control Unit, via a Computer Interface Command (CIC01) or Serial Port Module (NSPM01, IMSPM01). System integrity, functionality, and data throughput is maintained by utilizing standard exception reporting techniques.

Versions of N90DDE Server are available which support the Wonderware InTouch, Rockwell Software RSView and Intellution FIX 32 (via their DDE client driver) third party Man Machine Interfaces. All of these versions support desktop DDE client applications such as Microsoft Excel, Access and Word.

The InTouch version N90DDE was developed using the Wonderware DDE development toolkit. It supports the Wonderware Suitelink and “fast” DDE data exchange mechanism. The RSView version N90DDE was developed using the Rockwell Software DDE development toolkit. When used with RSView it supports the Rockwell Software “advanced” DDE data exchange mechanism and Wonderware “fast” DDE when used with InTouch. This version may also be used with Intellution FIX 32 running in the Windows 95/98/NT environment. For this configuration, the Intellution DDE client driver must also be present and the “normal” DDE data exchange mechanism is utilized.

This document is intended for individuals who are familiar with the Bailey configuration principals in terms of how to enable system level exception reports. An understanding of how to use the client DDE application is also required.

2. Terms

Throughout this document the RoviSys designed Command Series / NETWORK 90 / INFI 90 DDE server will be referenced as N90DDE server or server. The term *N90* is defined to include Command Series, NETWORK 90 and INFI 90.

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 N90DDE server is designed to always operate as a server.

3. Functional Description

To exchange data with the N90DDE server the client opens a channel to it by specifying two things. The first is the server's application name, which for the N90DDE server is always *N90DDE*. The second is the topic name. A topic name is one that the user defines using the N90DDE server program. Topic names are defined to represent a meaningful collection of data from the server. For example, *ICICOM1* would describe data exchange with the Bailey Infi-net to Computer Interface attached to the PC COM1 port.

Once a channel is opened using the application name and topic, data exchange can begin for all items defined within the topic. N90DDE server items are pre-defined names and subscripts associated with each type of N90 exception report. Included as part of the item name is the address within N90 at which the exception report function block resides. An example item might be STN.PV,1,2,3,50 which specifies the process variable of a N90 control station located at ring 1, node 2 module 3, block 50.

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 N90DDE server is the lesser of the maximum number of allowable MMI tag names or unique N90 addresses (defined within the MMI tag name items) allowed to be downloaded to the Bailey computer interface device as points. For example a 128 tag InTouch system interfaced to a Bailey INICI01 interface which supports 10,000 points limits the data exchange to 128 tags. Whereas an unlimited tag version of InTouch interfaced to this same device has a tag limit that exceeds 10,000. This is due to the fact that each point within the N90 interface has multiple pieces of information made available within InTouch as individual tag name / items. An example of this concept is the N90 control station. The data associated with a control station is a single point within the N90 interface device. Available data associated with this single point is process variable, set point, control output and mode. To access this data within the MMI, four tags and there corresponding *STN* items with unique (.PV, .SP, .CO, .MODE) subscripts would be defined, all specifying the same N90 address which translates to a single point within the N90 interface.

4. Installation

Attach the N90DDE 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 N90DDE are not provided with this key.) N90DDE Server software is comprised of three disks. To install N90DDE 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.

Two versions of N90DDE Server are installed. The first is developed with the Wonderware DDE toolkit which supports Wonderware Suitelink and "fast DDE". It is installed in the "N90DDE\WW" directory. This version should be used with Wonderware InTouch versions 6.x and higher or other non MMI 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 "N90DDE\RS" directory. This version should be used with Rockwell Software RSView32 or other DDE clients.

Setup will install a few sample files such as Bailey TXTEWS module configuration files and Excel spreadsheets. A Bailey TXTEWS configuration file called 10103.CFG corresponds with the Excel files DATATYPE.XLS and BLOCK.XLS stored in the samples directory. The Excel spreadsheet file DATATYPE.XLS is used by RoviSys to validate all data items and subscripts supported by the N90DDE server. The Excel spreadsheet file BLOCK.XLS is a subset of DATATYPE.XLS which only contains DDE links related to reading and tuning N90 configurations. These files provide a good test example to show item syntax and validate communication with the Bailey system. These sample Excel version 7.0 files reference a MFC or MFP residing at Ring 1, Node 1 and Module 3, a N90DDE server topic named "N90DDE" and assumes the Bailey interface device resides at ring 1, node 4. Install the configuration file 10103.CFG into the Bailey controller using TXTEWS to demonstrate live data updates into the DATATYPE.XLS spreadsheet or change the item addresses to match existing blocks within the Bailey system being interfaced.

In some cases the version based on the Wonderware toolkit might need some of the DLLs installed in the "Program File\N90DDE Server\FactorySuite\Common" subdirectory. Wonderware installations already have these DLLs installed. For non-Wonderware installations if you receive a missing DLL error when running N90DDE 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. 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.

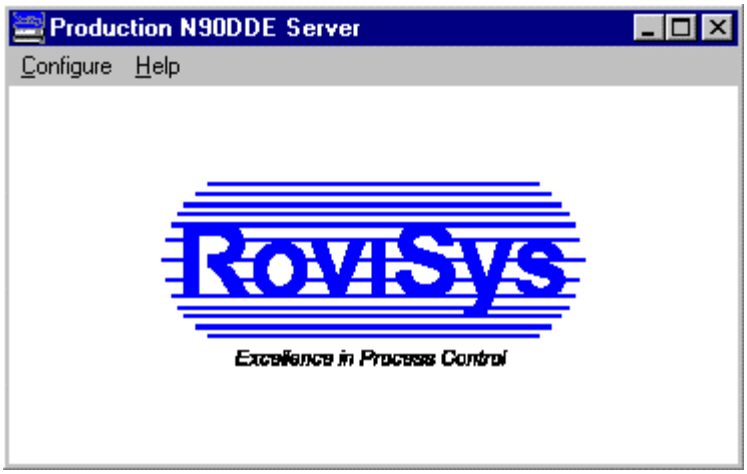
5. Getting Started

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 PC to N90 interface device hardware requirements have been met. The MMI software and N90DDE server has been installed on the target machine, and you are ready to begin defining N90 points that the MMI is to acquire. The following sub-sections demonstrate running the N90DDE server, setting up the COM port, defining a topic, defining server operational parameters and optionally defining N90 time synchronization. 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 N90DDE Server Program

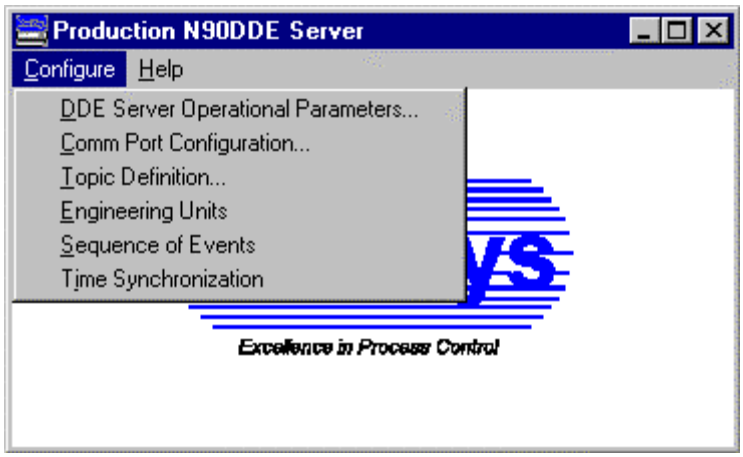
To execute the N90DDE Server program double click the appropriate 

icon, in the RoviSys N90DDE 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 N90DDE Server (minutes left: 60)". *Please, DO NOT utilize the demo copy for control within your facility. Its automatic shutdown after 1 hour of operation could abandon control during a critical operation.*

This screen is used to configure the N90DDE server. To select configure activities, type ALT C or single click on the Configure menu item which will display the following screen:



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

Type ALT C or single click on Comm Port Configuration to configure server COM port communication characteristics.

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

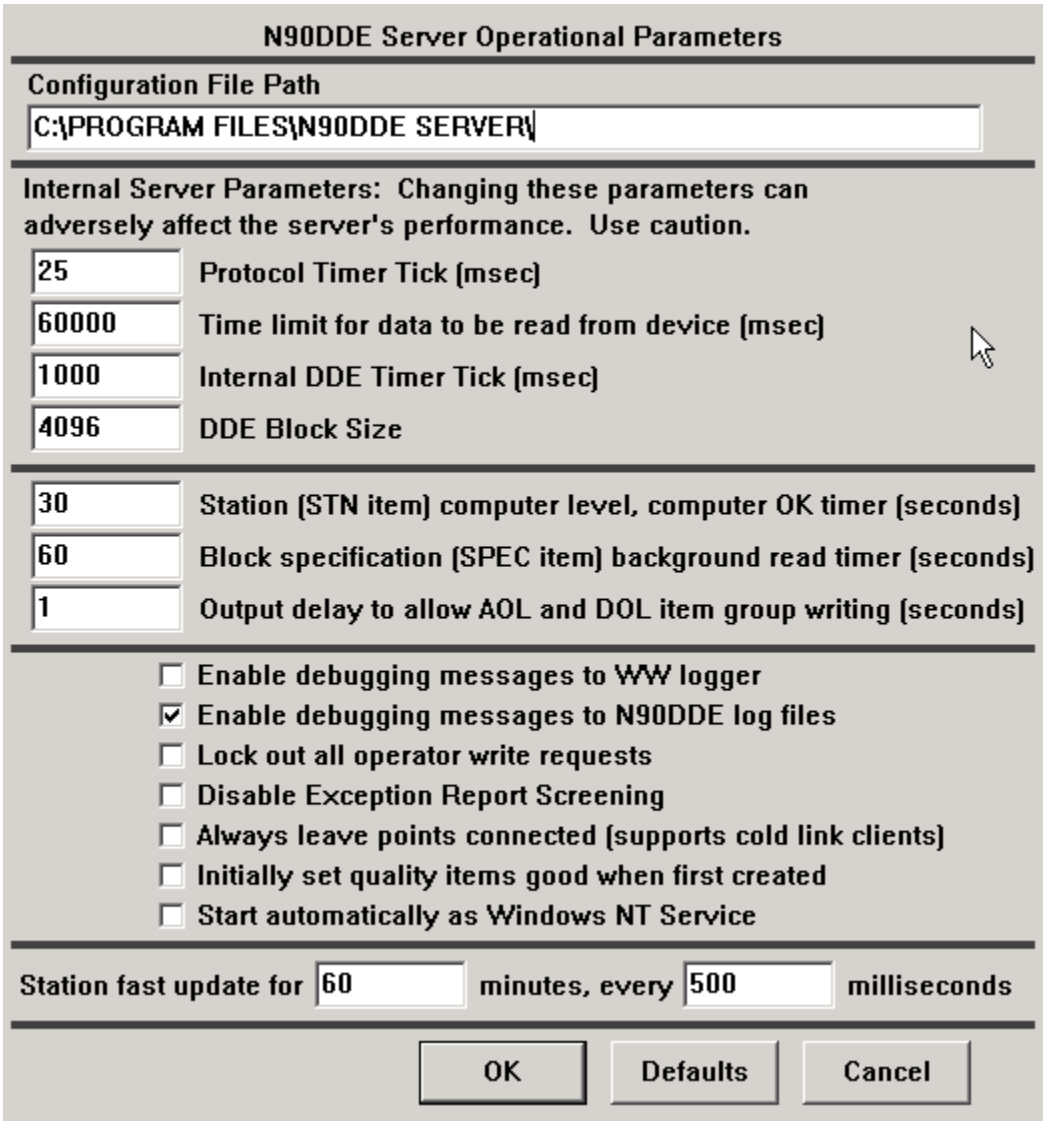
Type ALT E or single click on Engineering Units to configure server engineering unit names.

Type ALT S or single click on Sequence of Events Server to configure SOE capture file storage directories and SOE point names.

Type ALT I or single click on Time Synchronization to enable / disable server time synchronization of N90 and setup node maps for N90 systems based on the Plantloop communication system.

5.2. Defining Operational Parameters

The DDE Server Operational Parameters sub-menu selection allows definition of N90DDE 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 *N90DDE32.CFG*. There is no limit to the number of configuration files created however, each must be saved in a different directory. It is a good practice to organize each application in a separate application directory which will also contain the server configuration file.

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

Time limit for data to be read from device:
is a spare parameter that currently has no functionality within the N90DDE 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.

Station Computer Level, Computer OK Timer :
is the amount of time between sending "Computer OK" messages to STN.MODE DDE tags which have been commanded to the computer level. This parameter should be set to one half of the smallest Bailey station block computer watchdog time period (FC 80, S31 or FC 53, S7) used for computer level control. Most users do not use computer level control and can ignore this setting.

Block Specification Background Read Timer :
is the amount of time between updates of all SPEC.Sx and STN.K* DDE tags.

Output Delay to Allow AOL and DOL Item Group Writing:
is the period of time to wait on updates to AOL and DOL item values before those updates are grouped into a single message and sent to the Bailey interface. When this setting is zero, each AOL or DOL update is sent immediately in individual output messages. By configuring a small delay, the updates are grouped into a single output group message which provides for a more efficient use of the interface serial channel.

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

Enable Debugging Messages to N90DDE Log Files :
is used to enable user selectable debugging messages to daily log files maintained by the server. These log files provide useful information for debugging DDE Client configuration and other runtime problems. This feature is useful for systems that don't have access to the Wonderware logger.

Lock Out All Operator Write Requests:
is used to request N90DDE to lock out write capability for all DDE tags in all topics. See the STATS.LOCK item to programmatically lock out writes on a topic basis.

Disable Exception Report Screening:
is used to turn off the Bailey interface exception reporting screening option. Doing so forces the Bailey interface to report all data updates it receives even if they are identical to the last value reported. Identical values are received as a result of the maximum report time transpiring. Note that enabling this option will slightly reduce data throughput.

Always Leave Points Connected:
is used to request N90DDE to not send a disconnect point command to the Bailey interface for DDE points that are deactivated by the DDE Client. Normally N90DDE connects points that are activated and disconnects them when they are deactivated. This action increases effective throughput of the communication link since only active point data will be exchanged. Cold link DDE clients typically activate, send a point poll request and then deactivate the point. This action can cause stale data to be returned since often the point poll request is received and processed before the point is connected and an updated exception report received. Since this option leaves all points connected the cold link DDE client poll will return most current value. Note that enabling this option will slightly reduce data throughput.

Initially Set Quality Items Good When First Created:
is used to request N90DDE to set quality items "*.Q" good (0) when initially activated instead of defaulting to the bad condition (1) until the point is set up in the Bailey interface and a value received for it.

Start Automatically As a NT Service :
is used to enable N90DDE 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.

Station Fast Update :
is used to define the duration and polling interval for the Station fast update feature. This feature is very useful when using N90DDE with loop tuning software. Its purpose is to acquire the Bailey Station (PID control loop) process variable and control output (STN.PV and STN.CO items) via polling when fast update is activated. Activation of fast update automatically occurs whenever any STN.K* item related to PID tuning parameter is written by the DDE Client. It can also be activated whenever the DDE Client sets the STN.KTUNE item. Fast update stays in effect for the defined duration or until the STN.KTUNE item is reset by the DDE Client. Anytime a STN.K* item is written that tunes one of the PID gains or the STN.KTUNE item is set while fast update is already in effect, the internal timer associated with determining when the fast update duration has expired is reset thus extending the fast update period. The fast update feature is disabled whenever the duration is defined to a value of zero.

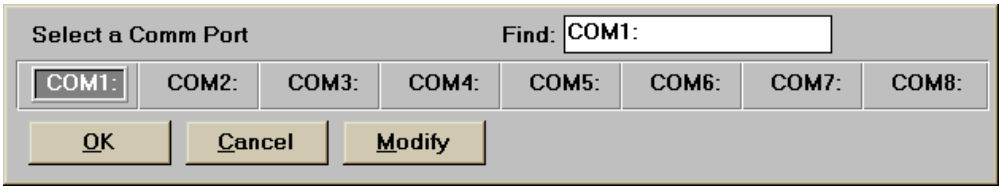
Default Button:
Click on this button to set the server operational parameters to default settings.

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.

5.3. Setting COM Port Communication Characteristics

The Comm Port Configuration sub-menu selection allows configuration of COM port communication characters which the server is to utilize when exchanging data with the N90 interface. When selected the following screen will be displayed:



Find:
Type in the COM port to be modified.

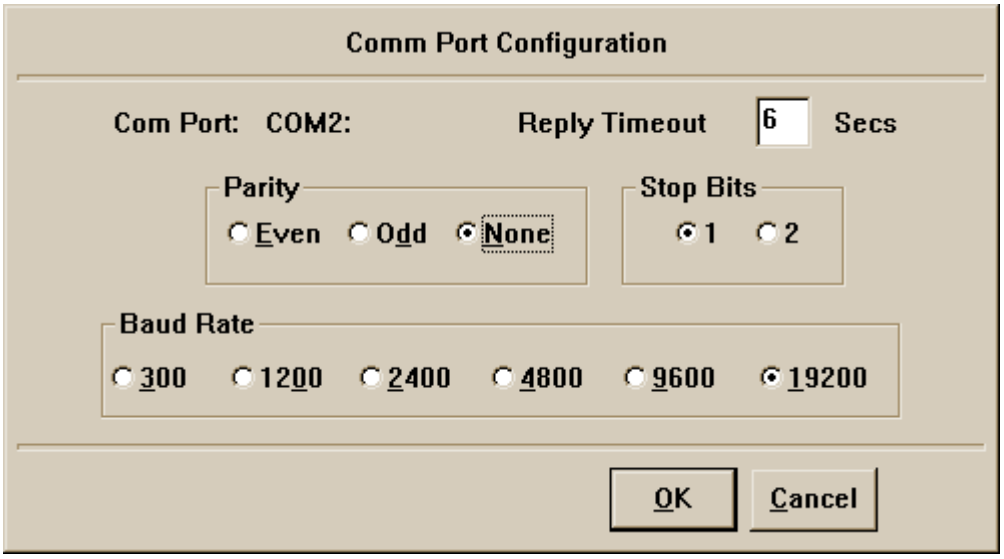
COMX:
Click on one of these boxes to select the specific COM port to be configured. A single click selects the COM port, double click selects the COM port and brings up the Comm Port Configuration dialog.

Modify Button:
Click on this button to bring up the Comm Port Configuration dialog for the selected COM port.

Cancel Button:
Click on this button to cancel selection of a COM port.

OK Button:
Click on this button to accept current COM port selection.

After selecting a COM port the following COMM Port Configuration dialog will be displayed:



This dialog shows the currently selected COM port for which configuration will be take place. It allows configuration of reply time out, parity, stop bits and baud rate. The selected settings must match the N90 interface device. Typically it is 19200 baud, no parity and one stop bit.

5.4. Defining Topics

The Topic Definition sub-menu selection allows definition of N90DDE server topics. A topic is a name which represents a meaningful collection of data from the server. For example, *ICICOM2* could describe data exchange with the Bailey Infi-net to Computer Interface attached to the PC COM2 port. Previous versions of N90DDE only supported one topic being active for any given port. N90DDE version 7.0 and later allow multiple topics to be active for any given COM port with two restrictions. The first is items with identical N90 addresses that receive data via exception reporting cannot be duplicated in multiple topics. The second is the topic with the lowest exception report poll time will be used to determine the actual exception report polling interval. Note that the poll update interval (applies to POLLAO and POLLDO items) remains unique for multiply active topics on the same COM port. The Topic Definition sub-menu selection displays the following dialog:

This dialog shows names of existing topics. Currently this example shows the definition of two topics called ICICOM2 and N90DDE. A new topic can be defined by clicking on the new button. Existing topic definition can be modified by double clicking on it or the modify button. A topic can be deleted by clicking on the delete button.

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

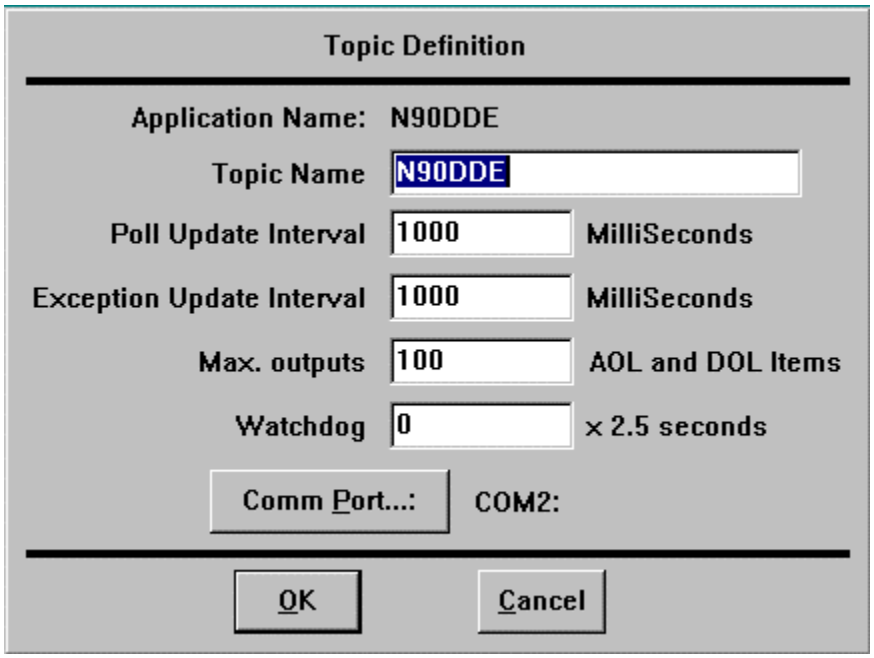
Modify Button:
Click on this button to bring up the topic definition dialog for the selected existing topic.

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 *N90DDE .*

Topic Name:
is a topic name such as *N90DDE* defined by this example.

Poll Update Interval:
frequency in milliseconds at which non-exception report items (POLLAO and POLLDO) are read. A rule of thumb formula for computing this setting is:

$$\frac{1{,}800{,}000 \times \text{sum}(\text{total POLLAO} + \text{total POLLDO})}{\text{COM port baud rate}}$$

Exception Update Interval:
frequency in milliseconds at which exception report items are read from the N90 interface device. Settings in the range of 500 to 3000 milliseconds are common.

Max. outputs:
maximum number of output items (AOL and DOL) the server should allocate point indices for within the N90 interface device. It is important to note that these indices are specified as part of the AOL and DOL item definition and **must** always fall in the range of one to the Max. outputs entered.

Watchdog:
desired watchdog timer to be initiated between the server and N90 interface device. The timer is expressed in 2.5 second counts. A value of zero disables the watchdog timer. The maximum value is 255 which is equivalent to 637.5 seconds. When the watchdog timer is enabled, the N90 interface device will remove itself from communication loop if the elapsed time in which the server communicates with it exceeds the watchdog timer value. Server output item values (AOL and DOL) will thereafter automatically be marked as bad quality.

Comm Port… Button:
Click on this button to select the COM port associated with this topic

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

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

It is important to note that two categories of data collection can occur. The first is polling for the exception reported values and the second is polling for the non-exception values.

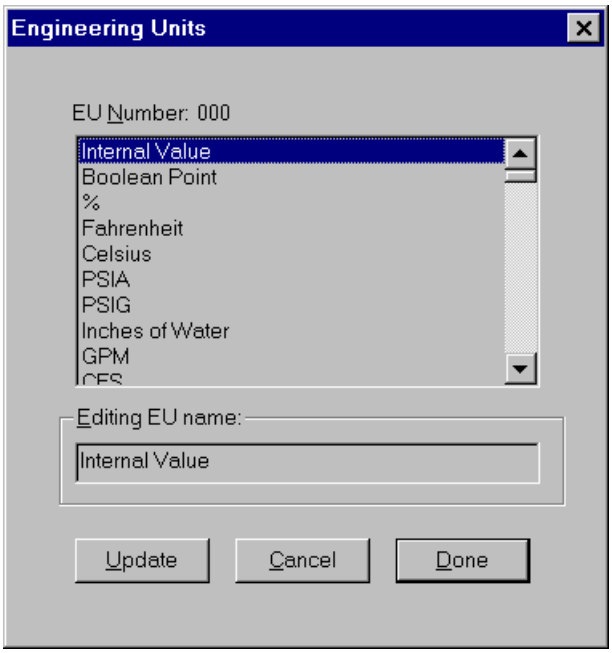
Exception report polling occurs at the rate setup by the *Exception Update Interval*. Its purpose is to acquire the data values that have changed significantly since the last time they were updated.

The response to exception requests contain a count of the number of values returned. Zero indicates nothing has changed since the last read. A non-zero value indicates the number of sets of indices and corresponding status and values. The server issues up to three exception requests before continuing on to the next poll cycle. Since the N90 interface maintains the values that have change significantly in a FIFO queue any values not read after the third command will eventually be read in subsequent exception poll cycles. After all exception commands have been processed, the server waits for the next exception poll cycle to begin.

Non-exception data is polled using the N90 interface READ BLOCK OUTPUT command (setup by the *Poll Update Interval*). Since this command only returns a single data value one must be issued for each POLLAO and POLLDO items configured in the database.

5.5. *Defining Engineering Units*

The Engineering Units sub-menu selection allows mapping of Bailey engineering unit codes to text messages. N90 specifies engineering units as a number in the range of 0 - 255. By definition each number represents a fixed engineering unit. Most N90 exception report function blocks have a specification used to define the engineering unit code. This code can be provided by N90DDE as *.EUNUM items but can also be automatically translated to a text message and made available as *.EUTEXT items. Selection of the engineering units sub-menu displays the following dialog which allows mapping of the N90 engineering unit codes to engineering unit text messages:



EU Number:
indicates the N90 engineering code viewed or defined.

Engineering Units List Box:
lists the current engineering unit definitions and allows selection of any particular engineering unit to determine its number and edit its text message.

Editing EU name:
displays the current engineering unit text name associated with the indicated engineering unit code. A new name can be entered which must be unique with respect to the other currently defined engineering units.

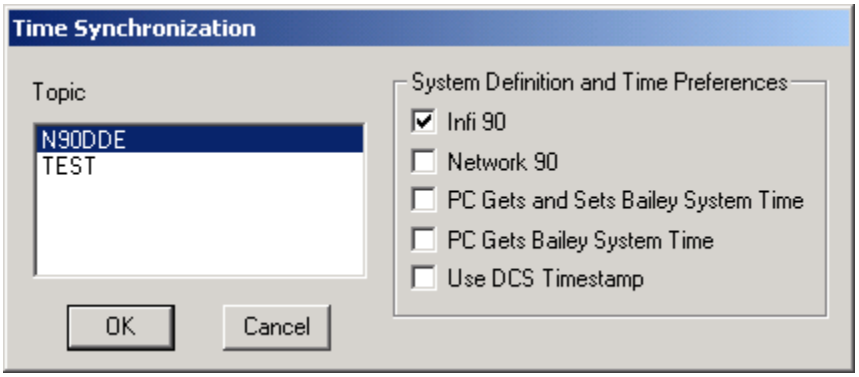
Update Button:
Click on this button to update the edited engineering unit name.

Cancel Button:
Click on this button to cancel changes made to engineering unit names.

Done Button:
Click on this button when engineering unit editing has been completed.

5.6. Defining Time Synchronization

The DDE Server Time Synchronization sub-menu selection allows selection of the type of Bailey system (Infi 90 or Network 90), definition of node maps (when the system type is Network 90) and user time update preferences. These preferences can be PC both gets and sets the Bailey system time, PC just gets the Bailey system time and the use DCS timestamp option that receives Bailey system generated timestamps with each exception reported value instead of the server internally generating those time stamps. Note that the Bailey SPM and CIU01 interfaces do not support any of the indicated time preferences. Selection of the Time Synchronization sub-menu displays the following dialog:



This dialog allows time preferences to be setup for each topic (Bailey interface device). Note that if multiple topics exist and their corresponding Bailey interfaces are all present on the same communication loop, the time preferences should be set to common values for these topics.

Network 90 Check Box:
Specifies a Bailey NETWORK 90 system based on the Plantloop communication system (uses LIM, Loop Interface Modules / BIM, Bus Interface Modules).

Infi 90 Check Box:
Specifies a Bailey INFI 90 system based on the Infinet communication system (uses NIS, Network Interface Slaves / NPM, Network Processing Module).

PC Gets and Sets Bailey System Time Check Box:
The server will set the Bailey system time equal to the current PCs time. With this option the server becomes the Bailey time sync master but will relinquish that role to any MCS or OIU present on the communication system if they take over the time sync mastership. Time synchronization may not function correctly when Bailey OIS stations are present on the communication system. If you observe erratic jumping of the PC clock and/or Bailey OIS stations, this option should be disable. If multiple MMI/N90DDE server nodes are present on the Bailey system, one will become the time sync master and the others will have their PC clocks adjusted to the single master.

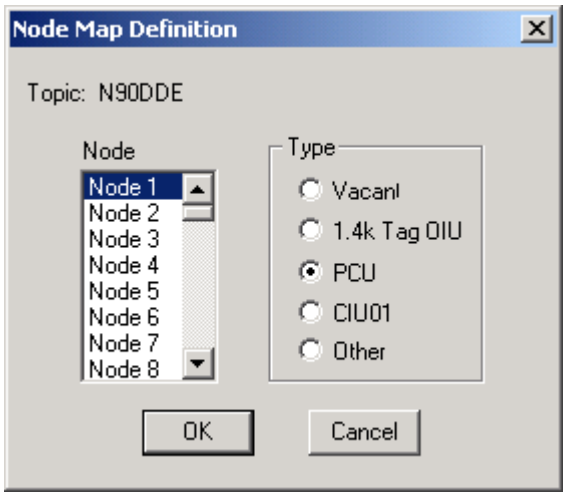
PC Gets Bailey System Time Check Box:
The server will set the PC time to the current Bailey system time. With this option, the server never becomes the Bailey time sync master but always forces the PC clock to match the Bailey system time.

Use DCS Timestamp:
This option is only supported for Infi 90 systems. It causes the server to receive timestamps being generated by the Bailey system for each received exception reported value. This option reduces exception report throughput since each value being received from the Bailey CIU has the time stamp appended to it. Its purpose is intended for a limited number of applications that need more chronologically precise timestamp resolution to aid in a more exact sequence of event determination.

OK Button:
Click on this button to accept time synchronization selections.

Cancel Button:
Click on this button to cancel time synchronization selections.

When the Network 90 check box is selected, the following node map definition dialog is displayed:



Topic:
Displays currently selected topic for which you are defining the node map.

Node:
Displays list box of all possible node addresses within a Network 90 system. Upon initial entry, each node address is defaulted to “Vacant”. For each node address, the actual type must be selected.

Vacant Radio Button:
Indicates that this node address is vacant (unused).

1.4k Tag OIU Radio Button:
Defines selected node as a 1.4k Tag OIU. Do not confuse this type with the 5000 Tag OIU which is classified under the "Other" radio selection.

PCU Radio Button:
Defines selected node as a PCU (Process Control Unit).

CIU01 Radio Button:
Defines selected node as a CIU01 (Computer Interface 01). Do not confuse this type with the other types of CIUs (CIU02/03/04, etc) which are classified under the "Other radio selection.

Other Radio Button:
Use this selection for all other node types not covered by the previous radio selections. This includes 5000 tag OIUs, MCS, OIS and all other Computer Interface devices except CIU01. This would include other MMI/N90DDE server nodes on the Plantloop communication system.

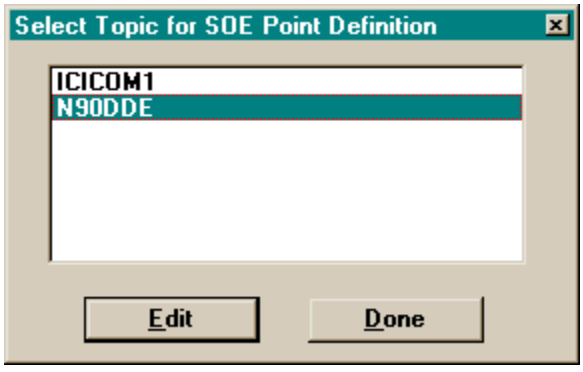
OK Button:
Click on this button to accept node map setup.

Cancel Button:
Click on this button to cancel node map setup.

For Time Synchronization to function correctly, it is very important that ALL nodes be defined and of the right type. Double check your definitions!

5.7. Defining Sequence of Events

The DDE Server Sequence of Events sub-menu selection allows definition of information related to Sequence of Events (SOE) data collection and storage. This features is designed to work in conjunction with the Bailey Sequence of Event function block (F.C. 99) and Controller attached to the Bailey Rochester Sequence of Event Recorder. Selection of the Sequence of Events sub-menu displays the following dialog:

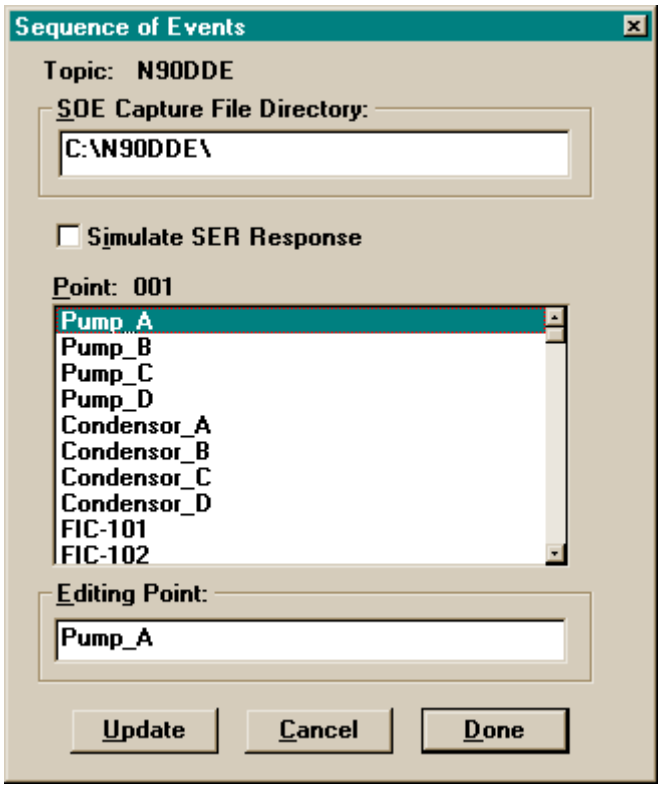


Highlight the topic for which SOE information will be defined by pointing and clicking on it or use the up and down arrow keys. Once the desired topic is selected click on the Edit button or type Alt-E to edit the SOE information related to the selected topic.

Edit Button:
Click on this button to edit the currently highlighted topic.

Done Button:
Click on this button when SOE definition has been completed.

Editing the SOE data for a particular topic displays the following dialog:



Topic:
Displays the current topic name for which this SOE data is being edited.

SOE Capture File Directory:
Defines the directory where the SOE capture files should be read and written for the selected topic. Make sure that the specified directory exists or the SOE capture files will not be saved.

Simulate SER Response:
Checking this box causes N90DDE to simulate the reception of SER event data. When selected it will generated random SOE events occurring approximately every other time an exception report is received from the associated SOE function block. The number of points per SOE event is random but will always be arranged from point one to point N occurring at the same date and time with the milliseconds equaling the point number. The quality, alarm, scan and value will cycle between all zeros and all ones. This option is defined to allow checkout of the client application SOE data management and presentation. Make sure this option is **not** selected when the client application is put into actual in plant use.

Point:
Displays the current SOE point number selected in the list and edit boxes. The point list box directly below the point number displays the currently defined SOE point names and allow selection of a point for editing its name. To select a point use the up and down arrow keys (when the list box has focus) or point and click on the point to be edited.

Edit Point:
Edit box used to edit the currently selected point. To give the edit box focus press ALT E or the tab key multiple times. When the edit box has focus, the edited point can be changed and added to the point list box by entering ALT U (update). Note that point names can be up to 32 characters in length, cannot contain spaces and must be unique. N90DDE will enforce these rules.

Update Button:
Click on this button to update the edited SOE point name.

Cancel Button:
Click on this button to cancel changes made to the SOE point names.

Done Button:
Click on this button when SOE point editing has been completed.

6. N90DDE Server Startup Activities

This section presents a technical overview of the various server startup activities associated with the N90 interface. **It is presented for informational purposes only and its understanding is not required for configuring third party MMI applications and the N90DDE server.**

6.1. Environment Identification

The first thing the server must accomplish is to determine what type of N90 interface device it is communicating with. Knowing the type allows the server to automatically adjust its performance to optimize its use of the interface and follow N90 interface specific rules of operation. Environment identification involves a series of commands and ranges of certain fields within the commands to determine the exact type of interface. This process starts with the ENVIRONMENT command but may also utilize the CIU RESTART and ESTABLISH POINT commands depending on the actual type of interface being utilized and whether or not it supports the ENVIRONMENT command.

6.2. Restarting Interface

Next the server must "restart" the N90 interface device. Restarting an interface clears its database and places the device on-line to the N90 system. The "CIU RESTART" command is used for this purpose. Included in this command is a field that instructs the interface device to disable itself from the N90 communication loop if it hasn't communicated with the server in a specific period of time. The time out value specified in the topic definition is utilized for this purpose.

6.3. Database Establishment

All N90DDE server items except POLLAO and POLLDO utilize exception reporting to acquire the data values. For each unique N90 block defined by MMI tag name items, an entry must be setup in the N90 interface database. The "ESTABLISH POINT" or "ESTABLISH AND CONNECT message is utilized for this purpose. This message requires six data fields to be specified. These fields are index, point type, ring, node, module and block number. The index number is a unique number assigned to each point established in the N90 interface. Indices are allocated starting at the maximum number of outputs (defined by the topic) plus one and proceeds sequentially higher as each MMI tag name item is processed.

As the name implies, the point type defines the type of exception report to be expected from the N90 block. The following N90DDE server item name to N90 interface point type relationship exists.

N90DDE Server Item Name	Direction of Data Flow	N90 Interface Point Type
AIL	import	5
AOL	export	12
DAANG	import/export	1, 2, 3, 4, 6, 8, 9, 10, 11, or 17
DD	import/export	15
DIL	import	7
DOL	export	13
MODSTAT	import	14, 23
MSDD	import/export	15
RCM	import/export	15
RMC	import/export	15
RMSC	import/export	19
SOE	import/export	15
STN	import/export	1, 2, 3, 4, 6, 8, 9, 10, 11, or 17
TEXT	import	5

For the N90DDE server item DAANG and STN, the actual N90 point types utilized depends on the type of N90 interface in use. The N90 NCIU01 interface requires a separate index to be established for each daang and station value read and written whereas, the other N90 interfaces support a single index to handle all the daang and station values.

6.4. Addressing

For N90 import points, the ring, node, module and block field values are defined by the N90DDE server item. Valid N90 block numbers are dependent on the type of N90 communication system. Systems based on Plant Loop allow a valid block range of 1 through 1023. Infinet based systems allow a valid block range of 1 through 9998.

The N90 export points are those values sent from the server into N90. The N90DDE server item names for these points are AOL and DOL which implements Analog Outputs and Digital Outputs to N90. These point types respond to N90 as if they are Function blocks 30 and 45 respectively. They are read by N90 at the ring and node address of the N90 interface device. The module number is always two and the block number is the index number assigned in the definition of the N90DDE server item. Note that this index number must be unique with respect to all other export points item definitions. It also must fall within the range of one to the maximum number of outputs defined by the N90DDE server topic.

6.5. Runtime Error Handling

Two categories of errors could occur as each point is established in the N90 interface database. The first category includes those types related to exceeding the capacity of the N90 interface. This includes memory or index numbers. For either of these situations, a message is posted in the WWLogger message window (if running WW version N90DDE) indicating the error, tag name, application, topic, item and N90 index assigned by the server. The second category is caused by duplicating N90 block numbers across multiple server items but with different point types. In this case the duplicated N90 block number is rejected by the N90 interface and will not receive data values. Like the first category, these errors are posted in the WWLogger message window along with the tag name, application, topic, item and N90 index assigned by the server.

6.6. Data Flow Enabling

To activate exception report data flow a "CONNECT" message must be sent for each point established in the N90 interface database.

All N90 interfaces except the NSPM01, NCIU01 and IMSPM01 allow the interface to be restarted and remain off-line until commanded on-line. For these N90 interfaces the database download can occur considerably faster than when restarted on-line. Under this condition, the last activity of startup is to command the N90 interface on-line utilizing the "CIU ONLINE/OFFLINE" command.

7. Process Data Items

N90DDE server process data items map the MMI DDE tag types to specific N90 point types. Export point type items are those in which data values are being generated by the MMI and reported by exception to N90 by the server. Supported point types are analog and digital with each having its own unique naming convention (see respective sub-section).

Import point types are those in which data values are being generated by N90 and reported by exception to the server or polled by the server. The naming convention for import items is as follows:

pointType.valueSubscript, ring, node, module, block

where:

Item field	Purpose
pointType	type of N90 point
valueSubscript	data item to be extracted for the N90 point
ring	N90 ring address, 1-250 Infinet, 1 PlantLoop
node	N90 node address, 1-250 Infinet, 1-63 PlantLoop
module	N90 module address 2 - 31
block	N90 block number 0 - 9998

When using a Bailey CIC interface the ring and node fields must be set to values of zero when defining the item.

Each point within a DCS system typically has several pieces of data. For instance a control loop is a single point within N90 but has many different values. For example, there is process variable, set point, control output and mode to name a few. Within the MMI each piece of data is a tag name. Therefore, for the N90DDE server, in some instances such as a control loop, there will be multiple MMI tag names each with an item addressing the same point address within N90 but requesting a different piece of data. The "valueSubcript" field of the item selects the specific data to be accessed. The following table lists all possible valueSubcripts supported by the server.

ValueSubscript	Purpose	DDE Type	Access
.ACKNAK	result of the last block request	integer	read
.ALMSTATE	alarm state: 0 - none, 1 - low, 2 - high, 3 - low & high	integer	read
.ALMLEVEL	alarm level: 0 - one, 2 - two, 3 - three	integer	read
.ALM	alarm: 0 - none, 1 - alarm	discrete	read
.AOBYPASS	analog output bypassed: 0-no, 1-yes	discrete	read
.BLINK	message blink code	discrete	read
.BYTE1	value of module status byte number 1	integer	read
.BYTE2	value of module status byte number 2	integer	read
.BYTE3	value of module status byte number 3	integer	read
.BYTE4	value of module status byte number 4	integer	read
.BYTE5	value of module status byte number 5	integer	read
.CALC	calculated flag: 0-no, 1-yes	discrete	read
.CO	control output	real	read/write
.COLOR	message color code	integer	read
.COMM	communication: 0 - good, 1 - bad	discrete	read
.COTRACK	control output track: 0-no, 1-yes	discrete	read
.COUNT	count of total SOE points in a given capture file	integer	read
.CPU	computer status: 0 - failed, 1 - OK	discrete	read/write
.DDERATE	DDE message per second	integer	read
.DEVALM	deviation alarm: 0 - none, 1 - low, 2 - high	integer	read
.DEVALMLIM	deviation alarm limit	real	read/write
.DSBAD	digital station bad: 0-no, 1-yes	discrete	read
.ERRCODE	error codes	integer	read
.ERRORS	module indicates one or more runtime errors	discrete	read
.EUNUM	engineering unit number	integer	read
.EUTEXT	engineering unit text message	message	read
.EVENT	signals that an SOE event has been captured	discrete	read/write
.F1 - .F4	feedback #1 - #4	discrete	read
.FB	feedback	discrete	read
.FBAD	feedback bad: 0 - no, 1 - yes	discrete	read
.FC	function code	integer	read
.FILE	indicates / selects current SOE capture file	message	read/write
.FLT	fault has occurred	discrete	read
.FLTACK	fault acknowledgment	discrete	read/write
.GS	good state: 0 - default, 1 - state1, 2 - state2, 3 - state3	integer	read
.GSTEXT	good state as text	message	read
.HARDFAIL	hardware failure: 0 - no, 1 - yes	discrete	read
.HIALM	high alarm: 0 - no, 1 - yes	discrete	read
.HIALMLIM	high alarm limit	real	read/write
.HIDEV	high deviation alarm: 0 - no, 1 - yes	discrete	read
.HILIM	high limit value	real	read
.K	PID gain multiplier	real	read/write
.KD	PID derivative constant (minute)	real	read/write
.KDIR	PID direction (0 = reverse, 1 = direct)	discrete	read/write
.KDLAG	PID derivative lag constant (F.C 156 only)	real	read/write
.KFC	PID function code number	Integer	read
.KI	PID integral constant (1/minute)	real	read/write
.KIONLY	PID integral term only	discrete	read/write
.KHILIM	PID output high limit	real	read/write
.KLOLIM	PID output low limit	real	read/write
.KP	PID proportional constant	real	read/write
.KPIDBLK	PID block number associated with station	integer	read/write
.KTUNE	PID tune notification, invokes fast STN update	discrete	read/write
.KTYPE	PID algorithm type indicator (see F.C. 156, S18)	integer	read

.LEVEL	level of station control: 0 - local, 1 - computer	discrete	read
.LIMIT	limited flag: 0 - no, 1 - yes	discrete	read
.LOALM	low alarm: 0 - no, 1 - yes	discrete	read
.LOALMLIM	low alarm limit	real	read/write
.LODEV	low deviation alarm: 0 - no, 1 - yes	discrete	read
.LOCK	Operator write lock flag: 0 - Unlocked, 1 - locked	discrete	read/write
.LOLIM	low limit value	real	read
.MODE	mode: 0 - local manual, 3 - computer manual, 1 - local auto, 4 - computer auto, 2 - local cascade 5 - computer cascade	item based	read/write
.MODELOCK	current mode locked: 0 - no, 1 - yes	discrete	read
.MSGRATE	interface messages per second	integer	read
.NEXTFILE	flag used to sequence through SOE capture files	discrete	read/write
.NEXTHI	next high limit	real	read
.NEXTLO	next low limit	real	read
.NEXTREC	flag used to sequence through SOE capture file records	discrete	read/write
.OFLOW	SOE overflow indication	discrete	read
.ONLINE	flags the Bailey interface has been determined and is online	discrete	read
.OUTRANGE	out of range: 0 - no, 1 - yes	discrete	read
.OVRC	override control: 0 - no, 1 - yes	discrete	read
.OVRS	override status: 0 - no, 1 - yes	discrete	read
.PERM1	permissive #1	discrete	read
.PERM2	permissive #2	discrete	read
.PNTALM	SOE point alarm: 0 - no, 1 - yes	discrete	read
.PNTDATE	SOE point date like 3/08/1997	message	read
.PNTNAME	SOE point name like Point_1	message	read
.PNTNUM	SOE point number 1 - 512	integer	read
.PNTQ	SOE point quality: 0 - good, 1 - bad, inactive	discrete	read
.PNTSCAN	SOE point scan flag: 0 - yes, 1 - deleted from scan	discrete	read
.PNTTIME	SOE point time like 23:30:02.123	message	read
.PNTTYPE	SOE point type: Standard, Summary, Pre-fault, Post-fault, Snapshot	message	read
.PNTV	SOE point value: 0 - reset, 1 - set	discrete	read
.POLLONLY	flags Bailey device or N90DDE Server license only supports polling items (POLLAO and POLLDO): 0 - no, 1 - yes	discrete	read
.POLLRATE	interface polling messages per second	integer	read
.PV	process variable	real	read
.Q	quality: 0 - good, 1 - bad	discrete	read
.RATEALM	rate alarm: 0 - none, 1 - low, 2 - high, 3 - low & high	integer	read
.RECORD	current SOE capture file record	integer	read/write
.RI	ratio index	real	read/write
.RS	requested state: 0 - default, 1 - state1, 2 - state2, 3 - state3	integer	read/write
.RSTEXT	requested state as text	message	read/write
.S1 - .S64	addressed block specification value	real	read/write
.SBAD	bad start has occurred	discrete	read
.SCAN	scan flag: 0 - on scan, 1 - off scan	discrete	read
.SETPERM	set permissive: 0 - no, 1 - yes	discrete	read
.SHOLD	status on hold	discrete	read
.SP	set point	real	read/write
.SPAN	span of process variable, set point or value	real	read
.SPTRACK	set point track: 0 - no, 1 - yes	discrete	read
.STATE	current module operating state (execute, configure, error)	message	read

.SUPALM	suppressed alarms: 0 - no, 1 - yes	discrete	read
.TAGS	total DDE tags established with the server	integer	read
.TIME	time (expressed as seconds since 1/1/1970) of last point value update	integer	read
.TIMETEXT	time (month, day, year, hour, minute, second) of last point value update	message	read
.TOTALDDE	total DDE messages sent to client(s)	integer	read
.TOTALMSG	total messages exchanged with interface	integer	read
.TOTALPOLL	total polling messages sent to interface	integer	read
.TOTALXRP	total exception reports exchanged with interface	integer	read
.TRV	travel indicator: 0 - no, 1 - yes	discrete	read
.TYPE	faceplate or module type	item based	read
.TYPETEXT	type of faceplate as text	message	read
.TYPEBLK	type of SOE log block: Not Determined, Standard, Summary, Pre-fault, Post-fault, Snapshot	message	read
.UPDATES	running count of updates received from N90 for a point	integer	read
.V	value	item based	item based
.VARALM	variable alarm: 0 - no, 1 - yes	discrete	read
.VERSION	N90DDE Server version number	real	read
.VTEXT	value as text or logic state descriptor	item based	item based
.XRPRATE	exception reports per second exchanged with interface	integer	read
.ZERO	zero value of value range	real	read
.ZEROPV	zero value of process variable range	real	read
.ZEROSP	zero value of set point range	real	read

The following table gives an overview of the server item point types, supported valueSubscript, corresponding N90 block type, N90 function code, and whether or not the block is of the type that permits control.

pointType names	supported subscript value	N90 Block Type	N90 Function Code	Control
AIL	.EUNUM .EUTEXT .HIALM .HIALMLIM .LOALM .LOALMLIM .Q .SPAN .TIME .TIMETEXT .UPDATES .V .ZERO	Analog Output / Loop	30, 70, 158	No
AOL	.V	Analog Input / Loop	26, 121	No
DAANG	.ALMLEVEL .ALMSTATE .CALC .DEVALM .DEVALMLIM .EUNUM .EUTEXT .HARDFAIL .HIALMLIM .LIMIT .LOALMLIM .MODE .NEXTHI .NEXTLO .OUTRANGE .Q .RATEALM .SCAN .SPAN .SUPALM .TIME .TIMETEXT .UPDATES .V .VARALM .ZERO	Data Acquisition Analog	177	Yes
DD	.ALM .F1 .F2 .FBAD .MODE .OVRS .Q .TIME .TIMETEXT .TYPE .UPDATES .V .VTEXT	Device Driver	123	Yes
DIL	.ALM .ALMSTATE .Q .TIME .TIMETEXT .UPDATES .V .VTEXT	Digital Output / Loop	45	No
DOL	.V	Digital Input / Loop	42, 122	No
MODSTAT	.BYTE1 .BYTE2 .BYTE3 .BYTE4 .BYTE5 .ERRORS .Q .STATE .TIME .TIMETEXT .UPDATES .TYPE	-	-	No
MSDD	.ALM .F1 .F2 .F3 .F4 .GS .GSTEXT .MODE .OVRC .OVRS .Q .RS .RSTEXT .TIME .TIMETEXT .TYPE .UPDATES .TRV .V .VTEXT	Multi-State Device Driver	129	Yes
POLLAO	.LOALM .HIALM .Q .TIME .TIMETEXT .UPDATES .V	All Analog Block Outputs	All Analog	No
POLLDO	.ALM .Q .TIME .TIMETEXT .UPDATES .V .VTEXT	All Digital Block Outputs	All Digital	No
RCM	.ALM .FB .Q .SETPERM .TIME .TIMETEXT .TYPE .UPDATES .V .VTEXT	Remote Control Memory	62	Yes
RMC	.ALM .ERRCODE .F1 .F2 .FLT .FLTACK .PERM1 .PERM2 .Q .SBAD .SHOLD .TIME .TIMETEXT .TYPE .UPDATES .V .VTEXT	Remote Motor Control	136	Yes
RMSC	.EUNUM .EUTEXT .HILIM .LOLIM .Q .SPTRACK .TIME .TIMETEXT .UPDATES .V	Remote Manual Set Constant	68	Yes
SOE	.COUNT .EVENT .FILE .NEXTFILE .NEXTREC .OFLOW .PNTALM .PNTDATE .PNTNAME .PNTNUM .PNTQ .PNTSCAN .PNTTIME .PNTTYPE .PNTV .Q .RECORD .TIME .TIMETEXT .UPDATES .TYPEBLK .V	Sequence of Events	99	No

STATS	.COMM .DDERATE .LOCK .MSGRATE .ONLINE .POLLONLY .POLLRATE .SYNCRING .SYNCNODE .TAGS .THISRING .THISNODE .TOTALDDE .TOTALMSG .TOTALPOLL .TOTALXRP .VERSION .XRPRATE	-	-	No
STN	.AOBYPASS .CO .COTRACK .CPU .DEVALMLIM .DSBAD .EUNUM .EUTEXT .HIALM .HIALMLIM .HIDEV .K .KD .KDIR .KDLAG .KFC .KI .KHILIM .KIONLY .KLOLIM .KP .KPIDBLK .KTUNE .KTYPE .LEVEL .LOALM .LOALMLIM .LODEV .MODE .MODELOCK .PV .Q .RI .SP .SPAN .SPTRACK .TIME .TIMETEXT .TYPE .TYPETEXT .UPDATES .ZEROPV .ZEROSP	Control Station	21-23, 80	Yes
TEXT	.BLINK .COLOR .Q .TIME .TIMETEXT .UPDATES .V	Text Selector	151	No

The following table is an overview of each NETWORK 90, INFI 90 or COMMAND SERIES computer interface device and its ability to handle the N90DDE Server process data items:

Item name	NSPM01 IMSPM01 IMCPM02	NCIC01 IMCPM03	NCIU01	NCIU02 NCIU03 NCIU04	INPCI01 INPCI02	INICI01 INICI03* INICI12*
AIL	No	Yes	Yes	Yes	Yes	Yes
AOL	No	No	Yes	Yes	Yes	Yes
BLOCK	Yes	Yes	Yes	Yes	Yes	Yes
DAANG	No	Yes	Yes	Yes	Yes	Yes
DD	No	Yes	Yes	Yes	Yes	Yes
DIL	No	Yes	Yes	Yes	Yes	Yes
DOL	No	No	Yes	Yes	Yes	Yes
MODSTAT	No	Yes	Yes	Yes	Yes	Yes
MSDD	No	Yes	Yes	Yes	Yes	Yes
POLLAO	Yes	Yes	Yes	Yes	Yes	Yes
POLLDO	Yes	Yes	Yes	Yes	Yes	Yes
RCM	No	Yes	Yes	Yes	Yes	Yes
RMC	No	Yes	Yes	Yes	Yes	Yes
RMSC	No	Yes	Yes	Yes	Yes	Yes
SOE	No	No	Yes	Yes	Yes	Yes
SPEC	Yes	Yes	Yes	Yes	Yes	Yes
STATS	Yes	Yes	Yes	Yes	Yes	Yes
STN	No	Yes	Yes	Yes	Yes	Yes
Time Sync	No	No	No	Yes	Yes	Yes
TEXT	No	Yes	Yes	Yes	Yes	Yes

* When using these interfaces, the ABB Bailey semAPI software environment is not required. Note that the INICI12 and INICI03 only support communication on the ABB Bailey termination unit or termination module port, labeled “terminal”. Currently the INICI03 SCSI connection is not supported.

* INICI03 sites containing INICT03A firmware revision ‘F’ require the ABB Bailey semAPI dongle to be installed on the printer port of the NTMP01 termination unit. The dongle is not required for firmware revisions prior to revision ‘F’ nor revisions ‘G’ or later. Please note that N90DDE will continue to operate when revision ‘F’ firmware is detected and the dongle is missing. After 70 hours of continuous operation, ABB Bailey has designed the INICI03 to lock out all communication until the dongle is installed and the interface physically reset. N90DDE will attempt to automatically restart the INICI03 every 68 hours to avoid the lock out condition. The recommended mode of operation for sites utilizing an INICI03 with revision ‘F’ firmware installed in the INICT03A module is to also have the ABB Bailey semAPI dongle installed on the NTMP01 printer port.

7.1. Analog Input / Loop - AIL

The AIL item point type is used to retrieve the exception reported output from N90 Analog Output / Loop blocks (function code 30), Analog Point Definition (function code 70) and Enhanced Analog Point Definition (function code 158).

Name: AIL

Restrictions: This item can be utilized with all N90 interface types except the serial port module. Ring and node addresses must be set to zero when the N90 interface is a CIC.

Subscripts	Purpose	DDE Type	Access
.EUNUM	engineering unit number	integer	read
.EUTEXT	engineering unit text message	message	read
.HIALM	high alarm indicator (0 - no, 1 - yes)	discrete	read
.HIALMLIM	high alarm limit	real	read/write
.LOALM	low alarm indicator (0 - no, 1 - yes)	discrete	read
.LOALMLIM	low alarm limit	real	read/write
.Q	quality (0 - good, 1 - bad)	discrete	read
.SPAN	span of value	real	read
.TIME	time (expressed as seconds since 1/1/1970) of last point value update	integer	read
.TIMETEXT	time (month, day, year, hour, minute, second) of last point value update	message	read
.UPDATES	running count of updates received for this point	integer	read
.V	value	real	read
.ZERO	zero value of value range	real	read

Example usage's are:

AIL.V,1,2,3,50
Reads N90 AO/L value at ring 1, node 2, module 3, block 50 into a DDE real tag.

AIL.Q,1,2,3,50
Reads N90 AO/L quality at ring 1, node 2, module 3, block 50 into a DDE discrete tag.

AIL.ZERO,1,2,3,50
Reads zero setting for N90 AO/L at ring 1, node 2, module 3, block 50 into a DDE real tag.

AIL.SPAN,1,2,3,50
Reads span setting for N90 AO/L at ring 1, node 2, module 3, block 50 into a DDE real tag.

AIL.HIALMLIM,1,2,3,50
Reads N90 AO/L high alarm limit at ring 1, node 2, module 3, block 50 into a DDE real tag.

AIL.HIALM,1,2,3,50
Reads N90 AO/L high alarm indicator at ring 1, node 2, module 3, block 50 into a DDE discrete tag.

AIL.LOALMLIM,1,2,3,50
Reads N90 AO/L low alarm limit at ring 1, node 2, module 3, block 50 into a DDE real tag.

AIL.LOALM,1,2,3,50
Reads N90 AO/L low alarm indicator at ring 1, node 2, module 3, block 50 into a DDE discrete tag.

AIL.EUNUM,1,2,3,50
Reads engineering unit number assigned to N90 AO/L at ring 1, node 2, module 3, block 50 into a DDE integer tag.

AIL.EUTEXT,1,2,3,50
Reads engineering unit number assigned to N90 AO/L at ring 1, node 2, module 3, block 50 and converts it to a text message (defined by user within N90DDE engineering units setup dialog) written to a DDE message tag.

AIL.V,2,249,31,9998
Reads AO/L value at ring 2, node 249, module 31 block 9998 into a DDE real tag.

AIL.TIME,2,249,31,9998
Returns the time expressed as seconds since 1/1/1970 of last update received from an AO/L value at ring 2, node 249, module 31 block 9998 into a DDE integer tag.

AIL.TIMETEXT,2,249,31,9998
Returns the time expressed as a text message like “Mon Oct 26 08:00:00 1998”of last update received from an AO/L value at ring 2, node 249, module 31 block 9998 into a DDE message tag.

AIL.UPDATES,2,249,31,9998
Returns a running count of the total updates received from an AO/L value at ring 2, node 249, module 31 block 9998 into a DDE integer tag.

7.2. Analog Output / Loop - AOL

The AOL item point type is used to send an exception reported output generated by the MMI into N90 Analog Input / Loop blocks (function code 26) and Analog Input / Infinet blocks (function code 121). The only supported value subscript is .V which is a DDE real read/write type. However, the N90DDE server will generate the appropriate quality, and alarming status based on the value being generated by the MMI and definition of the tag item. Since this item exports data to N90, it follows a different naming convention when defining it within a MMI tag.

Restrictions: This item can be utilized with all N90 interface types except serial port module (SPM) and computer interface command series (CIC).

Cautions: Make sure the block number field is unique with respect to the other export item definitions and is defined within the range of 1 to the maximum number of allowed outputs. This maximum is defined within the server topic selected for the MMI DDE tag. N90 receives data from this item at the ring and node address assigned to the N90 interface, module two and defined block number.

Name: AOL.V, Block, EU, Sig Change, Zero, Span, HA, LA, Init Val

Item field	Purpose
AOL	item point name
.V	data subscript is read/write DDE real
Block	block number at which value is to be established within N90 interface device. Its range is 1 to max output defined by topic definition.
EU	engineering units number (used by Bailey console nodes)
Sig Change	percent significant change when exception report is to be generated
Zero	zero of value in engineering units
Span	span of value in engineering units
HA	high alarm limit in engineering units
LA	low alarm limit in engineering units
Init Val	initial value of output when first established in N90 interface

Example usage's are:

AOL.V,1,2,0.1,0.0,100.0,95.0,5.0,0.0
Writes a value to N90 interface block 1 which has an engineering code of 2, significant change of .1 percent, zero of 0.0, span of 100.0, high alarm of 95.0, low alarm of 5.0 and its initial value will be set to 0.0.

AOL.V,10,3,1.0,-1.0,2,.5,-.9,0.0
Writes a value to N90 interface block 10 which has an engineering code of 3, significant change of 1.0 percent, zero of -1.0, span of 2.0, high alarm of .5, low alarm of -.9 and its initial value will be set to 0.0.

7.3. Data Acquisition Analog - DAANG

The DAANG item point type is used to retrieve and control the exception reported output from N90 Data Acquisition Analog blocks (function code 177).

Name: DAANG

Restrictions: This item can be utilized with all N90 interface types except the serial port module. Ring and node addresses must be set to zero when the N90 interface is a CIC.

Subscripts	Purpose	DDE Type	Access
.ALMLEVEL	alarm level: 0 - level one 2 - level two 3 - level three	integer	read
.ALMSTATE	alarm state: 0 - none 1 - low alarm 2 - high alarm 3 - low & high alarm	integer	read
.CALC	calculated value (0 - no, 1 - yes)	discrete	read
.DEVALM	deviation alarm state: 0 - none 1 - low 2 - high	integer	read
.DEVALMLIM	deviation alarm limit	real	read
.EUNUM	engineering unit number	integer	read
.EUTEXT	engineering unit text message	message	read
.HARDFAIL	hardware failure, bad quality input (0 - no, 1 - yes)	discrete	read
.HIALMLIM	high alarm limit	real	read
.LIMIT	value limited (0 - no, 1 - yes)	discrete	read
.LOALMLIM	low alarm limit	real	read
.MODE	mode: 0 - manual 1 - auto input 2 - auto-calculated input 3 - suppress alarms 4 - unsuppress alarms 5 - off scan 6 - on scan	integer	read/write*
.NEXTHI	next high alarm limit	real	read
.NEXTLO	next low alarm limit	real	read
.OUTRANGE	out of range flag (0 - no, 1 - yes)	discrete	read
.Q	quality (0 - good, 1 - bad)	discrete	read
.RATEALM	rate alarm (0 - no, 1 - yes)	discrete	read
.SCAN	scanning (0 - on, 1 - off)	discrete	read
.SPAN	span of value	real	read
.SUPALM	suppress alarms (0 - no, 1 - yes)	discrete	read
.TIME	time (expressed as seconds since 1/1/1970) of last point value update	integer	read
.TIMETEXT	time (month, day, year, hour, minute, second) of last point value update	message	read

.UPDATES	running count of updates received for this point	integer	read
.V	value	real	read/write**
.VARALM	variable alarms (0 - no, 1 - yes)	discrete	read
.ZERO	zero value of value range	real	read

* mode is always read back as zero for manual and 1 for auto. It is also used to command the other aspects (ranges 2 - 6) which are reported back in their corresponding subscript tag. For example commanding a mode of two will cause mode to report back as auto (1) and CALC to be yes (1).

** value should only be written when mode is manual.

Example usage's are:

DAANG.V,11,13,14,100
Reads/writes N90 DAANG value at ring 11, node 13, module 14, block 100 to/from a DDE real tag.

DAANG.Q,11,13,14,100
Reads N90 DAANG quality at ring 11, node 13, module 14, block 100 into a DDE discrete tag.

DAANG.MODE,1,2,4,90
Reads/writes N90 DAANG mode at ring 1, node 2, module 4, block 90 into a DDE integer tag.

7.4. Device Driver - DD

The DD item point type is used to retrieve and control the exception reported output from N90 Device Driver blocks (function code 123).

Name: DD

Restrictions: This item can be utilized with all N90 interface types except the serial port module. Ring and node addresses must be set to zero when the N90 interface is a CIC.

Subscripts	Purpose	DDE Type	Access
.ALM	alarm (0 - none, 1 - alarm)	discrete	read
.F1	feedback #1	discrete	read
.F2	feedback #2	discrete	read
.FBAD	feedback bad (0 - no, 1 - yes)	discrete	read
.MODE	mode: 0 - manual 1 - auto	discrete	read/write
.OVRS	override status (0 - no, 1 - yes)	discrete	read
.Q	quality (0 - good, 1 - bad)	discrete	read
.TIME	time (expressed as seconds since 1/1/1970) of last point value update	integer	read
.TIMETEXT	time (month, day, year, hour, minute, second) of last point value update	message	read
.TYPE	faceplate type code	integer	read
.UPDATES	running count of updates received for this point	integer	read
.V	value	discrete	read/write
.VTEXT	value as text (defined at point creation)	message	read/write

Example usage's are:

DD.V,10,12,14,102
Reads/writes N90 DD value at ring 10, node 12, module 14, block 102 to/from a DDE discrete tag.

DD.VTEXT,10,12,14,102,Stopped,Running
Reads/writes N90 DD value at ring 10, node 12, module 14, block 102 to/from a DDE message tag. When the N90 DD value is zero, the message “Stopped” will be returned, when one “Running” will be returned. These messages can also be written to reset and set the N90 DD. Also text messages of “0” or “1” can be written to reset and set the N90 DD.

DD.Q,10,12,14,102
Reads N90 DD quality at ring 10, node 12, module 14, block 102 into a DDE discrete tag.

DD.MODE,10,12,14,102
Reads/writes N90 DD mode at ring 10, node 12, module 14, block 102 into a DDE discrete tag.

DD.V,10,12,14,103
Reads/writes DD value at ring 10, node 12, module 14, block 103 to/from a DDE discrete tag.

DD.TIME,10,12,14,103
Returns the time expressed as seconds since 1/1/1970 of last update received from a DD value at ring 10, node 12, module 14 block 103 into a DDE integer tag.

DD.TIMETEXT,10,12,14,103
Returns the time expressed as a text message like “Mon Oct 26 08:00:00 1998”of last update received from a DD value at ring 10, node 12, module 14 block 103 into a DDE message tag.

DD.UPDATES,10,12,14,103
Returns a running count of the total updates received from a DD value at ring 10, node 12, module 14 block 103 into a DDE integer tag.

DD.TYPE,10,12,14,103
Reads the type indicator from a DD at ring 10, node 12, module 14 block 103 into a DDE integer tag. The type indicator can be used to vary the display faceplate based on the DD type indicated within the N90 block configuration.

7.5. Digital Input / Loop - DIL

The DIL item point type is used to retrieve the exception reported output from N90 Digital Output / Loop blocks (function code 45).

Name: DIL

Restrictions: This item can be utilized with all N90 interface types except the serial port module. Ring and node addresses must be set to zero when the N90 interface is a CIC.

Subscripts	Purpose	DDE Type	Access
.ALM	alarm indicator (0 - no, 1 - yes)	discrete	read
.ALMSTATE	alarm state: 0 - none, 1 - low, 2 - high, 3 - low & high	integer	read
.Q	quality (0 - good, 1 - bad)	discrete	read
.TIME	time (expressed as seconds since 1/1/1970) of last point value update	integer	read
.TIMETEXT	time (month, day, year, hour, minute, second) of last point value update	message	read
.UPDATES	running count of updates received for this point	integer	read
.V	value	discrete	read
.VTEXT	value as text (defined at point creation)	message	read

Example usage's are:

DIL.V,1,1,2,50
Reads N90 DO/L value at ring 1, node 1, module 2, block 50 into a DDE discrete tag.

DIL.V,2,60,20,6000
Reads N90 DO/L value at ring 1, node 2, module 3, block 100 into a DDE discrete tag.

DIL.VTEXT,2,60,20,6000,Off,On
Reads N90 DO/L value at ring 1, node 2, module 3, block 100 into a DDE message tag. When the N90 DIL value is zero, the message “Off” will be returned, when one “On” will be returned.

DIL.Q,2,60,20,6000
Reads DO/L quality at ring 2, node 60, module 20 block 6000 into a DDE discrete tag.

DIL.ALMSTATE,2,60,20,6000
Reads DO/L alarm state definition at ring 2, node 60, module 20 block 6000 into a DDE integer tag.

DIL.ALM,2,60,20,6000

Reads DO/L alarm indicator at ring 2, node 60, module 20 block 6000 into a DDE discrete tag. Note that the DO/L function block (F.C. 45) can be configured to set its alarm indication for either logic level (zero or one) or never alarm.

DIL.TIME,2,60,20,6000
Returns the time expressed as seconds since 1/1/1970 of last update received from a DO/L value at ring 2, node 60, module 20 block 6000 into a DDE integer tag.

DIL.TIMETEXT,2,60,20,6000
Returns the time expressed as a text message like “Mon Oct 26 08:00:00 1998”of last update received from a DO/L value at ring 2, node 60, module 20 block 6000 into a DDE message tag.

DIL.UPDATES,2,60,20,6000
Returns a running count of the total updates received from a DO/L value at ring 2, node 60, module 20 block 6000 into a DDE integer tag.

7.6. Digital Output / Loop - DOL

The DOL item point type is used to send an exception reported output generated by the MMI into N90 Digital Input / Loop blocks (function code 42) and Digital Input / Infinet blocks (function code 122). The only supported value subscript is .V which is a DDE discrete read/write type. However, the N90DDE server will generate the appropriate quality, and alarm status based on the value being generated by the MMI and definition of the tag item. Since this item exports data to N90, it follows a different naming convention when defining it within an MMI tag.

Restrictions: This item can be utilized with all N90 interface types except serial port module (SPM) and computer interface command series (CIC).

Cautions: Make sure the block number field is unique with respect to the other export item definitions and is defined within the range of 1 to the maximum number of allowed outputs. This maximum is defined within the server topic selected for the MMI DDE tag. N90 receives data from this item at the ring and node address assigned to the N90 interface, module two and defined block number.

Name: DOL.V, Block, AlarmState, Initial State

Item field	Purpose
DOL	item point name
.V	data subscript is read/write DDE discrete
Block	block number at which value is to be established within N90 interface device. Its range is 1 to max output defined by topic definition.
AlarmState	state of value which is considered in alarm, where: 0 - alarm when zero 1 - alarm when one 2 - alarming disabled
Init Val	initial value of output when first established in N90 interface

Example usage's are:

DOL.V,1,0,0
Writes a discrete value to N90 interface block 1 which will be marked in alarm when the value is zero. Its initial value will be set to zero.

DOL.V,95,1,0
Writes a discrete value to N90 interface block 95 which will be marked in alarm when the value is one. Its initial value will be set to zero.

DOL.V,97,2,1
Writes a discrete value to N90 interface block 97 which will never be marked in alarm regardless of its value. Its initial value will be set to one.

7.7. Module Status - MODSTAT

The MODSTAT item point type is used to retrieve module status summary information. Definition of this item should exclude the block number field within the item name definition. This item is designed to work with all node and module types. The messages for module state and type are defined in the N90DATA.INI file stored in the windows directory. It may be edited to customize message generation. The [MODULE STATE] section defines the messages for module state. The [MODULE TYPE] section defines the messages for module type. This section contains the Bailey defined module type code and associated module name. You may edit the names or add new type codes and associated names that correspond to new Bailey modules introduced after any given revision to N90DDE server. Note that by definition all MMI message tags must have a total length that does not exceed 132 characters.

Name: MODSTAT

Restrictions: This item can be utilized with all N90 interface types except the serial port module. Ring and node addresses must be set to zero when the N90 interface is a CIC.

Subscripts	Purpose	DDE Type	Access
.BYTE1	value of module status byte number 1	integer	read
.BYTE2	value of module status byte number 2	integer	read
.BYTE3	value of module status byte number 3	integer	read
.BYTE4	value of module status byte number 4	integer	read
.BYTE5	value of module status byte number 5	integer	read
.ERRORS	error summary (0 - none, 1 - one or more module runtime errors exists)	discrete	read
.Q	quality (0 - good, 1 - bad)	discrete	read
.STATE	current module operating state (execute, configure, error)	message	read
.TIME	time (expressed as seconds since 1/1/1970) of last point value update	integer	read
.TIMETEXT	time (month, day, year, hour, minute, second) of last point value update	message	read
.UPDATES	running count of updates received for this point	integer	read
.TYPE	module type, ex AMM or MFP	message	read

Example usage's are:

MODSTAT.BYTE3,2,3,4
Reads the 3rd module status byte for the module at N90 address; ring 2, node 3, module 4, to a DDE integer tag.

MODSTAT.BYTE4,2,3,4
Reads the 4th module status byte for the module at N90 address; ring 2, node 3, module 4, to a DDE integer tag.

MODSTAT.BYTE5,2,3,4

Reads the 5th module status byte for the module at N90 address; ring 2, node 3, module 4, to a DDE integer tag.

MODSTAT.ERRORS,2,3,4
Reads the error summary status bit for the module at N90 address; ring 2, node 3, module 4, to a DDE discrete tag.

MODSTAT.STATE,2,3,4
Reads the module operating state for the module at N90 address; ring 2, node 3, module 4, to a DDE message tag.

MODSTAT.TYPE,2,3,4
Reads the module type for the module at N90 address; ring 2, node 3, module 4, to a DDE message tag.

7.8. Multi-State Device Driver - MSDD

The MSDD item point type is used to retrieve and control the exception reported output from N90 Multi-State Device Driver blocks (function code 129).

Name: MSDD

Restrictions: This item can be utilized with all N90 interface types except the serial port module. Ring and node addresses must be set to zero when the N90 interface is a CIC.

Subscripts	Purpose	DDE Type	Access
.ALM	alarm (0 - none, 1 - alarm)	discrete	read
.F1	feedback #1	discrete	read
.F2	feedback #2	discrete	read
.F3	feedback #3	discrete	read
.F4	feedback #4	discrete	read
.GS	good state: 0 - default 1 - state1 2 - state2 3 - state3	integer	read
.GSTEXT	good state as text (defined at point creation)	message	read
.MODE	mode: 0 - manual 1 - auto	discrete	read/write
.OVRC	override control (0 - no, 1 - yes)	discrete	read
.OVRS	override status (0 - no, 1 - yes)	discrete	read
.Q	quality (0 - good, 1 - bad)	discrete	read
.RS	requested state (same as .GS)	integer	read/write
.RSTEXT	requested state as text (defined at point creation)	message	read/write
.TIME	time (expressed as seconds since 1/1/1970) of last point value update	integer	read
.TIMETEXT	time (month, day, year, hour, minute, second) of last point value update	message	read
.TYPE	faceplate type code	integer	read
.UPDATES	running count of updates received for this point	integer	read
.TRV	travel indicator (0 - no, 1 - yes)	discrete	read
.V	value (1st block output)	discrete	read
.VTEXT	value as text (defined at point creation)	message	read

Example usage's are:

MSDD.RS,2,3,4,110
Reads/writes N90 MSDD requested state at ring 2, node 3, module 4, block 110 to/from a DDE integer tag.

MSDD.RSTEXT,2,3,4,110,default,state1,state2,state3
Reads/writes N90 MSDD requested state at ring 2, node 3, module 4, block 110 to/from a DDE message tag. The N90 MSDD requested states of zero, one, two and three will return the respective messages “default”, “state1”, “state2” and

“state3”. These state messages can also be written to request a new MSDD state or text messages of “0”, “1”, “2” or “3”.

MSDD.GS,2,3,4,110
Reads N90 MSDD good state at ring 2, node 3, module 4, block 110 into a DDE integer tag.

MSDD.GSTEXT,2,3,4,110,Off,Slow,Medium,Fast
Reads N90 MSDD good state at ring 2, node 3, module 4, block 110 into a DDE message tag. The N90 MSDD good states of zero, one, two and three will return the respective messages “Off”, “Slow”, “Medium” and “Fast”.

MSDD.TRV,2,3,4,110
Reads N90 MSDD travel indicator at ring 2, node 3, module 4, block 110 into a DDE discrete tag.

MSDD.V,2,3,4,110
Reads N90 MSDD first output value at ring 2, node 3, module 4, block 110 into a DDE discrete tag.

MSDD.TIME, 2,3,4,110
Returns the time expressed as seconds since 1/1/1970 of last update received from a MSDD value at ring 2, node 3, module 4 block 110 into a DDE integer tag.

MSDD.TIMETEXT, 2,3,4,110
Returns the time expressed as a text message like “Mon Oct 26 08:00:00 1998”of last update received from a MSDD value at ring 2, node 3, module 4 block 110 into a DDE message tag.

MSDD.UPDATES, 2,3,4,110
Returns a running count of the total updates received from a MSDD value at ring 2, node 3, module 4 block 110 into a DDE integer tag.

MSDD.TYPE, 2,3,4,110
Reads the type indicator from a MSDD at ring 10, node 12, module 14 block 103 into a DDE integer tag. The type indicator can be used to vary the display faceplate based on the MSDD type indicated within the N90 block configuration.

7.9. Poll Analog Output - POLLAO

The POLLAO item point type is used to retrieve polled output values from any N90 function block that generates analog (real) outputs. Its primary usage is intended for node level data acquisition when the N90 interface type is a N90 serial port module.

Name: POLLAO

Cautions: Usage of this item with the N90 interfaces other then serial port modules should be limited to those few N90 block output values not currently being exception reported. Data is obtained for this item by polling. Since N90 is optimized for exception reporting, polling for N90 outputs will slightly lower the data throughput of other exception reported values being collected by the server. However, it will not interfere with exception report activity not directly associated with the server. Usage of this item with the N90 interfaces that sit on the N90 communication loop should only add a minimal overhead to the PCU nodes from which data is being polled. Ring and node addresses must be set to zero when the N90 interface is a CIC.

Subscripts	Purpose	DDE Type	Access
.HIALM	high alarm indicator (0 - no, 1 - yes)	discrete	read
.LOALM	low alarm indicator (0 - no, 1 - yes)	discrete	read
.Q	quality (0 - good, 1 - bad)	discrete	read
.TIME	time (expressed as seconds since 1/1/1970) of last point value update	integer	read
.TIMETEXT	time (month, day, year, hour, minute, second) of last point value update	message	read
.UPDATES	running count of updates received for this point	integer	read
.V	value	real	read

Example usage's are:

POLLAO.V,1,2,3,50
Reads N90 function block output value at ring 1, node 2, module 3, block 50 into a DDE real tag.

POLLAO.Q,1,2,3,50
Reads N90 function block output quality at ring 1, node 2, module 3, block 50 into a DDE discrete tag.

POLLAO.V,2,249,31,9998
Reads N90 function block output value at ring 2, node 249, module 31 block 9998 into a DDE real tag.

POLLAO.HIALM, 2,249,31,9998
Reads N90 function block output high alarm indicator at ring 2, node 249 module 31, block 9998 into a DDE discrete tag.

POLLAO.LOALM, 2,249,31,9998
Reads N90 function block output low alarm indicator at ring 2, node 249, module 31, block 9998 into a DDE discrete tag.

POLLAO.TIME,2,249,31,9998
Returns the time expressed as seconds since 1/1/1970 of last update received from block output at ring 2, node 249, module 31 block 9998 into a DDE integer tag.

POLLAO.TIMETEXT,2,249,31,9998
Returns the time expressed as a text message like “Mon Oct 26 08:00:00 1998”of last update received from block output at ring 2, node 249, module 31 block 9998 into a DDE message tag.

POLLAO.UPDATES,2,249,31,9998
Returns a running count of the total updates received from an block output at ring 2, node 249, module 31 block 9998 into a DDE integer tag.

7.10. Poll Digital Output - POLLDO

The POLLDO item point type is used to retrieve polled output values from any N90 function block that generates digital (discrete) outputs. Its primary usage is intended for node level data acquisition when the N90 interface type is a N90 serial port module.

Name: POLLDO

Cautions: Usage of this item with the N90 interfaces other then serial port modules should be limited to those few N90 block output values not currently being exception reported. Data is obtained for this item by polling. Since N90 is optimized for exception reporting, polling for N90 outputs will slightly lower the data throughput of other exception reported values being collected by the server. However, it will not interfere with exception report activity not directly associated with the server. Usage of this item with the N90 interfaces that sit on the N90 communication loop should only add a minimal overhead to the PCU nodes from which data is being polled. Ring and node addresses must be set to zero when the N90 interface is a CIC.

Subscripts	Purpose	DDE Type	Access
.ALM	alarm (0 - none, 1 - alarm)	discrete	read
.Q	quality (0 - good, 1 - bad)	discrete	read
.TIME	time (expressed as seconds since 1/1/1970) of last point value update	integer	read
.TIMETEXT	time (month, day, year, hour, minute, second) of last point value update	message	read
.UPDATES	running count of updates received for this point	integer	read
.V	value	discrete	read
.VTEXT	value as text (defined at point creation)	message	read

Example usage's are:

POLLDO.V,1,2,3,51
Reads N90 function block output value at ring 1, node 2, module 3, block 51 into a DDE discrete tag.

POLLDO.Q,1,2,3,51
Reads N90 function block output quality at ring 1, node 2, module 3, block 51 into a DDE discrete tag.

POLLDO.V,2,249,31,9996
Reads N90 function block output value at ring 2, node 249, module 31 block 9996 into a DDE discrete tag.

POLLDO.VTEXT, 2,249,31,9996,Ok,Tripped
Reads N90 function block output value at ring 2, node 249, module 31, block 9996 into a DDE message tag. When the N90 block output value is zero, the message “Ok” will be returned, when one “Tripped” will be returned.

POLLDO.TIME,2,249,31,9996
Returns the time expressed as seconds since 1/1/1970 of last update received from block output at ring 2, node 249, module 31 block 9996 into a DDE integer tag.

POLLDO.TIMETEXT,2,249,31,9996
Returns the time expressed as a text message like “Mon Oct 26 08:00:00 1998”of last update received from block output at ring 2, node 249, module 31 block 9996 into a DDE message tag.

POLLDO.UPDATES,2,249,31,9996
Returns a running count of the total updates received from an block output at ring 2, node 249, module 31 block 9996 into a DDE integer tag.

7.11. Remote Control Memory - RCM

The RCM item point type is used to retrieve and control the exception reported output from N90 Remote Control Memory blocks (function code 62).

Name: RCM

Restrictions: This item can be utilized with all N90 interface types except the serial port module. Ring and node addresses must be set to zero when the N90 interface is a CIC.

Subscripts	Purpose	DDE Type	Access
.ALM	alarm (0 - none, 1 - alarm)	discrete	read
.FB	feedback (0 - off, 1 - on)	discrete	read
.Q	quality (0 - good, 1 - bad)	discrete	read
.SETPERM	set permissive (0 - no, 1 - yes)	discrete	read
.TIME	time (expressed as seconds since 1/1/1970) of last point value update	integer	read
.TIMETEXT	time (month, day, year, hour, minute, second) of last point value update	message	read
.TYPE	faceplate type code	integer	read
.UPDATES	running count of updates received for this point	integer	read
.V	value	discrete	read/write
.VTEXT	value as text (defined at point creation)	message	read/write

Example usage's are:

RCM.ALM,5,6,7,80
Reads N90 RCM alarm bit at ring 5, node 6, module 7, block 80 into a DDE discrete tag.

RCM.FB,5,6,7,80
Reads N90 RCM feedback signal at ring 5, node 6, module 7, block 80 into a DDE discrete tag.

RCM.V,5,6,7,80
Reads/writes N90 RCM value at ring 5, node 6, module 7, block 80 to/from a DDE discrete tag.

RCM.VTEXT,5,6,7,80,Reset,Set
Reads/writes N90 RCM value at ring 5, node 6, module 7, block 80 to/from a DDE message tag. When the N90 RCM value is zero, the message “Reset” will be returned, when one “Set” will be returned. These messages can also be written to reset and set the N90 RCM. Also text messages of “0” or “1” can be written to reset and set the N90 RCM.

RCM.Q,5,6,7,80
Reads N90 RCM quality at ring 5, node 6, module 7, block 80 into a DDE discrete tag.

RCM.SETPERM,4,5,3,50
Reads RCM set permissive signal at ring 4, node 5, module 3, block 50 into a DDE discrete tag.

RCM.V,4,100,30,1001
Reads/writes RCM value at ring 4, node 100, module 30, block 1001 to/from a DDE discrete tag.

RCM.TIME,4,100,30,1001
Returns the time expressed as seconds since 1/1/1970 of last update received from a RCM value at ring 4, node 100, module 30, block 1001 into a DDE integer tag.

RCM.TIMETEXT,4,100,30,1001
Returns the time expressed as a text message like “Mon Oct 26 08:00:00 1998”of last update received from a RCM value at ring 4, node 100, module 30, block 1001 into a DDE message tag.

RCM.UPDATES,4,100,30,1001
Returns a running count of the total updates received from a RCM value at ring 4, node 100, module 30, block 1001 into a DDE integer tag.

RCM.TYPE,4,100,30,1001
Reads the type indicator from a RCM at ring 4, node 100, module 30, block 1001 into a DDE integer tag. The type indicator can be used to vary the display faceplate based on the RCM type indicated within the N90 block configuration.

7.12. Remote Motor Control - RMC

The RMC item point type is used to retrieve and control the exception reported output from N90 Remote Motor Control blocks (function code 136).

Name: RMC

Restrictions: This item can be utilized with all N90 interface types except the serial port module. Ring and node addresses must be set to zero when the N90 interface is a CIC.

Subscripts	Purpose	DDE Type	Access
.ALM	alarm (0 - none, 1 - alarm)	discrete	read
.ERRCODE	error code for fault or bad start condition: 0 - no error, 1 - stop input, 2 - interlock #1 input 3 - interlock #2 input 4 - interlock #3 input 5 - interlock #4 input 6 - feedback #1 input is 0 7 - feedback #2 input is 0 8 - feedback #1 input is 1 9 - feedback #1 input is 1	integer	read
.F1	feedback #1 (0 - off, 1 - on)	discrete	read
.F2	feedback #2 (0 - off, 1 - on)	discrete	read
.FLT	a fault has occurred (0 - no, 1 - yes)	discrete	read
.FLTACK	set to 1 to acknowledge a fault or bad start and module will reset to 0 after acknowledgment is accepted	discrete	read/write
.PERM1	start permissive #1 (0 - no, 1 - yes)	discrete	read
.PERM2	start permissive #2 (0 - no, 1 - yes)	discrete	read
.SBAD	bad start (0 - no, 1 - yes)	discrete	read
.SHOLD	status on hold (0 - no, 1 - yes)	discrete	read
.Q	quality (0 - good, 1 - bad)	discrete	read
.TIME	time (expressed as seconds since 1/1/1970) of last point value update	integer	read
.TIMETEXT	time (month, day, year, hour, minute, second) of last point value update	message	read
.TYPE	faceplate type code	integer	read
.UPDATES	running count of updates received for this point	integer	read
.V	value (0 - stop, 1 - start)	discrete	read/write
.VTEXT	value as text (defined at point creation)	message	read/write

Example usage's are:

RMC.ALM,5,6,7,80
Reads N90 RMC alarm bit at ring 5, node 6, module 7, block 80 into a DDE discrete tag.

RMC.ERRCODE,5,6,7,80
Reads N90 RMC error code at ring 5, node 6, module 7, block 80 into a DDE discrete tag.

RMC.F1,5,6,7,80
Reads N90 RMC feedback #1 signal at ring 5, node 6, module 7, block 80 into a DDE discrete tag.

RMC.F2,5,6,7,80
Reads N90 RMC feedback #2 signal at ring 5, node 6, module 7, block 80 into a DDE discrete tag.

RMC.V,5,6,7,80
Reads/writes N90 RMC value at ring 5, node 6, module 7, block 80 to/from a DDE discrete tag.

RMC.VTEXT,5,6,7,80,Stop,Start
Reads/writes N90 RMC value at ring 5, node 6, module 7, block 80 to/from a DDE message tag. When the N90 RMC value is zero, the message “Stop” will be returned, when one “Start” will be returned. These messages can also be written to reset and set the N90 RMC. Also text messages of “0” or “1” can be written to reset and set the N90 RMC.

RMC.Q,5,6,7,80
Reads N90 RCM quality at ring 5, node 6, module 7, block 80 into a DDE discrete tag.

RMC.V,4,100,30,1001
Reads/writes RCM value at ring 4, node 100, module 30, block 1001 to/from a DDE discrete tag.

RMC.TYPE,4,100,30,1001
Reads the type indicator from a RMC at ring 4, node 100, module 30, block 1001 into a DDE integer tag. The type indicator can be used to vary the display faceplate based on the RMC type indicated within the N90 block configuration.

7.13. Remote Manual Set Constant - RMSC

The RMSC item point type is used to retrieve and control the exception reported output from N90 Remote Manual Set Constant blocks (function code 68).

Name: RMSC

Restrictions: This item cannot be utilized when the N90 interface type is a serial port module. Ring and node addresses must be set to zero when the N90 interface is a CIC.

Subscripts	Purpose	DDE Type	Access
.EUNUM	engineering unit number	integer	read
.EUTEXT	engineering unit text message	message	read
.HILIM	high limit of settable value	real	read
.LOLIM	low limit of settable value	real	read
.Q	quality (0 - good, 1 - bad)	discrete	read
.SPTRACK	set point tracking: (0 - no, 1 - yes)	discrete	read
.TIME	time (expressed as seconds since 1/1/1970) of last point value update	integer	read
.TIMETEXT	time (month, day, year, hour, minute, second) of last point value update	message	read
.UPDATES	running count of updates received for this point	integer	read
.V	value	real	read/write

Example usage's are:

RMSC.V,2,1,4,75
Reads/writes N90 RMSC value at ring 2, node 1, module 4, block 75 to/from a DDE real tag.

RMSC.Q,2,1,4,75
Reads N90 RMSC quality at ring 2, node 1, module 4, block 75 into a DDE discrete tag.

RMSC.SPTRACK,2,1,4,75
Reads N90 RMSC set point track flag at ring 2, node 1, module 4, block 75 into a DDE discrete tag.

RMSC.V,2,1,4,75
Reads/writes RMSC value at ring 2, node 1, module 4, block 75 to/from a DDE real tag.

RMSC.HILIM,2,1,4,75
Reads N90 RMSC high limit at ring 2, node 1, module 4, block 75 into a DDE real tag.

RMSC.LOLIM,2,1,4,75
Reads N90 RMSC low alarm limit at ring 2, node 1, module 4, block 75 into a DDE real tag.

RMSC.EUNUM,2,1,4,75
Reads engineering unit number assigned to N90 RMSC at ring 2, node 1, module 4, block 75 into a DDE integer tag.

RMSC.EUTEXT,2,1,4,75
Reads engineering unit number assigned to N90 RMSC ring 2, node 1, module 4, block 75 and converts it to a text message (defined by user within N90DDE engineering units setup dialog) written to a DDE message tag.

7.14. Sequence of Events - SOE

The SOE item point type is used to retrieve sequence of event recorder logs from the exception reported N90 Sequence of Events Log blocks (function code 99).

Name: SOE

Restrictions: This item cannot be utilized when the N90 interface type is a serial port module (SPM) or command interface controller (CIC).

Subscripts	Purpose	DDE Type	Access
.COUNT	Total points logged to the currently selected SOE capture file	integer	read
.EVENT	Signals that a SOE event has been captured (0-armed and waiting, 1-captured and locked)	discrete	read/write
.FILE	Indicates / selects the current file name for the captured SOE data	message	read/write
.NEXTFILE	Flag used to sequence through the most recent to oldest SOE capture files	discrete	read/write
.NEXTREC	Flag used to sequence through the records of the currently selected SOE capture file	discrete	read/write
.OFLOW	The SOE data overflowed the N90 SOELOG block buffer. (0-ok, 1-overflow occurred)	discrete	read
PNTALM	SOE point alarm (0-no, 1-yes)	discrete	read
.PNTDATE	SOE point date when capture occurred	message	read
.PNTNAME	SOE point name	message	read
.PNTNUM	SOE point number	integer	read
.PNTQ	SOE point quality (0-good, 1-bad, inactive)	discrete	read
.PNTSCAN	SOE point scan flag (0-yes, 1-deleted from scan)	discrete	read
.PNTTIME	SOE point time when capture occurred	message	read
.PNTTYPE	SOE point type (“Standard”, “Summary”, “Pre-fault”, “Post-fault” and “Snap-shot”)	message	read
.PNTV	SOE point value at capture (0 or 1)	discrete	read
.Q	Quality of this point, Bailey controller communication port and RA-3800 SER status. A value of 0 indicates ok otherwise the point address is wrong, the Bailey controller is not communicating with the RA-3800 SER or it has problems.	discrete	read
.RECORD	Indicates / selects a record for the currently selected capture file	integer	read/write
.TIME	time (expressed as seconds since 1/1/1970) of last point value update	integer	read
.TIMETEXT	time (month, day, year, hour, minute, second) of last point value update	message	read
.TYPEBLK	Type of SOE log block (“Not determined”, “Standard”, “Summary”, “Pre-fault”, “Post-fault” and “Snap-shot”	message	read
.UPDATES	running count of updates received for this point	integer	read
.V	Value indicating whether or not the SOELOG block has data present. (0-empty, 1-data present)	discrete	read/write

Note the .TYPEBLK must be configured and always active for the other items to function correctly.

Example usage's are:

SOE.COUNT,1,2,3,50
Reads the total count of SOE points (records) stored in the SOE capture file indicated by the associated SOE.FILE item for the N90 SOELOG block at ring 1, node 2, module 3, block 50.

SOE.EVENT,1,2,3,50
Reads/Writes a flag associated with the N90 SOELOG block at ring 1, node 2, module 3, block 50 into a DDE discrete tag which is used to indicate a SOE event has been captured or request reading of the most recent or previous capture event file. A value of zero flags the armed and waiting condition. A value of one indicates SOE data has been captured and stored in the file indicated by the associated SOE.FILE item. After the SOE data is retrieved from that file the client must set this DDE tag to zero to re-arm for the next capture. When this tag is set the user may request the next oldest capture file by setting the associated SOE.NEXTFILE tag or writing the name of the desired capture file to the SOE.FILE tag.

SOE.FILE,1,2,3,50
Reads/Writes the name of a file containing SOE capture data for the N90 SOELOG block at ring 1, node 2, module 3, block 50 from/to a DDE message tag. When a SOE event occurs, N90DDE gathers the data from the SOELOG block and stores it into the SOE capture file indicated by this tag. The associated SOE.EVENT DDE tag is then set to signal the client that a SOE event has occurred and the data is ready to be retrieved from the indicated file.

By definition file names are the date of capture followed by a two digit (01 - 99) sequence code. The capture file types are “STD”, “SUM”, “PRE”, “POS” or “SNP” which corresponds to standard, summary, pre-fault, post-fault and snapshot type logs.

Consider the following examples:

1.) 03089702.STD
2.) 02289701.SUM
3.) 03089701.PRE
4.) 03089701.POS
5.) 12259699.SNP

Example 1 is the 2nd standard SOE log captured on March 8, 1997. Example 2 is the 1st summary SOE log captured on February 28, 1997. Example 3 is the 1st pre-fault SOE log captured on March 8, 1997 . Example 4 is the 1st post-fault SOE log captured on March 8, 1997. And finally example 5 is the 99th

snapshot SOE log captured on December 25th, 1996. Note that this naming convention supports up to 99 capture files (xxxxxx01.STD – xxxxxx99.STD) for each given log type to be recorded per day. After generation of the 99th log file (xxxxxx99.STD) it will wrap back around to the beginning (xxxxxx01.STD). The capture files are stored in the sub-directory defined by the Sequence of Events configuration dialog discussed earlier in this manual.

The data in the log file is stored in a space separated ASCII format. The first line is a count of the number of SOE records contained by the file. Each subsequent line is a SOE record containing the point number, date string, time string its quality, alarm state, scan state, value, point type string and point name string. Consider the following example data stored in the file “02019701.PRE”:

```
4
1  2/01/1997  23:23:50.120 0 0 0 1 Pre-fault  Pump_A
2  2/01/1997  23:23:50.125 0 0 1 0 Pre-fault  Pump_B
6  2/01/1997  23:23:50.203 0 1 0 1 Pre-fault  Compressor_1
9  2/01/1997  23:23:50.987 1 0 0 0 Pre-fault  Compressor_2
```

The log is the first one to occur on February 1, 1997 for pre-faulted data points. It contains four SOE records as indicated by the first line.

The first record is point #1 which was logged by the SER on February 1, 1997 at 23:23:50.120. It has good quality, its not in alarm, has not been deleted from the scan and has a value of one. The point type is pre-fault and it called “Pump_A”.

The second record is point #2 which was logged by the SER on February 1, 1997 at 23:23:50.125. It has good quality, its not in alarm, has been deleted from the scan and has a value of zero. The point type is pre-fault and it called “Pump_B”.

The third record is point #6 which was logged by the SER on February 1, 1997 at 23:23:50.203. It has good quality, its in alarm, has not been deleted from the scan and has a value of one. The point type is pre-fault and it called “Compressor_1”.

The fourth record is point #9 which was logged by the SER on February 1, 1997 at 23:23:50.987. It has bad quality, its not in alarm, has not been deleted from the scan and has a value of zero. The point type is pre-fault and it called “Compressor_2”.

SOE.NEXTFILE,1,4,5,100
Read/Write DDE discrete tag used to request the next most recent capture file of a particular SOE type to be displayed in the associated SOE.FILE and SOE.PNT_* tags associated with the N90 SOELOG block at ring 1, node 4, module 5, block 100. The associated SOE.EVENT tag must also be set (1) for the SOE.NEXTFILE tag to be utilized. Under this condition, SOE.NEXTFILE is used to sequence through the most recent to oldest capture file for the associated SOE type. For example after setting SOE.EVENT, each time SOE.NEXTFILE is set the next oldest capture file will be displayed in SOE.FILE and the associated SOE.PNT_* tags. SOE.NEXTFLE will be reset back to False (0) to acknowledge that the next capture file has been selected. When SOE.FILE becomes a null string all capture files have been selected. The next time SOE.NEXTFILE is set the cycle starts over with the most recent capture file being selected again.

SOE.NEXTREC,1,4,5,100
Read/Write DDE discrete tag used to request the next record of a particular SOE point from a currently selected SOE capture file associated with the N90 SOELOG block at ring 1, node 4, module 5, block 100. The associated SOE.EVENT tag must be set (1) and the SOE.FILE tag must contain a valid file name for the SOE capture file of interest for the SOE.NEXTREC tag to be utilized. Under this condition, SOE.NEXTREC is used to sequence through the total number of records in the file as indicated by the SOE.COUNT tag. For example after a capture has occurred, N90DDE will set the associated SOE.EVENT tag, indicate the SOE capture file in SOE.FILE, write the total number of items captured in that file to the SOE.COUNT tag and display the first record data in the SOE.PNT_* tags. The next SOE point record can be requested by setting the SOE.NEXTREC tag to True (1). The next record will be read from the capture file and displayed in the individual SOE.PNT_* tags. SOE.NEXTREC will be set back to False (0) to acknowledge that the read request has been completed. To cycle through all records in a SOE capture file, the SOE.NEXTREC should be set True for the number of times indicated by SOE.COUNT - 1. Note that SOE.RECORD may also be used to indicate and select random records.

SOE.OFLOW,1,6,7,200
Reads N90 Sequence of Events Log block overflow indicator at ring 1, node 6, module 7, block 200 into a DDE discrete tag. A value of zero indicates OK. A value of one indicates an overflow occurred. The N90 SOELOG block buffer size must be increased to clear this condition.

SOE.PNTALM,1,4,5,100
Reads a SOE point alarm flag from a currently selected SOE log file associated with the N90 SOELOG block at ring 1, node 4, module 5, block 100 into a DDE discrete tag. A value of zero indicates no alarm. A value of one indicates an alarm condition. The associated SOE.NEXTREC tag is used to sequence through the various SOE point alarms within a selected SOE log file.

SOE.PNTDATE,1,4,5,100
Reads a SOE point date string from a currently selected SOE log file associated with the N90 SOELOG block at ring 1, node 4, module 5, block 100 into a DDE message tag. This string contains the date when the point was logged by the Bailey SER device. Its format is like “12/02/1997”. The associated SOE.NEXTREC tag is used to sequence through the various SOE point log date strings within a selected SOE log file.

SOE.PNTNAME,1,4,5,100
Reads a SOE point name from a currently selected SOE log file associated with the N90 SOELOG block at ring 1, node 4, module 5, block 100 into a DDE message tag. The SOE point names must be pre-configured using the N90DDE Configure/SOE menu selection. Otherwise, defaulted names of “POINT_1”, “POINT_2”, etc will be provided. The associated SOE.NEXTREC tag is used to sequence through the various SOE point names within a selected SOE log file.

SOE.PNTNUM,1,4,5,100
Reads a SOE point number from a currently selected SOE log file associated with the N90 SOELOG block at ring 1, node 4, module 5, block 100 into a DDE integer tag. The SOE point number is that which has been configured in the Bailey SER device. The associated SOE.NEXTREC tag is used to sequence through the various SOE point numbers within a selected SOE log file.

SOE.PNTQ,1,4,5,100
Reads a SOE point quality flag from a currently selected SOE log file associated with the N90 SOELOG block at ring 1, node 4, module 5, block 100 into a DDE discrete tag. A value of zero indicates good quality. A value of one indicates bad, inactive quality. The associated SOE.NEXTREC tag is used to sequence through the various SOE point qualities within a selected SOE log file.

SOE.PNTSCAN,1,4,5,100
Reads a SOE point scan indication from a currently selected SOE log file associated with the N90 SOELOG block at ring 1, node 4, module 5, block 100 into a DDE discrete tag. A value of zero indicates point is on scan. A value of one indicates point is deleted from the scan. The associated SOE.NEXTREC tag is used to sequence through the various SOE point scans within a selected SOE log file.

SOE.PNTTIME,1,4,5,100
Reads a SOE point time string from a currently selected SOE log file associated with the N90 SOELOG block at ring 1, node 4, module 5, block 100 into a DDE message tag. This string contains the time when the point was logged by the Bailey SER device. Its format is like “23:24:25.097”. The associated SOE.NEXTREC tag is used to sequence through the various SOE point log date strings within a selected SOE log file.

SOE.PNTTYPE,1,4,5,100
Reads a SOE point type string from a currently selected SOE log file associated with the N90 SOELOG block at ring 1, node 4, module 5, block 100 into a DDE message tag. The type strings can be “Standard”, “Summary”, “Pre-fault”, “Post-fault” or “Snapshot”. The associated SOE.NEXTREC tag is used to sequence through the various SOE point types within a selected SOE log file.

SOE.PNTV,1,4,5,100
Reads a SOE point value (its value during SER capture) from a currently selected SOE log file associated with the N90 SOELOG block at ring 1, node 4, module 5, block 100 into a DDE discrete tag. The associated SOE.NEXTREC tag is used to sequence through the various SOE point states within a selected SOE log file.

SOE.Q,1,6,7,200
Reads N90 Sequence of Events Log block quality at ring 1, node 6, module 7, block 200 into a DDE discrete tag.

SOE.RECORD,1,6,7,200
Reads/Writes record number of the currently selected capture file for the N90 Sequence of Events Log block at ring 1, node 6, module 7, block 200 into / from a DDE integer tag.

SOE.TYPEBLK,2,20,25,350
Reads the type of SOE log configured for the N90 SOELOG block at ring 2, node 20, module 25, block 350 into a DDE message tag. The type strings can be “Not determined”, “Standard”, “Summary”, “Pre-fault”, “Post-fault” and “Snapshot”. N90DDE determines the type by reading the SOE function block and examining the S1 setting. It is ***important to note*** this item **must** be configured, always active and that If the block is not a F.C. 99 the value of this tag remains at “Not determined” and the other associated tags like SOE.FILE, SOE.NEXTREC, SOE.NEXTFILE, SOE.PNT* are disabled.

SOE.V,2,20,25,350
Reads a flag indicating whether or not the N90 SOELOG block at ring 2, node 20, module 25, block 350 has SOE data present and a capture is in progress into a DDE discrete tag. A value of zero indicates it is empty. A value of one indicates data is present and a capture is occurring. If the associated SOE.TYPE indicates a summary SOE log writing a value of 1 to SOE.V requests the summary data to be generated and collected.

7.15. Specification - SPEC

The SPEC item point type is used to retrieve and tune (if tunable) a N90 function block specification.

Name: SPEC

Cautions: SPEC values are acquired via polling block configuration information. The default polling interval is once every 60 seconds. This interval can be adjusted using the “DDE Server Operational Parameters” dialog available under the “Configure” N90DDE Server menu selection. The algorithm used to poll for SPEC values is designed to minimize interference with the other ongoing data collection activities. Basically, one poll request per like set of blocks is issued and must be completed before progressing to the next like block set. Once the values for all like block sets have been acquired the next poll time is calculated to be the current time plus the configured SPEC background read interval. Polling is suspended until that time arrives at which time the cycle is repeated. Use care when setting up the SPEC data items. Requesting specs from non-existing rings, nodes, modules or blocks wastes communication bandwidth. Ring and node addresses must be set to zero when the N90 interface is a CIC.

Subscripts	Purpose	DDE Type	Access
.FC	block function code number	integer	read
.S1 - .S64	specification value	real	read/write*

*Note the only tunable specs can be written.

Example usage's are:

SPEC.FC,2,3,4,400
Reads N90 function code for the block located at ring 2, node 3, module 4, block 400 into a DDE integer tag. If invalid block is referenced, the value will remain at zero which is not a valid Bailey function code number.

SPEC.S1,2,3,4,400
Reads/writes N90 specification number 1 located at ring 2, node 3, module 4, block 400 into a DDE real tag. If the referenced specification is tunable it will automatically be tuned by writing to this tag.

SPEC.S6,1,2,3,40
Reads/writes N90 specification number 6 located at ring 1, node 2, module 3, block 40 into a DDE real tag. If the referenced specification is tunable it will automatically be tuned by writing to this tag.

7.16. Statistics - STATS

The STATS item point type is used to retrieve N90DDE server operational statistics. Its intent is to provide server operational data such as data exchange rates and other miscellaneous counters.

Name: STATS

Subscripts	Purpose	DDE Type	Access
.COMM	communication (0 - good, 1 - bad)	discrete	read
.ONLINE	interface is determined and on-line (0 - no, 1 - yes)	discrete	read
.POLLONLY	interface type only supports polling or N90DDE Server license is poll only (0 - no, 1 - yes)	discrete	read
.DDERATE	DDE messages per second	integer	read
.LOCK	Operator write lock flag (0 - Unlocked, 1 - locked)	discrete	read/write
.MSGRATE	interface messages per second	integer	read
.POLLRATE	interface polling messages per second	integer	read
.XRPRATE	exception reports per second exchanged with interface	integer	read
.TAGS	total DDE tags established with the server	integer	read
.SYNCRING	ring number of current N90 drop that is the current time sync master	integer	read
.SYNCNODE	node number of current N90 drop that is the current time sync master	integer	read
.THISRING	N90 interface ring number	integer	read
.THISNODE	N90 interface node number	integer	read
.TOTALDDE	total DDE values sent to client(s)	integer	read
.TOTALMSG	total messages exchanged with interface	integer	read
.TOTALPOLL	total polling messages sent to interface	integer	read
.TOTALXRP	total exception reports exchanged with interface	integer	read
.VERSION	N90DDE Server version number	real	read

Example usage:

STATS.COMM
Reads communication status of the N90DDE Server link with the N90 interface device.

STATS.ONLINE
Reads online status of the N90 interface device. Initially N90DDE Server must determine the interface type and then command it online. This DDE tag indicates when that process has been completed.

STATS.POLLONLY
Returns poll only indication for the N90 interface device or current N90DDE Server license. When one, only POLLAO and POLLDO items are valid for the current N90 interface or N90DDE Server license. This DDE item is valid after

STATS.ONLINE is one unless the N90DDE Server license is poll only in which case the value returned does not depend on the STATS.ONLINE item.

STATS.THISRING
Reads ring number of the N90 interface N90DDE Server is communicating with. This DDE tag is not valid for SPM, CIC and CIU01 interfaces.

STATS.THISNODE
Reads ring number of the N90 interface N90DDE Server is communicating with. This DDE tag is not valid for SPM and CIC interfaces.

STATS.LOCK
Read / write flag controlling whether or not writes to any DDE tag for this topic will be accepted. When set (1), all writes will be rejected. The default setting is reset (0). Note that the global write lock out found in the N90DDE “DDE Server Operational Parameters” submenu selection overrides this tag. In other words, if the global write lock out is enabled STATS.LOCK cannot be reset (0).

STATS.DDERATE
Indicates DDE messages (values) per second being sent by the server to the client(s).

STATS.MSGRATE
Indicates messages per second being exchanged between the server and the Bailey computer interface device.

STATS.POLLRATE
Indicates polling messages (result of using POLLAO and POLLDO item) per second being exchanged between the server and the Bailey computer interface device.

STATS.XRPRATE
Indicates exception reports (values) per second being exchanged between the server and Bailey computer interface device.

STATS.TAGS
Indicates running count of total DDE tags established between all DDE clients and the N90DDE server.

STATS.TOTALDDE
Indicates running count of total DDE values sent by the server to the clients(s).

STATS.TOTALMSG
Indicates running count of total messages exchanged between the server and Bailey computer interface device.

STATS.TOTALPOLL

Indicates running count of total polled values (result of using POLLAO and POLLDO items) received by the server from the Bailey computer interface device.

STATS.TOTALXRP
Indicates running count of total exception reports (values) exchanged between the server and Bailey computer interface device.

STATS.SYNCRING
Reads ring number of the current N90 Plantloop or Infinet highway drop that has assumed the time sync mastership role. This DDE tag is not valid for SPM, CIC and CIU01 interfaces.

STATS.SYNCNODE
Reads node number of the current N90 Plantloop or Infinet highway drop that has assumed the time sync mastership role. This DDE tag is not valid for SPM, CIC and CIU01 interfaces.

STATS.VERSION
Reads N90DDE Server software version number.

7.17. Station - STN

The STN item point type is used to retrieve and control the exception reported output from N90 Control Station blocks (function code 21, 22, 23, 80).

Name: STN

Restrictions: This item can be utilized with all N90 interface types except the serial port module. The .RI subscript is not supported by the NCIU01. Ring and node addresses must be set to zero when the N90 interface is a CIC.

Subscripts	Purpose	DDE Type	Access
.AOBYPASS	analog out bypassed: (0 - no, 1 - yes)	discrete	read
.CO	control output	real	read/write
.COTRACK	control output tracking: (0 - no, 1 - yes)	discrete	read
.CPU	computer level control status: (0 - failed, 1 - OK)	discrete	read/write
.DSBAD	digital station bad: (0 - no, 1 - yes)	discrete	read
.DEVALMLIM	deviation alarm limit	real	read
.EUNUM	engineering unit number	integer	read
.EUTEXT	engineering unit text message	message	read
.HIALM	high pv alarm indicator (0 - no, 1 - yes)	discrete	read
.HIALMLIM	high alarm limit	real	read/write
.HIDEV	high deviation alarm indicator (0 - no, 1 - yes)	discrete	read
.K	PID gain multiplier	real	read/write
.KDIR	PID direction 0 - reverse, error = SP - PV 1 - direct, error = PV - SP	discrete	read/write
.KD	PID derivative constant (minute)	real	read/write
.KDLAG	PID derivative lag constant (F.C. 156 only)	real	read/write
.KFC	PID function code number	integer	read
.KI	PID integral constant (1/minute)	real	read/write
.KIONLY	PID integral only on set point change (when 1)	discrete	read/write
.KHILIM	PID control output high limit	real	read/write
.KLOLIM	PID control output low limit	real	read/write
.KP	PID proportional constant	real	read/write
.KPIDBLK	PID block number associated with this STN item. The server will attempt to automatically determine this on startup but may have to be filled in by user if the attempt fails	integer	read/write
.KTUNE	Place this STN item in fast update mode which enables fast updating of STN.PV and STN.CO. Server automatically deactivates .KTUNE after user defined fast update duration expires.	discrete	read/write
.KTYPE	PID alogrithm type: (0 - classical, 1 - noninteracting, 2 - classical with external reset, 3 - manual reset noninteracting)	integer	read
.LEVEL	station control level: (0 - local, 1 - computer)	discrete	read
.LOALM	low pv alarm indicator (0 - no, 1 - yes)	discrete	read
.LOALMLIM	low alarm limit	real	read/write
.LODEV	low deviation alarm indicator (0 - no, 1 - yes)	discrete	read

.MODE	mode: 0 - local manual 3 - computer manual 1 - local auto 4 - computer auto 2 - local cascade 5 - computer cascade	integer	read/write
.MODELOCK	locked into current mode: (0 - no, 1 - yes)	discrete	read
.PV	process variable	real	read
.Q	quality (0 - good, 1 - bad)	discrete	read
.RI	ratio index	real	read/write
.SP	set point	real	read/write
,SPAN	span of process variable and setpoint range	real	read
.SPTRACK	set point tracking: (0 - no, 1 - yes)	discrete	read
.TIME	time (expressed as seconds since 1/1/1970) of last point value update	integer	read
.TIMETEXT	time (month, day, year, hour, minute, second) of last point value update	message	read
.TYPE	faceplate type code	integer	read
.TYPETEXT	faceplate type message	message	read
.UPDATES	running count of updates received for this point	integer	read
.ZEROPV	zero value of process variable range	real	read
.ZEROSP	zero value of set point range	real	read

Example usage's are:

STN.PV,1,10,20,500
Reads N90 STN process value at ring 1, node 10, module 20, block 500 into a DDE real tag.

STN.HIALMLIM,1,10,20,500
Reads N90 STN high alarm limit at ring 1, node 10, module 20, block 500 into a DDE real tag.

STN.LOALMLIM,1,10,20,500
Reads N90 STN low alarm limit at ring 1, node 10, module 20, block 500 into a DDE real tag.

STN.DEVALMLIM,1,10,20,500
Reads N90 STN deviation alarm limit at ring 1, node 10, module 20, block 500 into a DDE real tag.

STN.HIALM,1,10,20,500
Reads N90 STN high pv alarm indicator at ring 1, node 10, module 20, block 500 into a DDE discrete tag.

STN.LOALM,1,10,20,500
Reads N90 STN low pv alarm indicator at ring 1, node 10, module 20, block 500 into a DDE discrete tag.

STN.SP,1,10,20,500
Reads/writes N90 STN set point value at ring 1, node 10, module 20, block 500 to / from a DDE real tag.

STN.LODEV,1,10,20,500
Reads N90 STN low deviation alarm indicator at ring 1, node 10, module 20, block 500 into a DDE discrete tag. Indicates PV has exceeded SP by the configured engineering units limit (see Bailey F.C. 21, 22, 23 or 80 documentation).

STN.HIDEV,1,10,20,500
Reads N90 STN high deviation alarm indicator at ring 1, node 10, module 20, block 500 into a DDE discrete tag. Indicates SP has exceeded PV by the configured engineering units limit (see Bailey F.C. 21, 22, 23 or 80 documentation).

STN.CO,1,10,20,500
Reads/writes N90 STN control output value at ring 1, node 10, module 20, block 500 to/from a DDE real tag.

STN.K,1,3,4,33
Reads/writes PID gain constant associated with the N90 STN at ring 1, node 3, module 4, block 33 into a DDE real tag. When written, the associated PID gain constant is automatically tuned.

STN.KPIDBLK,1,3,4,33
Reads/writes PID block number associate with the N90 STN at ring 1, node 3, module 4, block 33 into a DDE integer tag. At startup, the server attempts to automatically determine the station associated PID block number. It checks the station S3 input for the existence of a PID of type FC 18, 19, or 156. If one is found it is written to this item. Otherwise the value is set to -1 to indicate PID block not found. This will be the case if the PID is not configured or buffered by one or more other function block connections. In this circumstance the user must fill in the associated PID block number to make the other STN tuning items functional. It is important to note that if the other STN.K* items are not functioning, check the value of this item making sure it has a valid PID block number.

STN.KTUNE,1,3,4,33
Reads/writes PID tune flag associated with the N90 STN at ring 1, node 3, module 4, block 33 into a DDE discrete tag. When set, the server updates STN.CO and STN.PV at the user defined fast update poll rate. The server will automatically reset this flag if no tune activity is detected for a period greater

than the user defined station fast update period which is defaulted to 60 minutes. Note that the STN.K* items are updated at the SPEC item polling rate. The station fast update period and polling interval can be set using the “DDE Server Operational Parameters” dialog available under the “Configure” N90DDE Server menu selection. It defaults to 60 minutes polling every 500 milliseconds. Note this item is automatically enabled when any of the STN.K* items associated with PID tuning values are written.

STN.COTRK,1,3,4,33
Reads N90 STN control output track flag at ring 1, node 3, module 4, block 33 into a DDE discrete tag.

STN.RI,1,10,20,500
Reads N90 STN ratio index value at ring 1, node 10, module 20, block 500 into a DDE real tag.

STN.MODE,1,10,20,500
Reads/writes N90 STN mode at ring 1, node 10, module 20, block 500 into a DDE integer tag. Note that after a computer level mode is written, the server automatically begins to send computer OK messages at the interval configured in the “N90DDE Server Operational Parameters” dialog (see section entitled “Defining Operational Parameters”).

STN.CPU,1,10,20,500
Reads/writes computer status flag at N90 STN ring 1, node 10, module 20, block 500 into a DDE discrete tag. Writing a value of one (1) when currently in local level control will transfer the N90 STN block to computer level control within the current mode. For example if the current mode is local auto (1) it will be transferred to computer auto (4). Writing a value of one (1) to this item when currently in computer level control causes no action to occur (the write is ignored). Writing a value of zero (0) when currently in computer level control instructs the server to stop sending computer OK messages to the STN block. After the STN block computer watchdog timer expires (see FC 80, S31) it will report computer failed to this item and drop the STN mode to the configured computer fail default mode (see FC 80, S17 or FCs 21, 22, 23, S15).

STN.LEVEL,1,1,2,30
Reads level of control (local or computer) from N90 STN at ring, 1, node 1, module 2, block 30 into a DDE discrete tag.

STN.MODELOCK,1,1,2,30
Reads N90 STN mode lock flag at ring 1, node 1, module 2, block 30 into a DDE discrete tag.

STN.Q,1,10,20,500
Reads N90 STN quality at ring 1, node 10, module 20, block 500 into a DDE discrete tag.

STN.Q,5,6,7,80
Reads N90 STN quality at ring 5, node 6, module 7, block 80 into a DDE discrete tag.

STN.PV,99,99,2,80
Reads N90 STN process value at ring 99, node 99, module 2, block 80 into a DDE real tag.

STN.PV,100,100,3,100
Reads N90 STN process value at ring 100, node 100, module 3, block 100 into a DDE real tag.

STN.EUNUM, 100,100,3,100
Reads engineering unit number assigned to N90 STN at ring 100, node 100, module 3, block 100 into a DDE integer tag.

STN.EUTEXT, 100,100,3,100
Reads engineering unit number assigned to N90 STN at ring 100, node 100, module 3, block 100 and converts it to a text message (defined by user within N90DDE engineering units setup dialog) written to a DDE message tag.

STN.TYPE,10,12,14,103
Reads the faceplate type indicator from a STN at ring 10, node 12, module 14 block 103 into a DDE integer tag. The type indicator can be used to vary the display faceplate based on the STN type indicated within the N90 block configuration.

STN.TYPETEXT,10,12,14,103
Reads the faceplate type indicator as a text message from a STN at ring 10, node 12, module 14 block 103 into a DDE integer tag.

7.18. Text Selector - TEXT

The TEXT item point type is used to retrieve the exception reported output from N90 Text Selector blocks (function code 151). This data is used to allow controller logic to convey message numbers and attributes to console displays.

Name: TEXT

Restrictions: This item can be utilized with all N90 interface types except the serial port module. Ring and node addresses must be set to zero when the N90 interface is a CIC.

Subscripts	Purpose	DDE Type	Access
.BLINK	message blinking is requested	discrete	read
.COLOR	color selection	integer	read
.Q	quality (0 - good, 1 - bad)	discrete	read
.TIME	time (expressed as seconds since 1/1/1970) of last point value update	integer	read
.TIMETEXT	time (month, day, year, hour, minute, second) of last point value update	message	read
.UPDATES	running count of updates received for this point	integer	read
.V	value (requested message number)	integer	read

Example usage's are:

TEXT.V,1,2,3,50
Reads N90 TEXT message number value at ring 1, node 2, module 3, block 50 into a DDE integer tag.

TEXT.BLINK,1,2,3,50
Reads N90 TEXT blink flag at ring 1, node 2, module 3, block 50 into a DDE discrete tag.

TEXT.COLOR,1,2,3,50
Reads N90 TEXT color number at ring 1, node 2, module 3, block 50 into a DDE integer tag.

TEXT.Q,1,2,3,50
Reads N90 TEXT quality at ring 1, node 2, module 3, block 50 into a DDE discrete tag.

TEXT.V,2,4,31,9998
Reads TEXT message number value at ring 2, node 4, module 31block 9998 into a DDE integer tag.

8. Block Configuration Data Item - BLOCK

The N90DDE server block configuration data item maps MMI DDE tags such that general purpose N90 block output value reading, configuration reading and tuning is supported. These tags allow any N90 function block output value to be read, block specifications displayed and tuned. Ring and node addresses must be set to zero when the N90 interface is a CIC. The “BLOCK” data item supports the following value subscripts:

Subscripts	Purpose	DDE Type	Access
.RING	ring address for targeted block activity	integer	read/write
.NODE	node address for targeted block activity	integer	read/write
.MODULE	module address for targeted block activity	integer	read/write
.BLOCK	block number for targeted activity	integer	read/write
.READ	read block request flag of the addressed block	discrete	read/write
.READOUT	read block output request flag of the addressed block	discrete	read/write
.NEXT	read next block request flag from current addressed block	discrete	read/write
.TUNE	tuning request flag for the addressed block with updated values	discrete	read/write
.ACKNAK	result of last block request	integer	read
.V	block output value of last block output read request	real	read
.Q	block output quality for the last output read request (0 - good, 1 - bad)	discrete	read
.FC	function code number of the last block read	integer	read
.FCNAME	function code name of the last block read	message	read
.OKS_1	flags addressed block spec 1 is ok - (when 1)	discrete	read
•	•	•	•
•	•	•	•
.OKS_63	flags addressed block spec 63 is ok - (when 1)	discrete	read
.TUNABLES_1	flags addressed block spec 1 is tunable - (when 1)	discrete	read
•	•	•	•
•	•	•	•
.TUNABLES_63	flags addressed block spec 63 is tunable - (when 1)	discrete	read
.VALUES_1	value of addressed block spec 1 (if spec is ok, OKS_1)	real	read/write
•	•	•	•
•	•	•	•
.VALUES_63	value of addressed block spec 63 (if spec is ok, OKS_63)	real	read/write
.DESCS_1	description of addressed block spec 1	message	read
•	•	•	•
•	•	•	•
.DESCS_63	description of addressed block spec 63	message	read

BLOCK.RING
Used to set the ring address the server should use when processing a read block output, read block, read next block, or tune block request. For Infinet based systems this range should be limited from 1 to 250. For Plant Loop or single node based systems the value of this item is not used in the block request.

BLOCK.NODE
Used to set the node address the server should use when processing a read block output, read block, read next block, or tune block request. For Infinet based systems this range should be limited from 1 to 250. For Plant Loop based systems the range should be limited from 1 to 63. For single node based systems the value of this item is not used in the block request.

BLOCK.MODULE
Used to set the module address the server should use when processing a read block output, read block, read next block, or tune block request. The range for this item should be limited from 2 to 31.

BLOCK.BLOCK
Used to set the block address the server should use when processing a read block output, read block, read next block, or tune block request. The range for this item is dependent on the destination module type. The absolute range should be limited from 0 to 9999.

BLOCK.READ
Used to request the server to read the block data currently addressed by the BLOCK.RING, BLOCK.NODE, BLOCK.MODULE, and BLOCK.BLOCK items. The server automatically resets this flag to zero after it has accepted and processed the read request.

BLOCK.READOUT
Used to request the server to read the block output currently addressed by the BLOCK.RING, BLOCK.NODE, BLOCK.MODULE, and BLOCK.BLOCK items. The server automatically resets this flag to zero after it has accepted and processed the block output read request.

BLOCK.NEXT
Used to request the server to read the next block data from the block currently addressed by the BLOCK.RING, BLOCK.NODE, BLOCK.MODULE, and BLOCK.BLOCK items. The server automatically resets this flag to zero after it has accepted and processed the read next request. It also updates the BLOCK.BLOCK item with the next block number read. This item is useful for “walking” through a N90 modules configuration.

BLOCK.TUNE
Used to request the server to tune the block currently addressed by the BLOCK.RING, BLOCK.NODE, BLOCK.MODULE, and BLOCK.BLOCK items. The server automatically resets this flag to zero after it has accepted and processed the tune request. Prior to requesting a tune, the block should have been read using BLOCK.READ and the newly desired tunable specification values set in the appropriate BLOCK.VALUES_x items.
BLOCK.ACKNAK

Indicates result of last BLOCK.READ, BLOCK.NEXT, BLOCK.TUNE or BLOCK.READOUT request. This item is updated after the corresponding request flag has been reset by N90DDE indicating the requested operation has completed. When this item has a value of zero, the requested BLOCK activity completed with no errors. Otherwise, its value indicates the type of error that has occurred. Common errors are:

ACKNAK	Description
0	Request completed with no errors.
16	Number out of range, (invalid ring, node, module or block address specified).
32	Destination node is off line.
39	Destination ring is off line.
100	Target module does not know how to respond to the request.
102	Target module is in a mode that cannot support the current request.
104	Target module request is for an invalid block number.
105	Target module request is for a valid block number but it has not been defined.
106	Target module request is for a block that does not have any readable specifications.
110	Target module is not responding.

BLOCK.V
The server writes the block output value retrieved to this item when a read block output request is issued.

BLOCK.Q
The server writes the block output quality retrieved to this item when a read block output request is issued.

BLOCK.FC
The server writes the function code number retrieved to this item when a read block or read next block request is issued.

BLOCK.FCNAME
The server writes the name of the function code to this item when a read block or read next block request is issued. Function code names are recorded in the F1_130.INI and F131_255.INI files located in the windows directory. These files can be edited using a standard ASCII editor such as notepad to add future function codes developed by Bailey Controls or customize the names of any UDF function codes you may have developed.

BLOCK.OKS_x
The server uses the BLOCK.OKS_1 through BLOCK.OKS_63 items to flag whether or not the corresponding specification number is valid for the function block just read. Since the maximum number of specifications a function block can have is 63, there are 63 of these items. A value of one indicates the specification is valid and zero means not valid. These flags can be used to enable or disable display or data input elements on an MMI graphic based on a specification being valid or not.

BLOCK.TUNABLES_x
The server uses the BLOCK.TUNABLES_1 through BLOCK.TUNABLES_63 items to flag whether or not the corresponding specification number for the function block just read is tunable. Since the maximum number of specifications a function block can have is 63, there are 63 of these items. A value of one indicates the specification is tunable and zero means it is not. (The item is zero if its corresponding specification is not valid.) These flags can be used to enable or disable display and data input elements on an MMI graphic based on a specification being tunable.

BLOCK.VALUES_x
The server uses the BLOCK.VALUES_1 through BLOCK.VALUES_63 items to convey the values for the corresponding specification number of the function block just read. Since the maximum number of specifications a function block can have is 63, there are 63 of these items. All values are converted to real regardless of there actual type within the N90 controller. Likewise, when a value is written to this item for tuning purposes, it is written as a real but the server will convert it to the type appropriate for the function block being tuned. (This item is set to zero if its corresponding specification is not valid.)

BLOCK.DESCS_x
The server uses the BLOCK.DESCS_1 through BLOCK.DESCS_63 items to convey the specification names for the corresponding specification number of the function block just read. Since the maximum number of specifications a function block can have is 63, there are 63 of these items. (This item is set to a blank message if its corresponding specification is not valid.) Function block specification names are recorded in the F1_130.INI and F131_255.INI files located in the windows directory. These files can be edited using a standard ASCII editor such as notepad to add future function codes and/or specification descriptions as they are developed by Bailey Controls or define customized names of any UDF function block specifications you may have assigned.

The BLOCK item and its corresponding subscripts enable development of a MMI graphic devoted to reading and tuning N90 controller blocks. Maximum flexibility is given as to the format of the window. By defining function code and specification names in the F1_130.INI and F131_255.INI files, server upgrades are not required to accommodate new or changes to Bailey function codes. End users can add support for new function codes by editing these files. Note that by default, function code and specification names will have their numbers appended in front of their descriptions. This can be disabled by editing the N90DATA.INI file located in the windows directory. Look for the [APPENDS] section and set the FC and/or Specs items to “No”.

The BLOCK.XLS spreadsheet found in the BAILEY subdirectory of the installation floppy disk gives a good example on how to use the BLOCK item. This spreadsheet was developed using Excel 5.0 and assumes a N90DDE Server topic name called “N90DDE”.

9. 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 the N90 interface, N90DDE 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” N90DDE 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 N90DDE server and DDE points within the MMI tag name database, the server will begin talking with the N90 interface when the MMI is commanded online. The startup pattern will vary based on the type of N90 interface being utilized. The first thing you should notice is the N90 interface serial processing card LEDs begin to sequence. Shortly thereafter you may hear the loop interface termination unit relays click on and off several times as the server is identifying the N90 interface type. Just prior to downloading the database to the N90 interface, it will be restarted and the loop interface termination unit relays will click off isolating it from the communication loop. Next you should observe the N90 interface serial processing card LEDs sequence at a steady rate as the database is being downloaded. Upon completing the database download, the N90 interface will be command on-line, at which time the loop interface termination unit relays will click on and the loop interface card LEDs begin to count loop messages. Thereafter, the N90 interface serial processing card LEDs will sequence sporadically at the poll interval setup by the server topic.

If you experience problems with establishing communication between the N90DDE server and the N90 interface, if possible it is a good idea to verify the setup by trying to communicate using the Bailey TXTEWS software. Generally if this software functions OK, you should not experience problems with N90DDE server.

All N90 interfaces have a series of four or eight red LEDs on the module that processes serial communication and manages the N90 database. Don't confuse this card with the module that handles the interface with the N90 communication loop which also has a series of LEDs. An indication that communication with the N90 interface is occurring can be determined by looking at the serial processing card LEDs. (Hereafter, these LEDs will be referenced as N90 interface LEDs.) The N90 interface LEDs count commands and replies occurring between the computer and N90 interface.

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

If while starting up the Wonderware application a dialog appears indicating that a particular DDE conversation could not be initiated or from Excel remote data is not accessible, this means a topic is being referenced that has not been defined using N90DDE. Determine the topic causing the error and define it within N90DDE.

9.2. Rejects an Item that Does Have the Right Syntax

If N90DDE rejects an item that you are positive has the right syntax, make sure that the item does not have the same N90 address (ring, node, module and block) of another previously created item of a different type. For example lets suppose the item STN.SP,1,2,3,4 was created and the item AIL.V,1,2,3,4 is attempted to be created. The AIL.V,1,2,3,4 item will be rejected even thought its syntax is correct because it has the same N90 address (1,2,3,4) as the previously created STN.SP item. This condition will be reflected in the WW logger when N90DDE error posting has been enabled and the WW toolkit version is being utilized. Also make sure that the item does not have a N90 address that is currently being used within another active topic.

9.3. No Communication

If the N90 interface LEDs do not sequence, this means the server is not able to successfully communicate with the N90 interface.

1.) Verify the server topic is associated with the same COM port of which the N90 interface has been cabled.

2.) Verify the N90 Interface device termination unit/module serial port jumpers are setup correctly.

3.) Verify RS232 cable is connected to the correct termination unit/module connector. Generally this is labeled as the terminal port.

4.) Verify the RS232 cable is connected to the correct PC COM port.

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

If the N90 interface LEDs sequence at a very steady and periodic rate, this means the server is connected to the N90 interface but the communication parameters are not set correctly.

1.) Verify the baud rate settings match between the N90 interface and COM port the server is using.

2.) Verify the parity (usually none) and stop bits (usually 1) match between the N90 interface and COM port the server is using.

3.) Verify the N90 interface has checksuming enabled.

If the N90 interface LEDs are not sequencing but the server indicates good communication status (STATS.COMM is zero)

1.) Verify the COM port associated with the topic definition is not a modem port.

2.) Verify the pin outs on the serial cable or N90 interface termination dipshunt settings do not have the TX and RX lines pinned together.

9.5. Not All N90 Server DDE Tags Are Receiving Data

1.) Verify point type indicated by item name actually exists and matches correct type within the N90 module data source.

2.) Verify "duplicated items" error message is not posted within WWLogger messages. If such an error is found, you have two or more items of different types pointing to the same address within N90. Search the tag name data base for duplicate addressing of different point types.

3.) 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.

4.) Make sure none of the Bailey controllers have addresses less than two.

5.) Indices or memory capacity of N90 interface has been exceeded. It is unlikely that you will encounter this error unless your application has a very large number of N90 points and the N90 interface device is a NCIU01 or INCIC01. Search the WWLogger messages for such an error.

6.) Memory capacity of N90 PCU node has been exceeded, check its status using the Bailey TXTEWS program. Note that this error is extremely rare.

9.6. Cannot Export or Control Data Within N90

1.) Verify that the Bailey interface device supports this type of activity.

2.) Verify that the appropriate MMI DDE tag has been marked as read/write.

3.) Verify the allowable ranges for the tag name match those configured within N90.

4.) Verify the N90 interface does not have monitor mode enabled (very unlikely). Monitor mode is set using an ASCII terminal attached to the termination unit/module printer port and that port selected for the utility mode (consult Bailey N90 interface manual). Normally, monitor mode is disabled when a N90 interface is received from the Bailey factory. It is unlikely that monitor mode is enabled unless someone had attached an ASCII terminal at one time and was "experimenting" with the available options within the N90 interface utility menus.

9.7. Data Values Exchanged Between N90 and Server Are Different

1.) Verify that the tag point Min EU matches the N90 zero and Min Raw is set equal to Min EU. Also verify that the tag point Max EU matches value set within N90 and Max Raw is set equal to Max EU. Note that the EU Max is calculated as the N90 zero plus span. Make sure you calculated the MMI settings correctly.

9.8. Server Will Not Time Sync N90

1.) Verify that the N90 interface device is not a NSPM01, NCIC01, or NCIU01. These devices do not allow a computer to send time synchronization commands to N90. If the N90 interface device is one of these types, disable PC Gets and / or Sets Bailey System Time option.

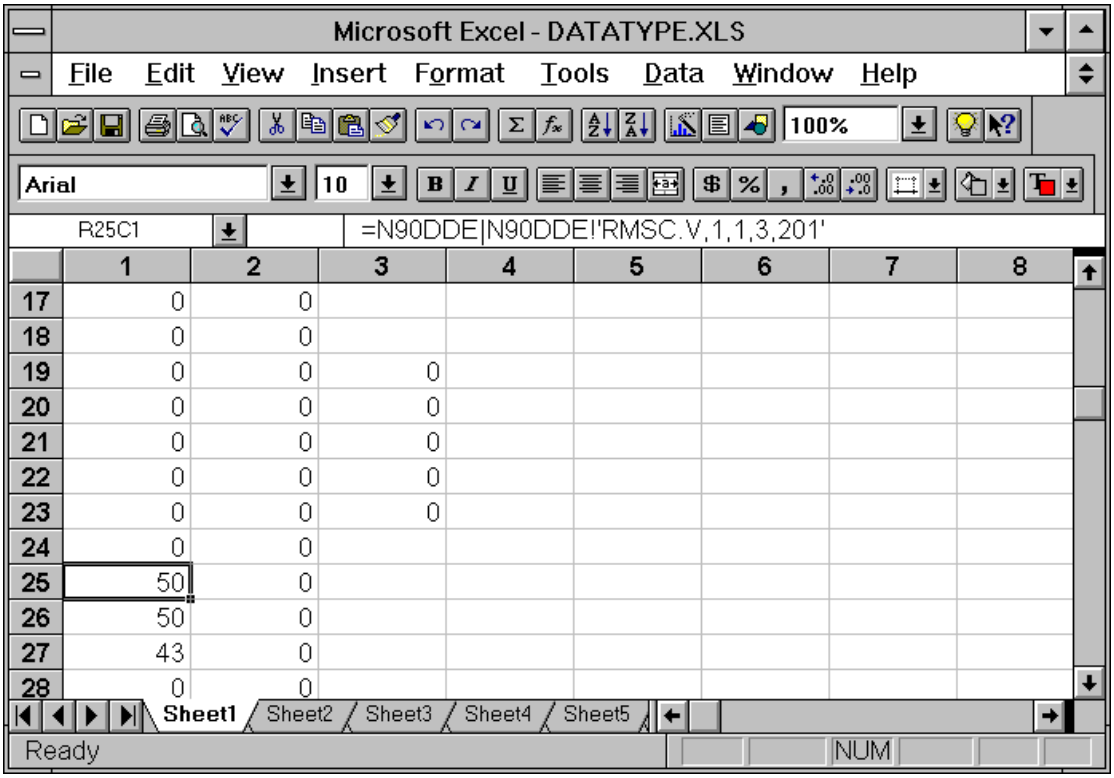
2.) Verify that time synchronization has been enabled for the correct type of N90 system (Infi 90 or Network 90).

3.) For Plantloop based N90 systems, verify that the node map is configured correctly for every node in the system regardless of whether or not data is being exchanged with all nodes.

4.) Note that the server gives precedence to MCS and OIS nodes as becoming the time synchronization master.

10. Reading DDE Values into Excel

N90DDE Server DDE values can be read into Excel by entering the appropriately formatted DDE remote reference formulas into the cells into which N90DDE data values are to be read. The following example shows the definition of one such formula:

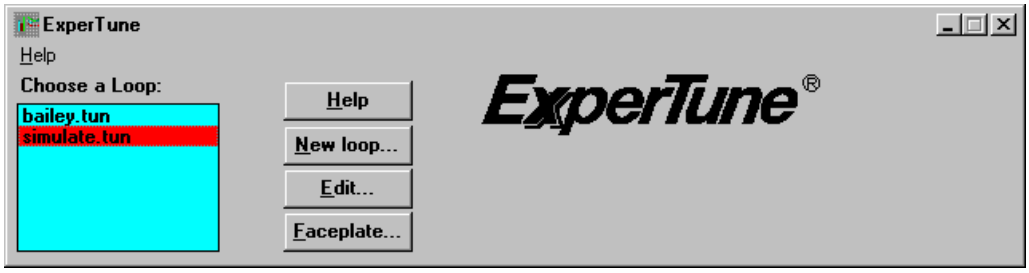


	1	2	3	4	5	6	7	8
17	0	0						
18	0	0						
19	0	0	0					
20	0	0	0					
21	0	0	0					
22	0	0	0					
23	0	0	0					
24	0	0						
25	50	0						
26	50	0						
27	43	0						
28	0	0						

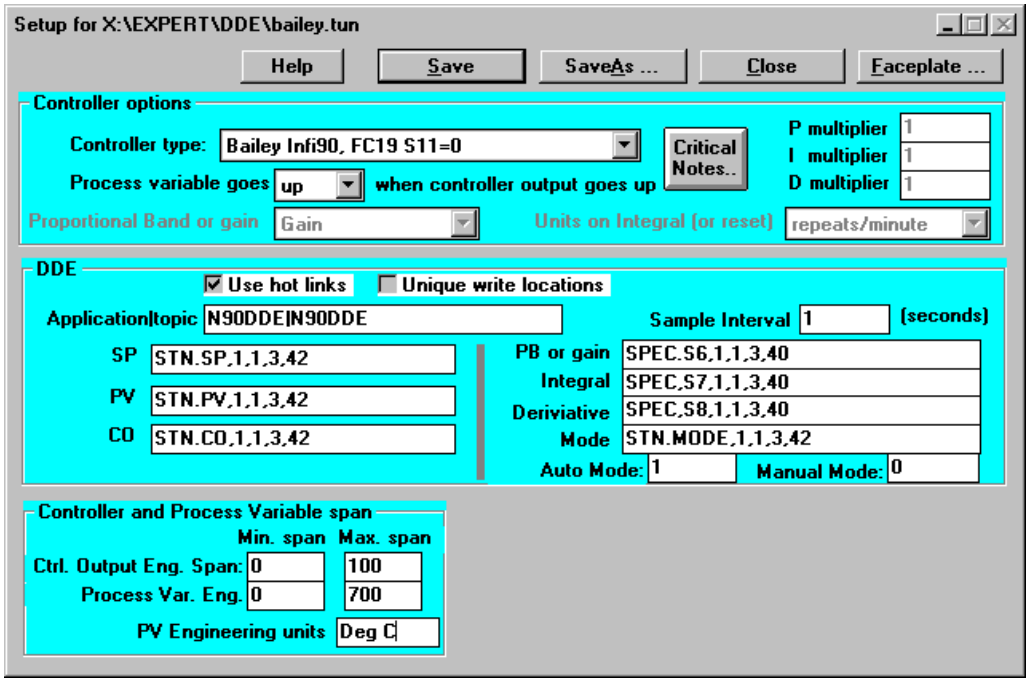
In this example the cell at R25C1 references a N90DDE tag value from a topic named N90DDE exchanging a RMSC value tag setup to read from node 1, PCU 1, module 3, and block 201. Notice that the remote reference formula requires the item name to be enclosed in single quotes as shown in the example as ‘RMSC.V,1,1,3,201’. The “DATATYPE.XLS” Excel 97 spreadsheet is included in the SAMPLES subdirectory where N90DDE has been installed. It contains examples for all point types and their subscripts. It also supports “POKING” values to the server for select items.

11. Using N90DDE Server With ExperTune

N90DDE Server is very easy to setup and use with the ExperTune PID tuning package available from Gerry Engineering Software Inc (414) 628-0088. All that is required is the definition of a topic and DDE tags based on the STN and SPEC items. The following example shows the ExperTune main screen with a loop definition called "bailey.tun":



When the "bailey.tun" loop is chosen and the edit button selected the following configuration screen is activated:



This screen is used to define data associated with a particular PID loop within the Bailey system. Newer versions of ExperTune have a wizard button that will cause the various fields within this dialog to be automatically configured by the wizard. All that is required to utilize the wizard is to have N90DDE Server running before running the wizard, have a known N90DDE topic predefined and know the base address of the Bailey Station block that has the associated PID block to be tuned. Consult the ExperTune user manual for the details of each field and operation of the wizard.

The example assumes a Bailey Station block (F.C. 21, 22, 23, or 80) located at ring 1, node 1, module 3, block 42 and PID block (F.C. 18 or 19) located at ring 1, node 1, module 3, block 40. The fields related to accessing data via N90DDE Server are:

ExperTune Field	Purpose
Application|Topic	Defines the DDE Server application name (always N90DDE) and topic (as setup using N90DDE Server) to be used for data access. The example assumes a topic also named N90DDE.
SP	Defines the N90DDE Server item used to read/write the Station function code setpoint value.
PV	Defines the N90DDE Server item used to read/write the Station function code process variable value.
CO	Defines the N90DDE Server item used to read/write the Station function code control output value.
PB or gain	Defines the N90DDE Server item used to read/tune the PID function code specification number for the proportional term.
Integral	Defines the N90DDE Server item used to read/tune the PID function code specification number for the integral term.
Derivative	Defines the N90DDE Server item used to read/tune the PID function code specification number for the derivative term.
Mode	Defines the N90DDE Server item used to read/write the Station function code mode.
Auto Mode	Defines the value to be written which corresponds with the Auto mode (always 1).
Manual Mode	Defines the value to be written which corresponds with the Manual mode (always 0).

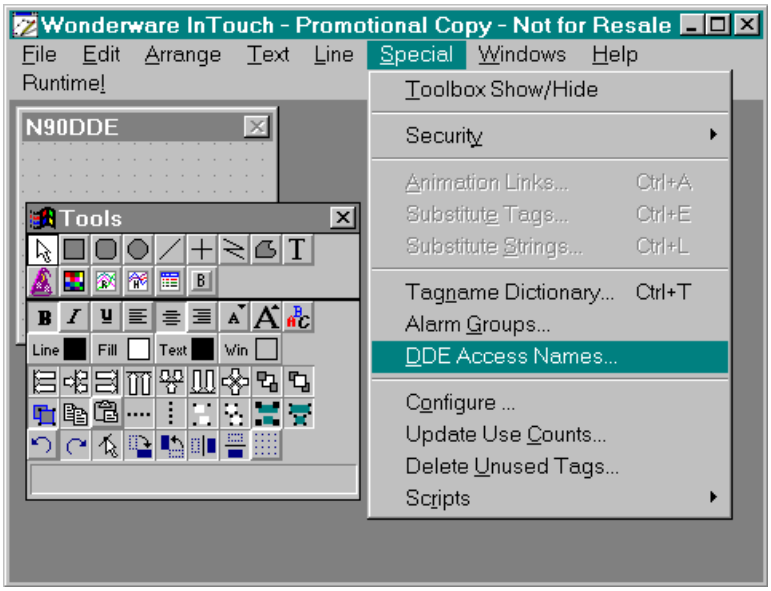
Note that ExperTune does not automatically run the N90DDE Server application. Therefore, make sure N90DDE Server is running prior to accessing the ExperTune faceplate screen (available from the ExperTune main screen).

12. Using N90DDE Server With InTouch

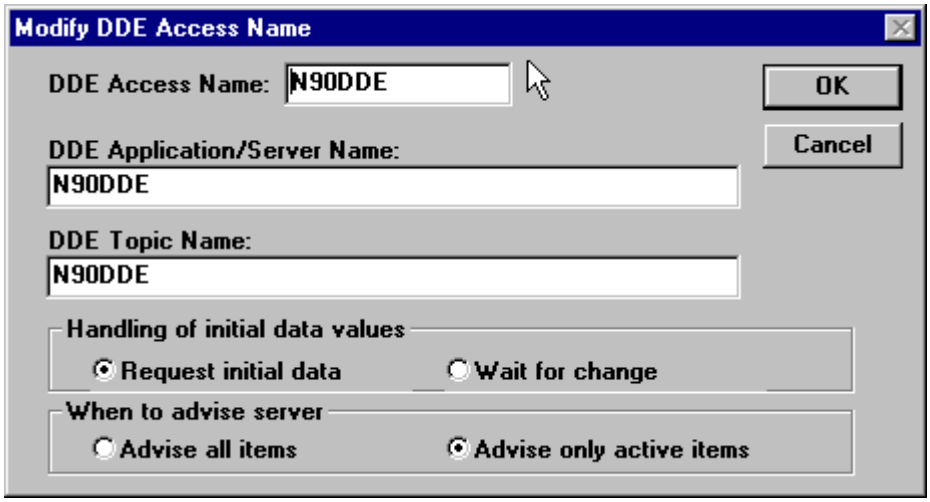
This section presents information related to using N90DDE Server with Wonderware Software Corporation's InTouch third party man-machine interface software. It assumes InTouch has already been installed on the target machine along with the version of N90DDE Server developed with the Wonderware DDE Server 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.

12.1. Definition of N90DDE Server DDE Points

InTouch WindowMaker is used to configure DDE points to be exchanged with the N90DDE 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. Typically the DDE access name is given the same name as the server which is N90DDE.

DDE Topic Name:
Topic name defined within the N90DDE server. In this example a topic name of ICICOM2 had been assigned which signifies a N90 ICI interface device attached to the COM2 port.

Request Initial Data Radio Button:
Selected **only** if the server program is other than a Wonderware DDE server such as N90DDE **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 N90DDE.

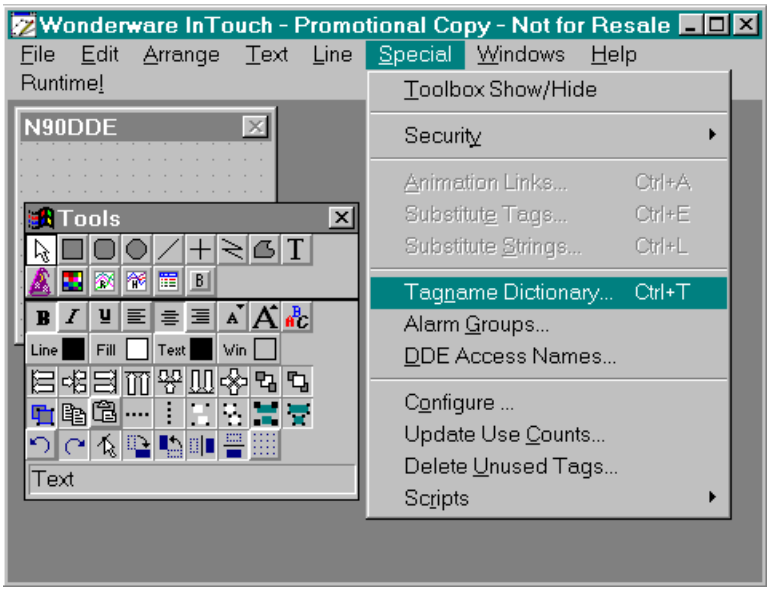
Advise All Items Radio Button:
Selected if the server program is to poll for all data whether or not it is in visible windows, alarmed, logged, trended or used in a script. If data throughput becomes sluggish for large N90DDE server applications and this option is selected, de-selecting it will slightly 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.

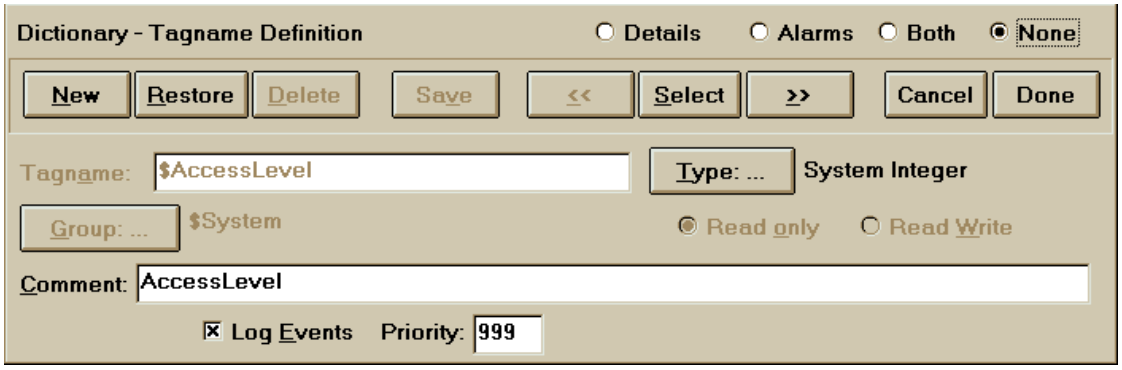
Cancel Button:
Click on this button to cancel changes to the definition of the DDE access name.

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

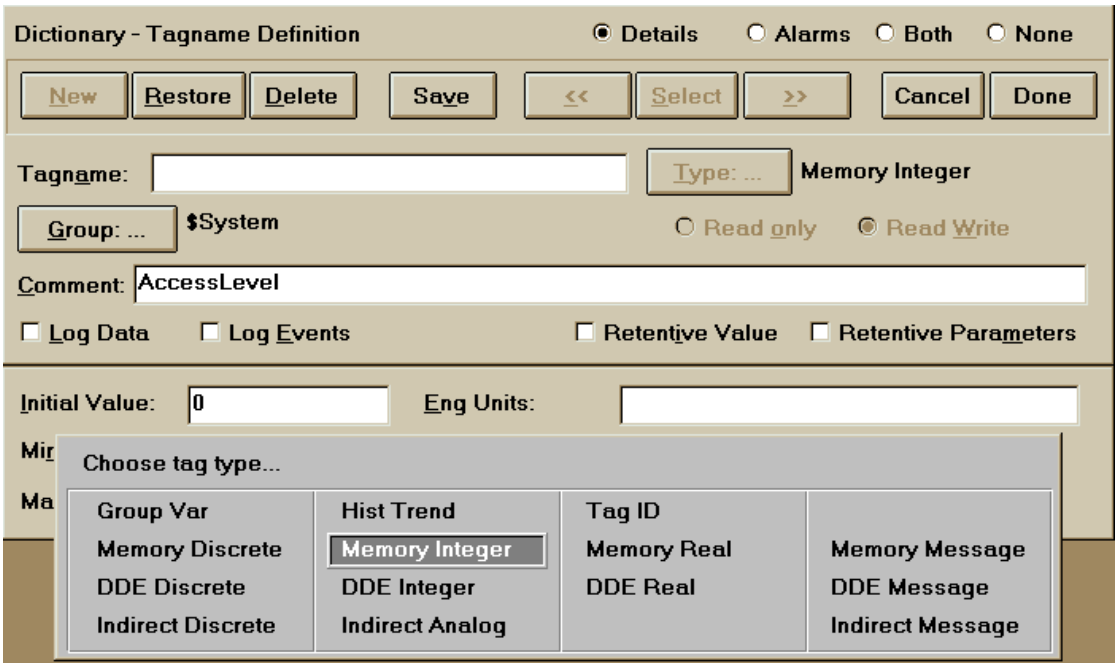
Once a DDE access name has been defined, tag names of DDE type can be defined. To define a tag name, the Tag Name Dictionary... sub-menu under the InTouch - WindowMaker Special menu. This selection is shown by the following screen.



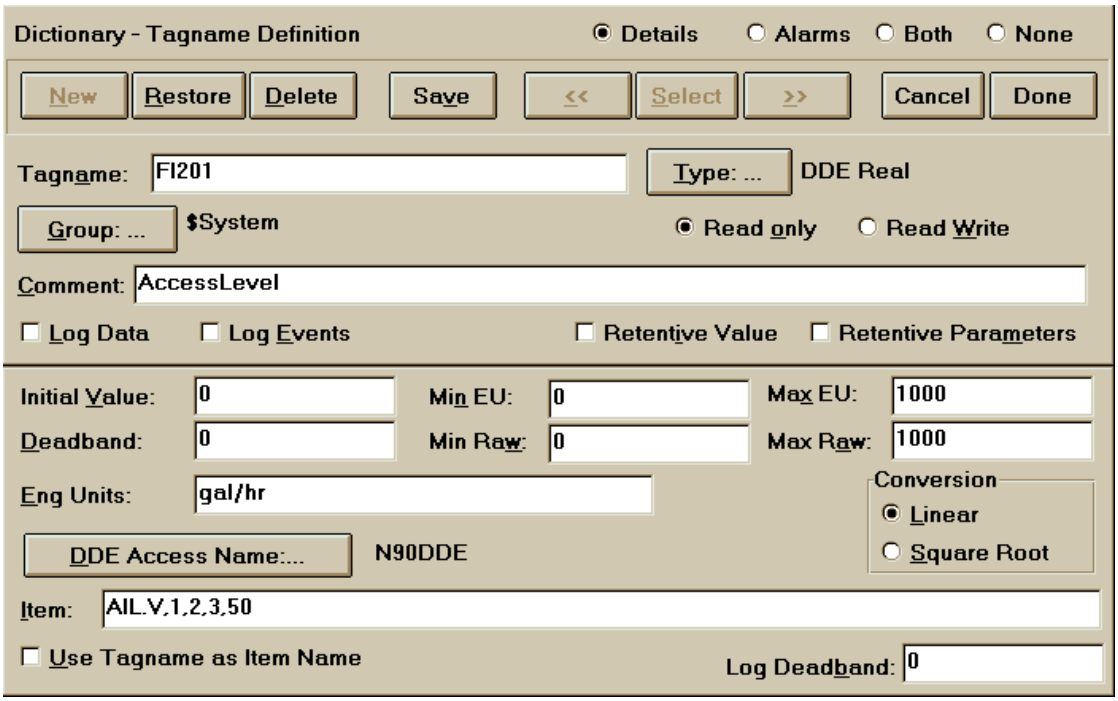
After selecting Tag Name Dictionary... the following dialog will be displayed:



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 N90DDE 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 value of a N90 Analog Output / Loop Function Block (F.C. 30) located at ring 1, node 2, module 3, block 50. This has been defined by the item name AIL, subscript V (value) and addressing information 1,2,3,50. Note that the N90DDE 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 N90DDE 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 N90DDE 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.

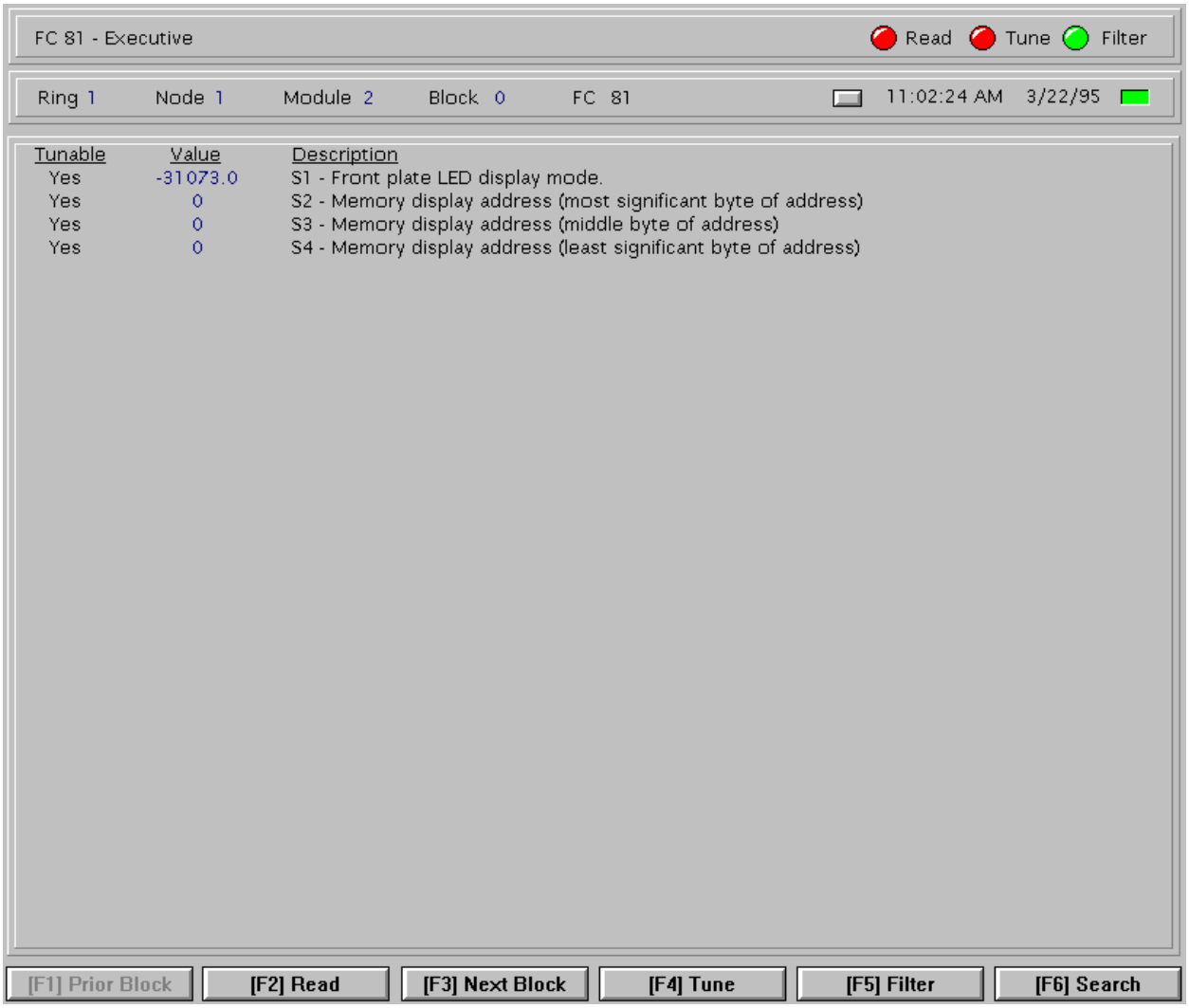
Conversion:
Not applicable to the N90DDE server since raw to engineering units conversion must always be disabled.

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 N90DDE server point type, its subscript and address within N90.

12.2. Configuration Reading and Tuning (CRT) Application

The Configuration Reading and Tuning (CRT) InTouch application has been designed and developed by RoviSys. Its purpose is to support reading and tuning of Bailey Controls function block control logic. CRT supports all models of controllers within the Bailey INFI 90, NETWORK 90 and Command series systems. CRT works with all computer interfaces supported by the N90DDE server. The server must be version 4.2 or later and configured for a topic named "N90DDE". CRT is optionally loaded by setup. Before running CRT under WindowViewer you must configure a topic within N90DDE Server by the name of N90DDE. When InTouch View runs the CRT application, CRT initially displays a startup window announcing itself. Thereafter, the following window is displayed:



The user interacts with CRT via pushbuttons, keyboard input and LED indicators. CRT is designed to be run from the keyboard only, mouse and any combination of the two. There are several methods for accessing user input fields such as ring, node, module or block. The first is to position the cursor over the field using

the mouse and click with the left button. This will cause an input box to be draw where the user enters the new value. Another method is to enter the “tab” key successively until the desired input field is outlined. Then the “enter” key is pressed to bring up the user input box. The third method is to push the “Ctrl” key and first letter of the desired field. For example “Ctrl R” will cause an input box to be drawn for ring entry. CRT may be merged into another application to provide block viewing and tuning capabilities. In addition to the database tags, CRT also utilizes data change scripts and application startup / runtime scripts. When merging CRT make sure the windows, database, and scripts are all included. Following is an explanation of each field on the CRT window.

Read LED:
Indicates block reading activity. Green indicates a block read in process with red indicating no activity.

Tune LED:
Indicates block tuning activity. Green indicates a block tune in process with red indicating no activity.

Filter LED:
Indicates search filter active. Green indicates search filter is enabled with red indicating disabled.

Comm LED:
Indicates communication status of the server with the Bailey interface device. Green indicates good with red being bad.

Information Button:
Push this button for overview information concerning CRT.

Ring #:
Current ring number selected for block reading and tuning. Enter “Ctrl R” to enter a new ring number.

Node #:
Current node number selected for block reading and tuning. Enter “Ctrl N” to enter a new node number.

Module #:
Current module number selected for block reading and tuning. Enter “Ctrl M” to enter a new module number.

Block #:
Current block number selected for block reading and tuning. Enter “Ctrl B” to enter a new block number.

FC #:

Current function code number for the last ring, node, module and block read or tuned.

[F1] Prior Block Button:
This button causes the block number read previously to the current block to be read again and its specification data displayed.

[F2] Read Button:
This button causes the currently assigned block number to be read and its specification data displayed.

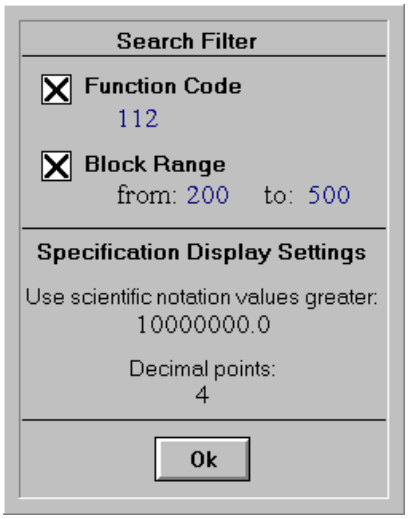
[F3] Next Block Button:
This button causes the next block number from the currently assigned block number to be read and its specification data displayed.

[F4] Tune Button:
This button causes any tunable specification values the user had entered to be written to the currently assigned block number. The module must be in execute mode.

[F5] Filter Button:
This button causes the search filter definition dialog to be displayed.

[F6] Search Button:
This button causes a search to be conducted for a user specified function code and optional block range that is setup using the filter button.

Pressing the “Filter” button causes the following dialog to be displayed:



This dialog is used to enable and disable the search filter. When enabled, the user defines a specific function code they are interested in finding. This is accomplished by clicking on the “Function Code” check box. This box is

alternatively enabled and disabled by entering “Ctrl F”. When enabled the user must “tab” to the function code user entry field (or click on in with the mouse) to enter the desired function code to be found. Optionally, a beginning and ending search block number can be specified by enabling the “Block Range” check box. This box is alternatively enabled and disabled by entering “Ctrl B”. If block range searching is not specified, the search begins at the currently selected block number.

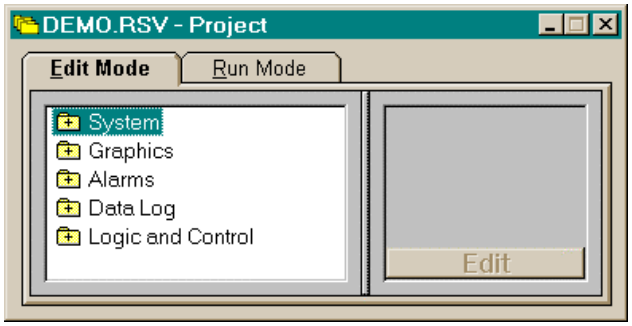
The filter dialog also allows the user to customize how specification values are displayed. The largest value before scientific notation is utilized can be entered along with the number of decimal point digits to be displayed.

13. Using N90DDE Server With RSView32

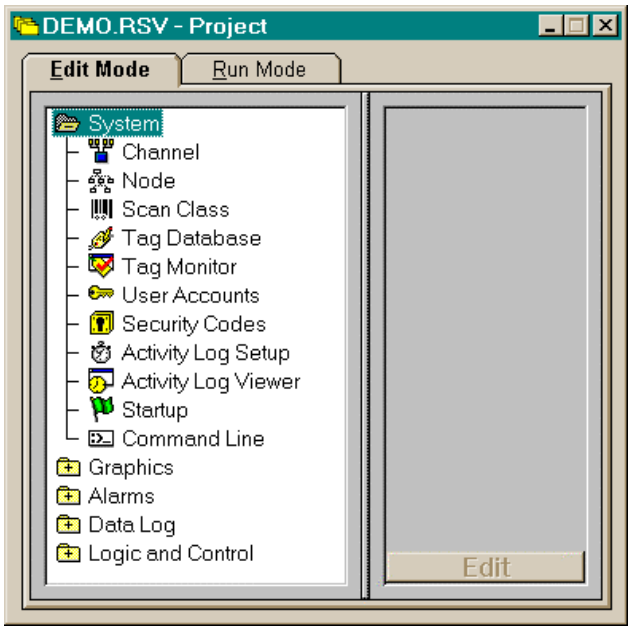
This section presents information related to using N90DDE Server with Rockwell Software's RSView32 third party man-machine interface software. It assumes RSView32 has already been installed on the target machine along with the version of N90DDE Server developed with the Rockwell Software DDE Server toolkit. Note that the screen snapshots presented in this section were taken from the RSView version software. The configuration principles remain the same for the RSView32 series software.

13.1. Definition of N90DDE Server DDE Points

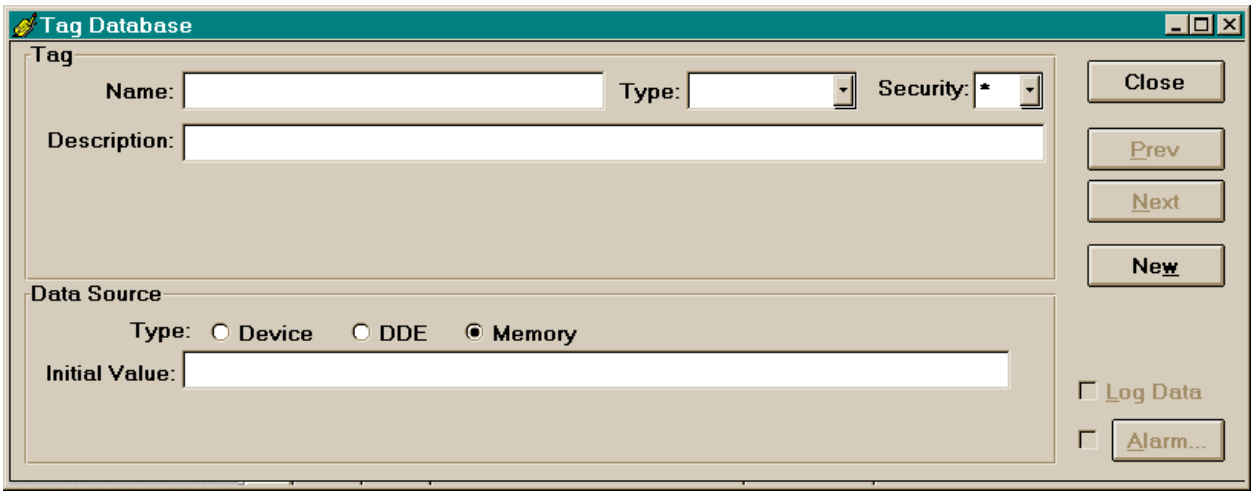
Defined a project using RSView Works. The following dialog is an example of a RSView demo project:



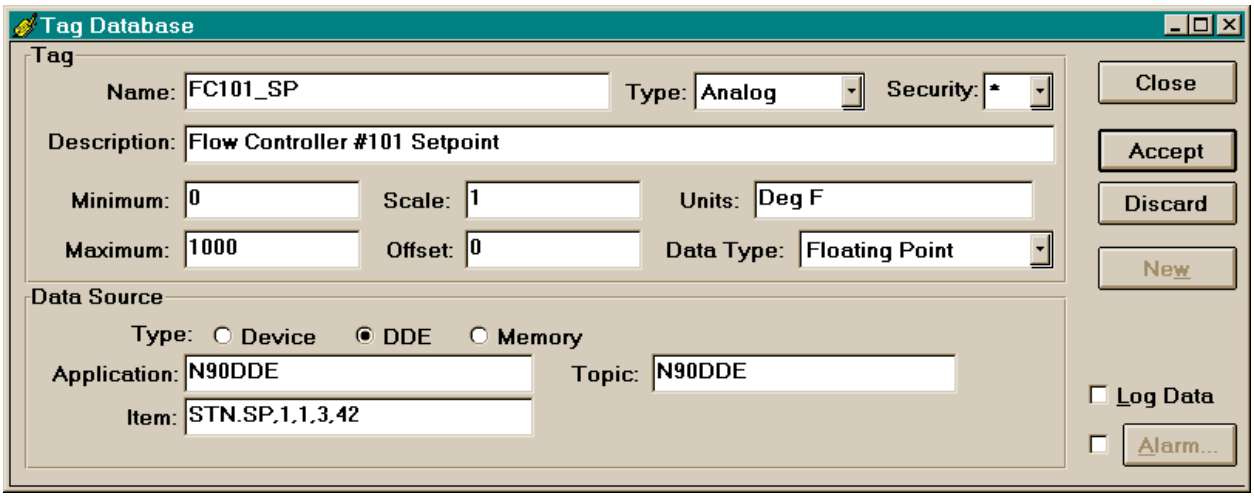
Double click on the "System" folder to expand it shown as follows:



Double click on “Tag Database” to configure DDE points to be exchanged with the N90DDE server. This will display the following point definition dialog:



This dialog allows definition of any type of RSView tag. Select the “DDE” radio button as the data source and click on the “New” button or press the “Alt W” keys to define a new tag. This will display the following dialog which has been filled out to define a DDE tag that will exchange the setpoint of a Bailey Control Station (F.C. 80) block located at ring 1, node 1, module 3, block 42:



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:
Selects the source of this data. Obviously points being provided by N90DDE server must be of type DDE.

Application:
This is the name of the DDE server application that is providing the data value. For N90DDE server this name is always N90DDE.

Topic:
Topic name defined within the N90DDE server. In this example a topic name of N90DDE has been assigned.

Item:
Name of the N90DDE server point type, its subscript and address within N90. Note that the N90 block address must always be the base function block number even for blocks with multiple outputs like the example station block.

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

Accept Button:
Accepts the current definition for this point.

Discard Button:
Discards the current definition for this point.

New Button:
Request definition of a new point.

Log Data Checkbox:
Request that logging be enabled for this point.

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.