# Table of Contents

| Revision | Description | Date |
|----------|-------------|------|
| A | Manual Released | 1999 |
| B | Name change, correct where applicable with document | 04/07 |

The Pro-face/Xycom DeviceNet slave communication module, (from now on referred to as the "COM-DNS"), allows OpenHMI to communicate on a DeviceNet network supporting up to 255 bytes of I/O to and from the network. The COM-DNS is only capable of solicited communication. This means that the master controller, (PLC), provides information displayed on the OpenHMI screen and requests the information required to interact with the process. The states of inputs and outputs at other nodes are acquired passively, where the master controller issues these states to the COM-DNS. This data gathering function is not automatic. Therefore, logic must be created in the master controller, directing information to both the intended I/O device and the COM-DNS.

This quick start will guide the user through the following processes:

1. Configuring the COM-DNS for the desired number of inputs and outputs

2. Using SyCon, setup the SMSI Server

3. Generate a simple screen in Intouch to graphically display the results

The configuration of the COM-DNS (using SyCon) is done on the Runtime Workstation. Defining Tagnames in the SMSIServer and the Intouch application is typically done in the development system environment.

Let us assume that a block of I/O exists at MacID (node address) 15. This I/O block has 1 byte of input (produced) data and 1 byte of output (consumed) data. The master controller is programmed to read and write data to and from this I/O block in a polled fashion. The controller is also programmed to send the same data previously sent to the I/O block, and the data it reads from the I/O block, to the COM-DNS, at MacID 9, for representation on the screen. In this way, the COM-DNS has an image of the inputs and outputs of the I/O block.

## Initial Configuration of Runtime Workstation

The first step is to configure the Slave Module for the correct number of inputs and outputs. This is done using a utility called SyCon (System Configurator). SyCon is provided free of charge as a demo version. The only limitation of demo mode is that only two network nodes can be configured. Since the Slave module is the only node we need to configure for the HMI application, demo mode is sufficient.

Start SyCon by selecting it from the Program Group < SyCon System Configurator > in the Start Menu.  From the Toolbar select File - New and from the bus selections pick DeviceNet.  A blank network configuration page will appear on the screen.  SyCon is designed such that a Master must be established on the network before slave devices can be added.  For the purposes of configuring the Slave Module, a DeviceNet master must exist on the network.

The Tool Button  is used to insert a master.

It can also be inserted by selecting Insert - Master.  Once done a selection window appears containing various master modules.  Select COM-DNM by clicking on it, then clicking Add followed by OK.  The DeviceNet-Master should appear at the top of the display and will be connected to the black vertical bar representing the DeviceNet network.

Now the Slave Module can be added to the network.

Click on the Tool Button  or use the Insert – Slave command.

Once done, an insertion point cursor will appear on the screen, representing the target position where the Slave Module will exist on the network.  Move the insertion point below the DeviceNet-Master and click the left mouse button to place the Slave Module.  A selection window should appear on the screen containing various types of slave modules.  Select COM-DNS by double clicking on it or by single clicking on it, then clicking Add followed by OK.

The COM-DNS should appear on the screen connected to the DeviceNet network as shown below.
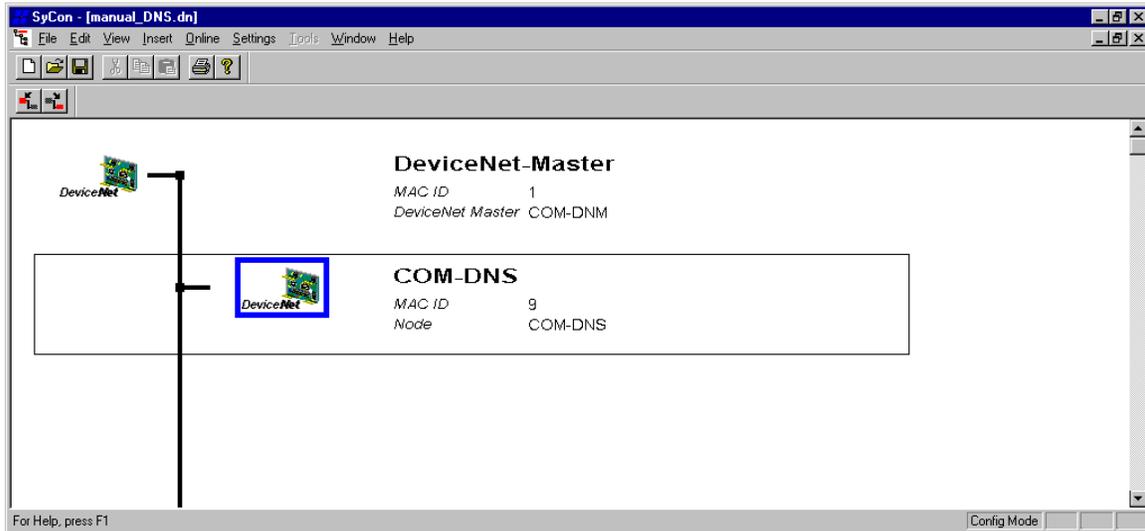


Right click on the COM-DNS and then select Slave Configuration, or (optionally) double click on the COM-DNS to bring up the Device Configuration window.
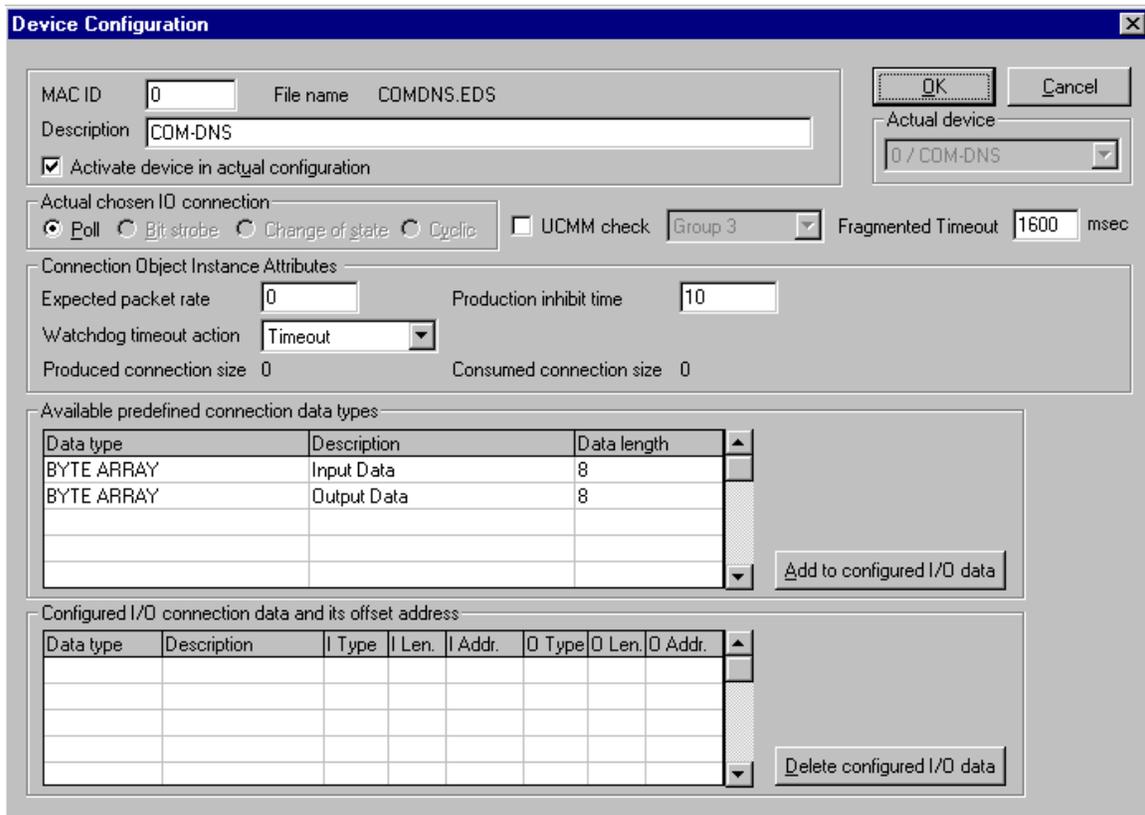
For the COM-DNS, the only valid selections are the MAC ID and the Configured I/O connection data.

The other parameters such as Expected packet rate and Production inhibit time are configured for a DeviceNet master (COM-DNM) and not the COM-DNS.

Change the **MAC ID** to 9, (which is what we previously chose for this application). Since the I/O block, on the DeviceNet network, has 1 byte of input and 1 byte of output, the COM-DNS will be set up for 2 bytes of output to allow the master controller to write the input and output images to the COM-DNS. Under **Available predefined connection data types**, double click on **"BYTE ARRAY    Output Data    8"** or select it and then click on **"Add to configured I/O data"**, to add this data type to the Configured I/O connection data. The default size for this is 8 bytes. For this example, we'll change it to 2 bytes. Select the "8" under **"O Len"** and use the backspace key to erase it, then type in 2. Pressing enter now will close the Device Configuration window. Be sure that all changes have been made correctly. The changes to the configuration should look like the following screen. You may press enter now or click on the OK.
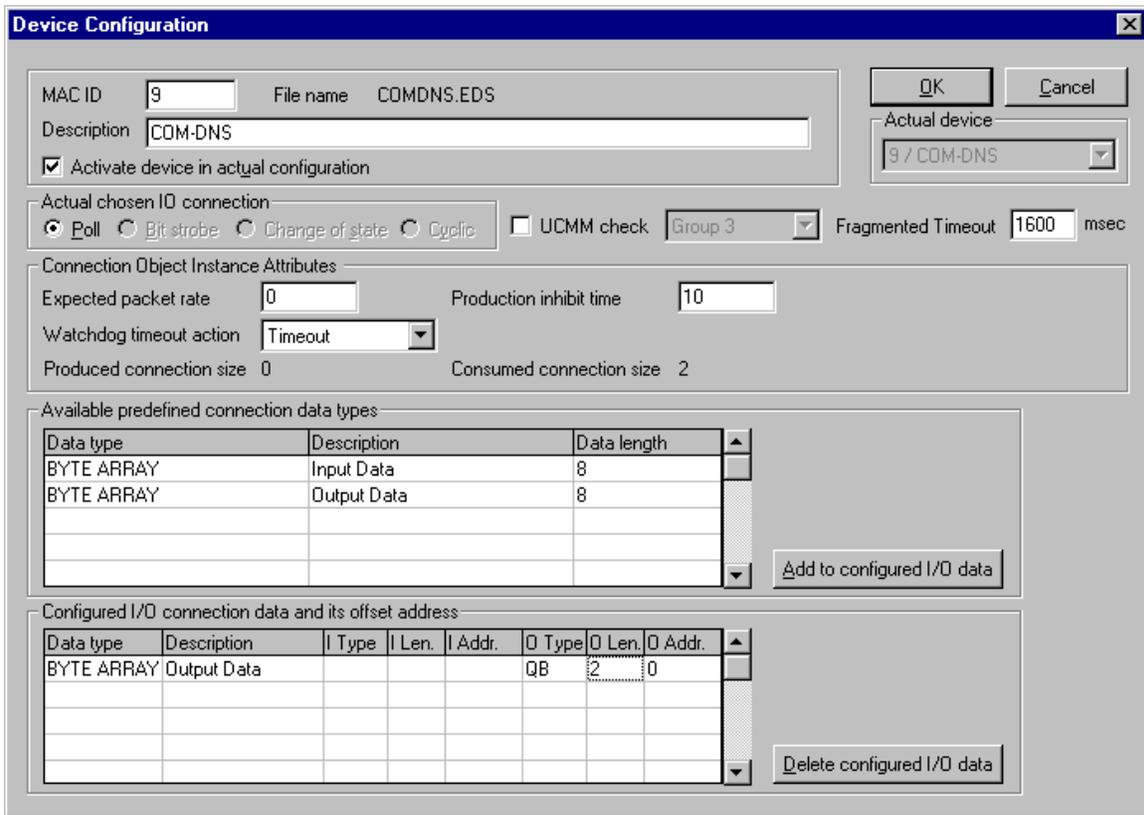


The COM-DNS is now configured for two bytes of output and MAC ID 9. This configuration must be downloaded to the flash memory within the COM-DNS. Once done, the COM-DNS will always power up running this configuration.

Right click on the COM-DNS and click on "Select as actual slave" in the pop-up window. A blue box should appear around the icon for the COM_DNS. This box indicates that the device plugged into the computer is the actual device to be downloaded to. From the menu, select Online then Download. A window will pop up asking if you really want to download. Select Yes.

A board assignment window will pop up requiring Board 0 to be defined. Pick the pull down menu arrow for Board 0. There should be at least two choices, "No device" and "COM-DNS". Pick COM-DNS. A download of the configuration to the COM-DNS should commence.

Now that the COM-DNS is configured, save the session by choosing File - Save from the menu bar.

**Board Assignment**

Select devices to be accessible by dual-port memory driver

|  | | DPM addr. | Firmware | | DPM-size | IRQ | Error |
|---|---|---|---|---|---|---|---|
| Board 0 | COM-DNS | D0000 | DNS | COM-DNS | 8 kB | 0 | 0 |
| Board 1 | No device | 00000 | not available | | 0 kB | 0 | 0 |
| Board 2 | No device | 00000 | not available | | 0 kB | 0 | 0 |
| Board 3 | No device | 00000 | not available | | 0 kB | 0 | 0 |

OK
Cancel

Select devices and the serial PC interface to be accessible with

|  | | Firmware | Error | |
|---|---|---|---|---|
| COM 1 | No device | not available | 0 | Check COM 1 |
| COM 2 | No device | not available | 0 | Check COM 2 |
| COM 3 | No device | not available | -20 | Check COM 3 |
| COM 4 | No device | not available | -20 | Check COM 4 |

Node ID          None          Description     COM-DNS

# SMSI Server

The SMSI Server is the link between Intouch and the Device Driver for the COM-DNS board.  Intouch communicates through a DDE link with the server.  The server shares the same Tagname as Intouch and conveys the translation to the proper addresses at the Dual Port Memory on the COM-DNS card.  The server must therefore be set up with Tagnames and their appropriate offsets.  These Tagnames will likely be pre-established in the development system environment.

The SMSI Server must read the COM-DNS directly to gather the configuration information.  This information is saved as a file and can be later transferred to and edited in the development system environment.

Invoke the SMSI Server from Start - Programs - SMSIServer - SMSIServer.  The following window should appear on the screen.
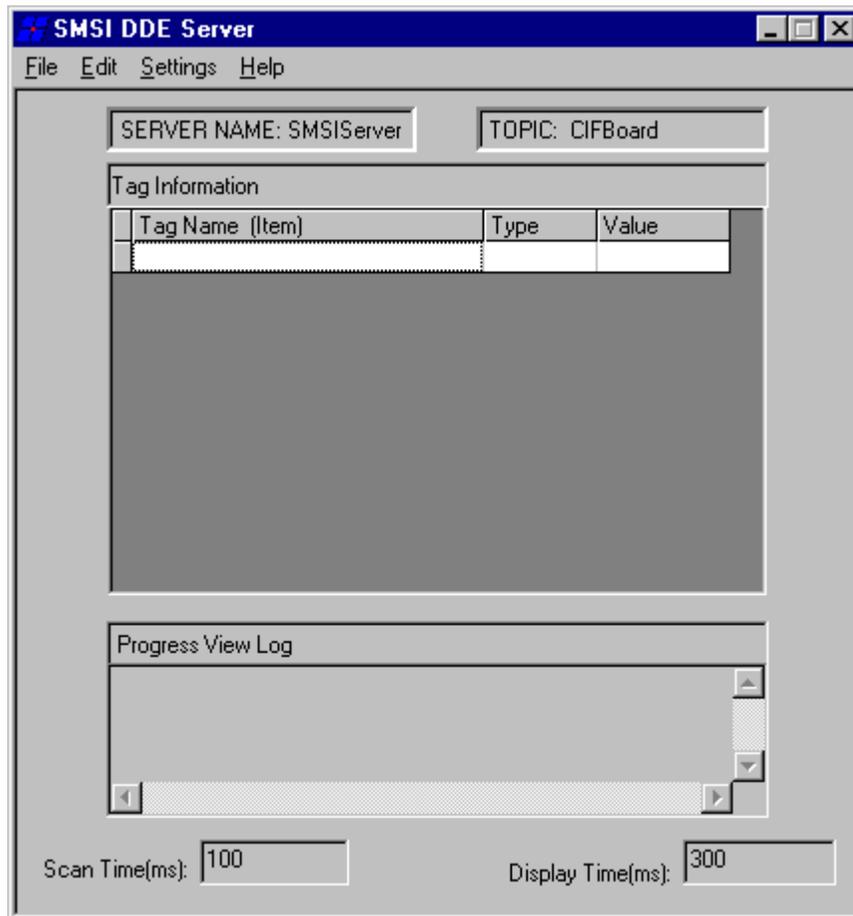
From the File menu, select New Board to bring up the following window.
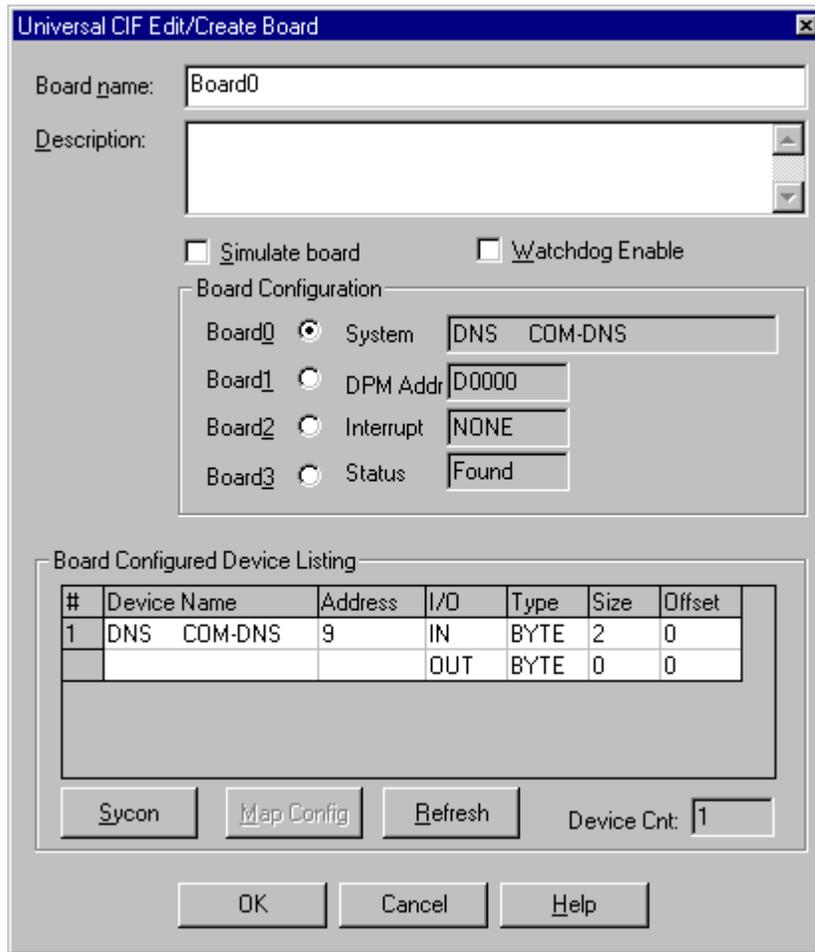


Notice that under the Board Configured Device Listing, the table shows the COM-DNS listed in the table. The SMSIServer automatically interrogated the COM-DNS to determine its configuration. Verify that it is set for Address 9, and that the Inputs are set to two and the outputs are zero.

Any data that Intouch receives from the server is considered an input. Data from Intouch to the server is considered output. Remember that the COM-DNS was configured for two bytes of output. During this configuration, SyCon uses the reference point of the master controller. Therefore, data sent to the COM-DNS from the controller is treated as output. This same data from the Intouch point of view is treated as input data. This is why the SMSI Server reports that there are two bytes of input, rather than the 2 bytes of output that was configured using SyCon.

Click OK to accept this new board.  The following window should appear.  Click on Save & Exit to save this configuration to a file.  Give it the name Example1.
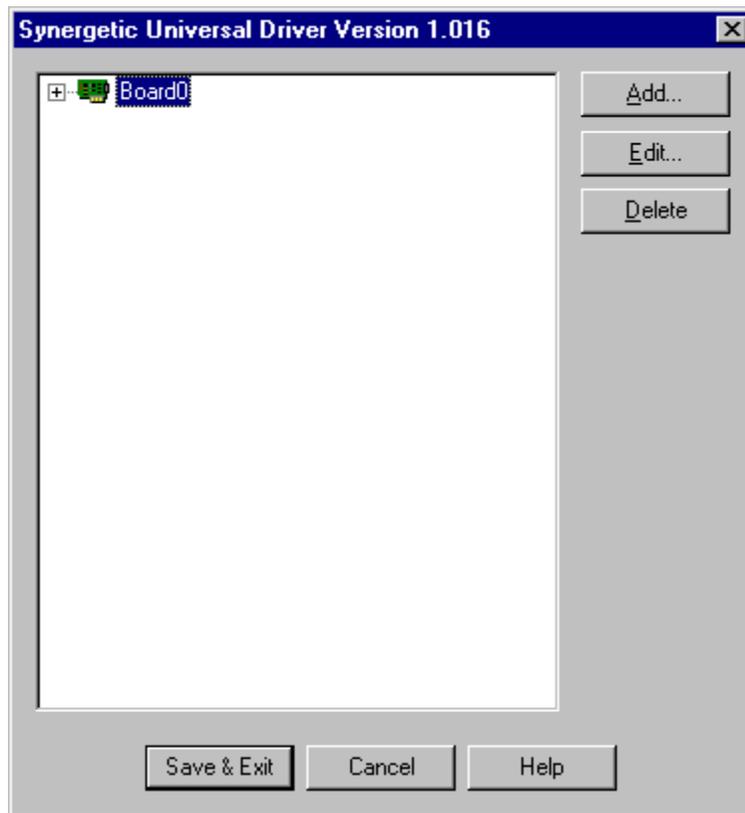
After the Save & Exit the following screen will appear. The SMSI Server automatically generates status tags for the board, some of which will say unused if it is inappropriate for the type of communication card plugged in. Definitions for these status tags can be found in the Help menu for the SMSI Server.
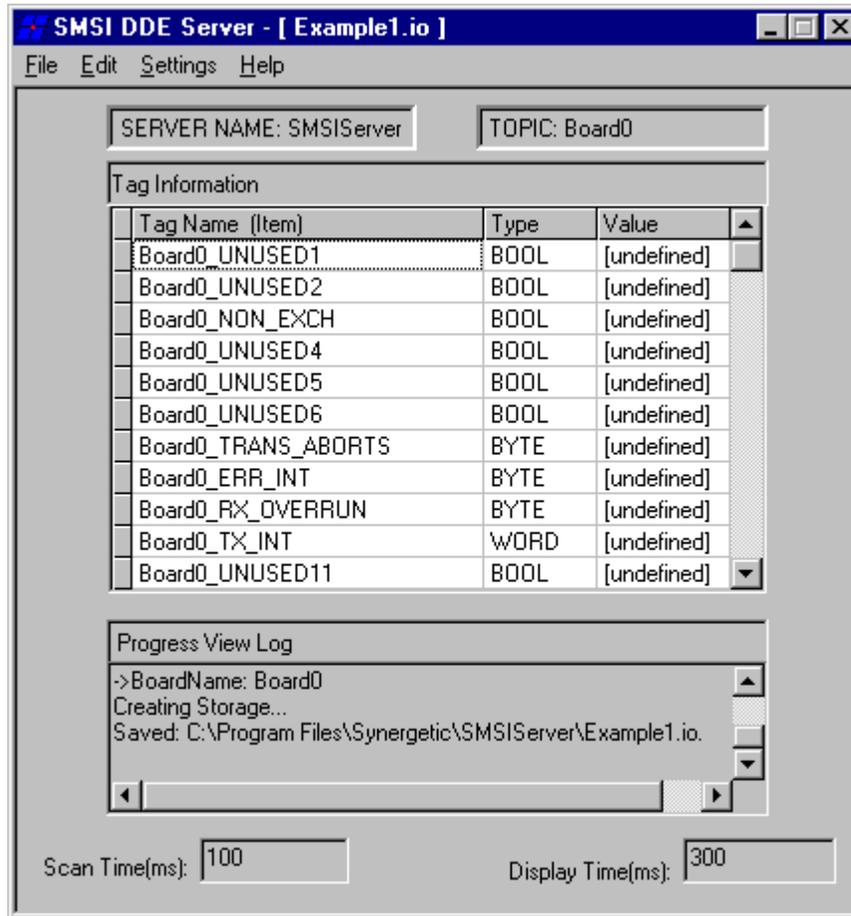
Before exiting the server, it is necessary to set it up to automatically load Example1.io whenever the SMSIServer is launched.  This is to avoid manual intervention.  To do this, select Settings – Startup Mode – Auto. This automatically loads the last file, (in this case Example1.io).  Now exit the server.

The file Example1.io can now be transferred to the development system.  On the development system, the file Example1.io should be placed in the folder C:\Program Files\Synergetic\SMSIServer.
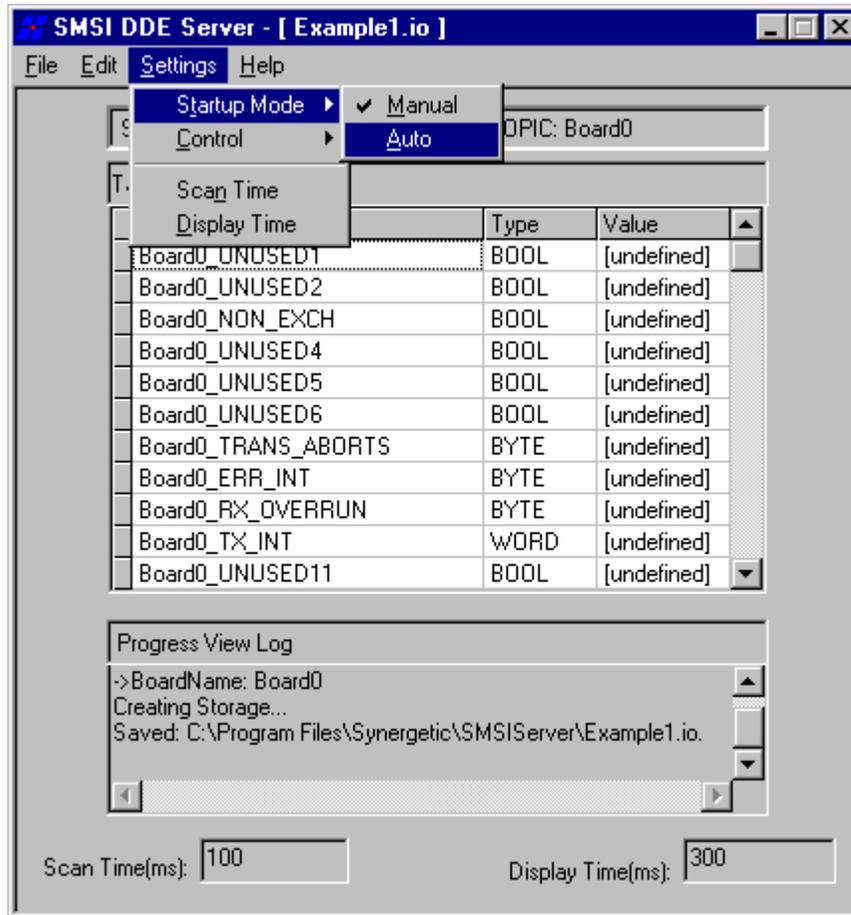
# Development System

Invoke the SMSI Server from Start - Programs - SMSIServer - SMSIServer.  The following window should appear on the screen.
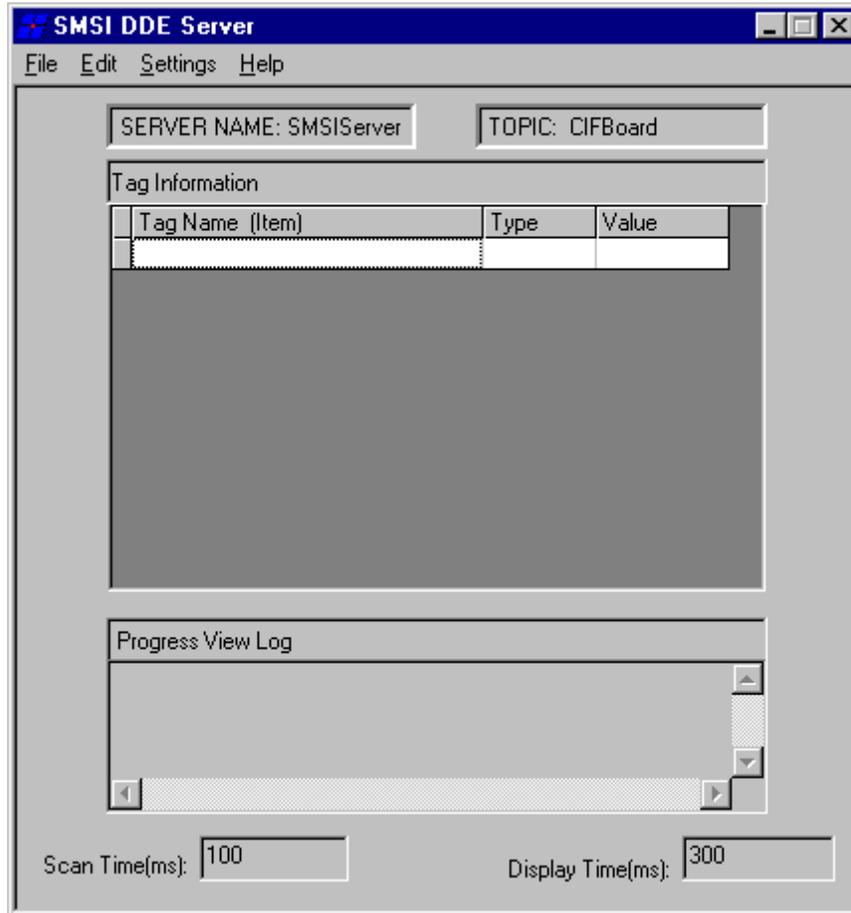
From the File menu, select Open Board and pick Example.io to bring up the following window.
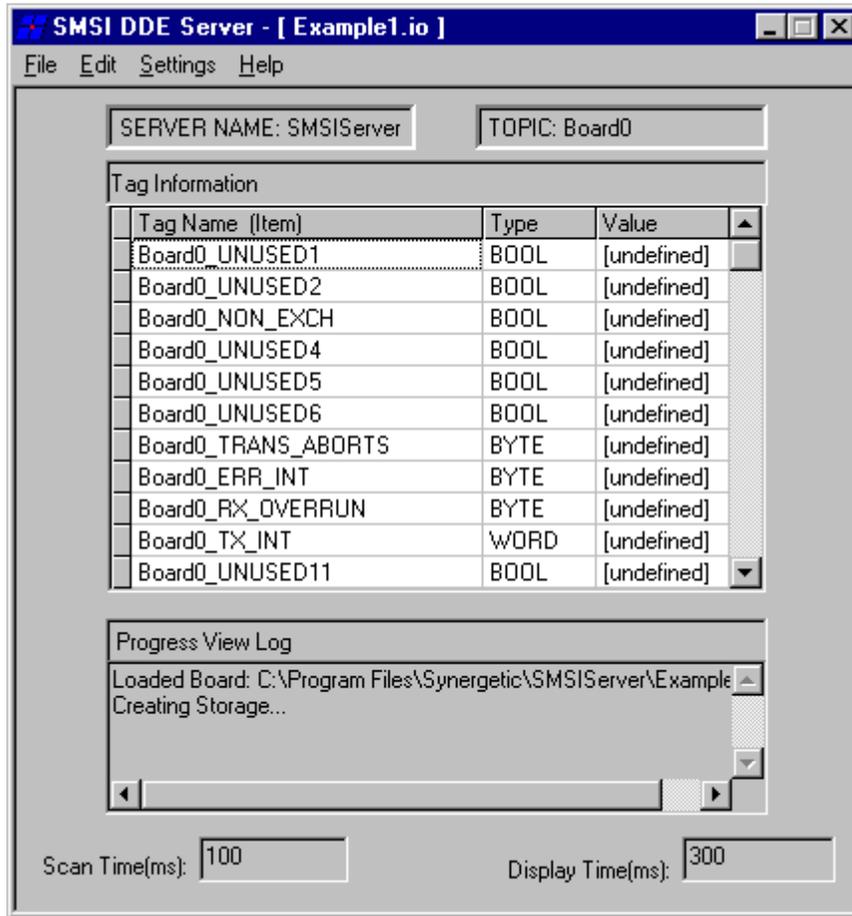
From the Edit menu, select Edit Board. When the window pops up, click on the "+" sign for Board0 to expand the tree as shown below. Addr9 should now be seen, which represents the COM-DNS at MAC ID 9. Click on the icon for Addr9 to highlight it, then select Add.
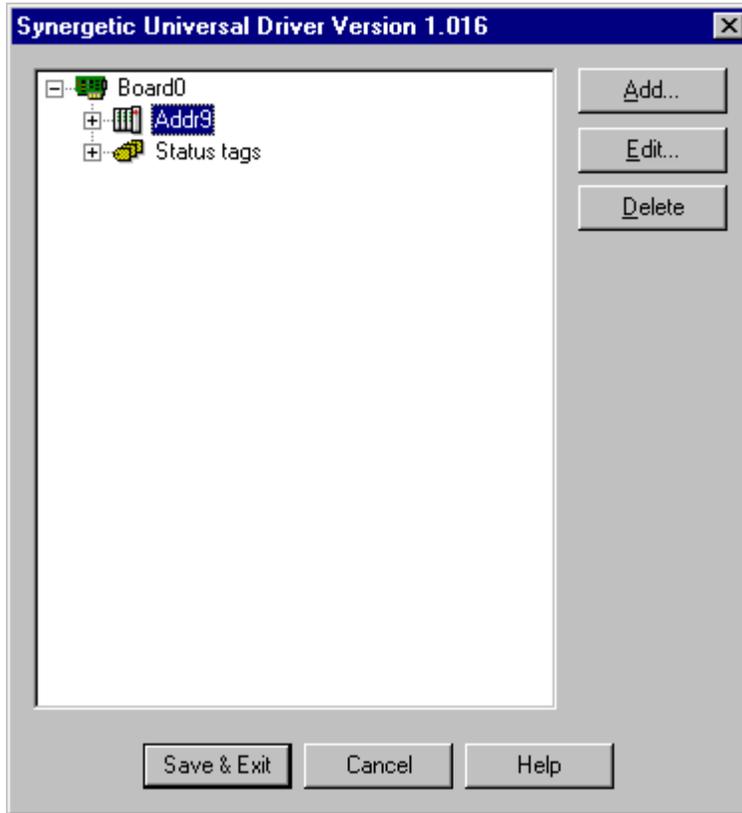
A "Create Tag" window will appear as shown below.
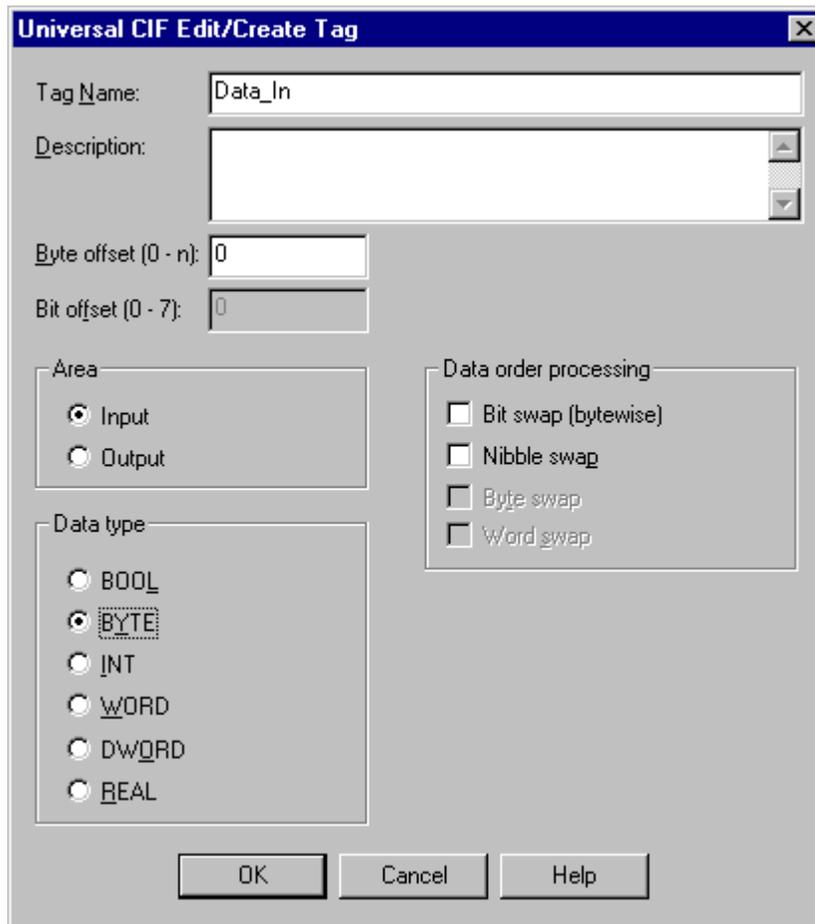


We will call the input data, from the I/O block, "Data_In" and the output data to the I/O block, "Data_Out".  Both of these images of the I/O block's inputs and outputs are written to the COM-DNS from the master controller and appear as inputs to Intouch.  Type in the first Tag name "Data_In".  You can optionally type in a meaningful description for the Tag.  Leave the "Area" defaulted to Input, and select BYTE for "Data type".  Click OK to accept this new Tag.  We should now be back at the Board0 edit window with "Data_In" shown as a Tag under Addr9.

Now click on the Addr9 icon again to highlight it and select "Add" to add the Tag "Data_out". Go through the same process for the "Data_In" but since this is the second of the two bytes, which will be sent to the COM-DNS, change the "Byte offset" from 0 to 1. If there had been three bytes to be sent to the COM-DNS, the third byte would be Byte offset 2, and so on. Then select OK to accept this Tag.
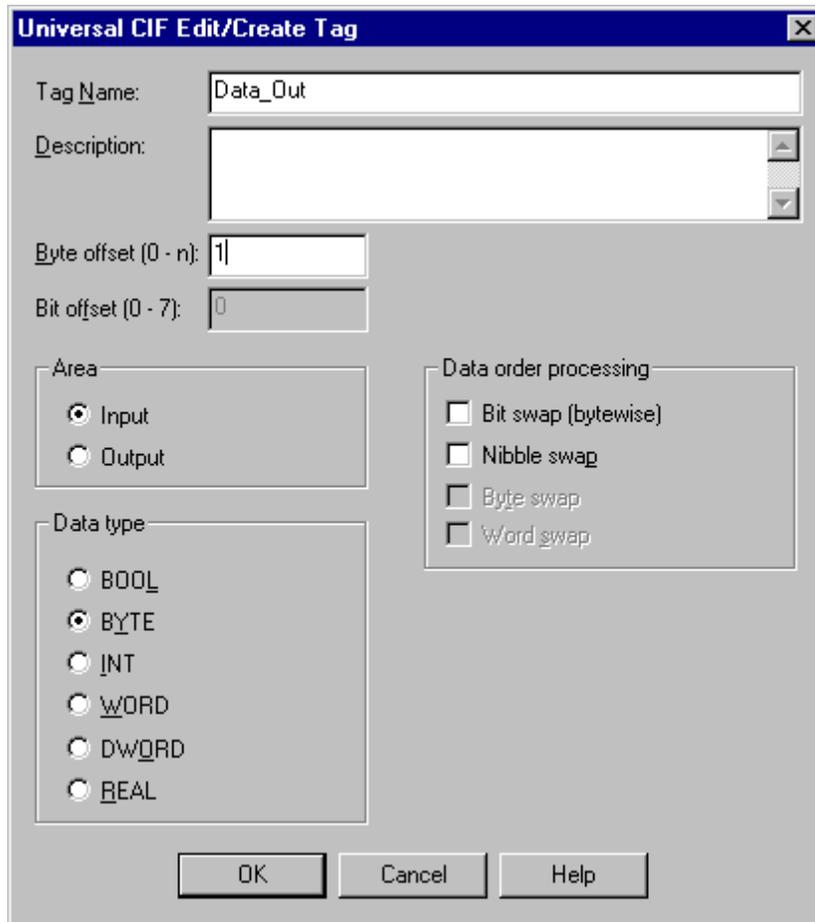
Now both Tags appear below Addr9.  Now click on Save & Exit to save the server's Tags to the Example1 file.
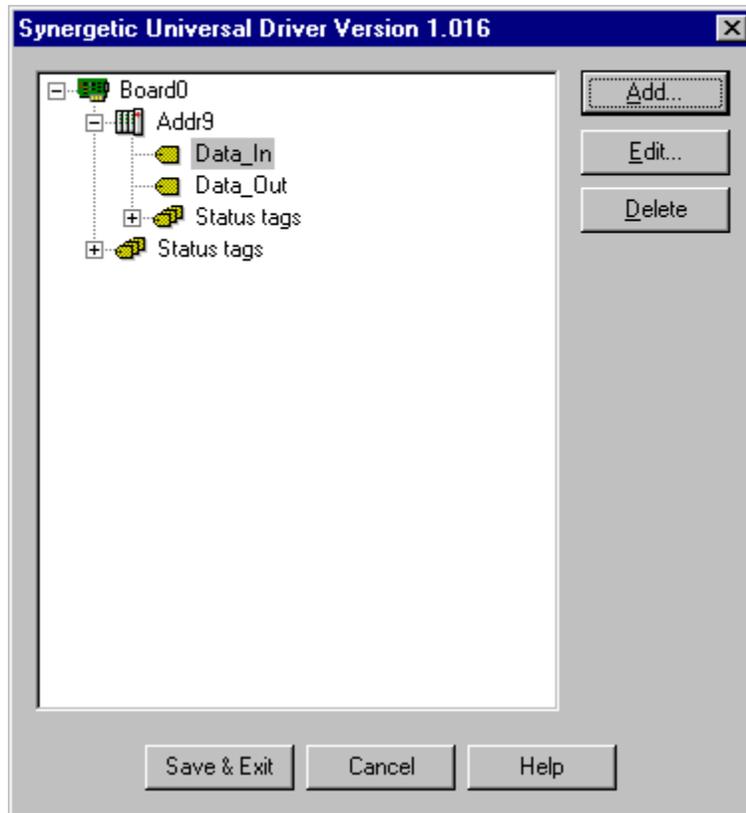
After the Save & Exit the following screen will appear.  Scroll down the list of Tagnames and note that the "Data_In" and "Data_Out" Tagname just entered appear at the bottom. Now exit the server.  The file, Example1.io should be found in the folder C:\Program Files\Synergetic\SMSISever.  This file must be downloaded to the Runtime Workstation system into the folder of the same name, C:\Program Files\Synergetic\SMSISever.

| Tag Name (Item) | Type | Value |
|---|---|---|
| Board0_UNUSED5 | BOOL | [undefined] |
| Board0_UNUSED6 | BOOL | [undefined] |
| Board0_TRANS_ABORTS | BYTE | [undefined] |
| Board0_ERR_INT | BYTE | [undefined] |
| Board0_RX_OVERRUN | BYTE | [undefined] |
| Board0_TX_INT | WORD | [undefined] |
| Board0_UNUSED11 | BOOL | [undefined] |
| Addr9_OnLine | BOOL | [undefined] |
| Addr9_Diag | BOOL | [undefined] |
| Data_In | BYTE | [undefined] |
| Data_Out | BYTE | [undefined] |

SERVER NAME: SMSISever    TOPIC: Board0

Tag Information

Progress View Log

->BoardName: Board0
Creating Storage...
Saved: C:\Program Files\Synergetic\SMSISever\Example1.io.

Scan Time(ms): 100        Display Time(ms): 300

# OpenHMI

Launch OpenHMI from **Start - Programs - WonderWare FactorySuite - OpenHMI.**
Create a new application and start Window Maker.  Create a new window by clicking on
the new window icon or by selecting "New Window" from the File menu.  Choose a
name for the window and window color then select **OK**.
Now we will create two value displays in the newly created window by utilizing the

"Wizards"  tool. Click on the Wizards tool to bring up the following selections.
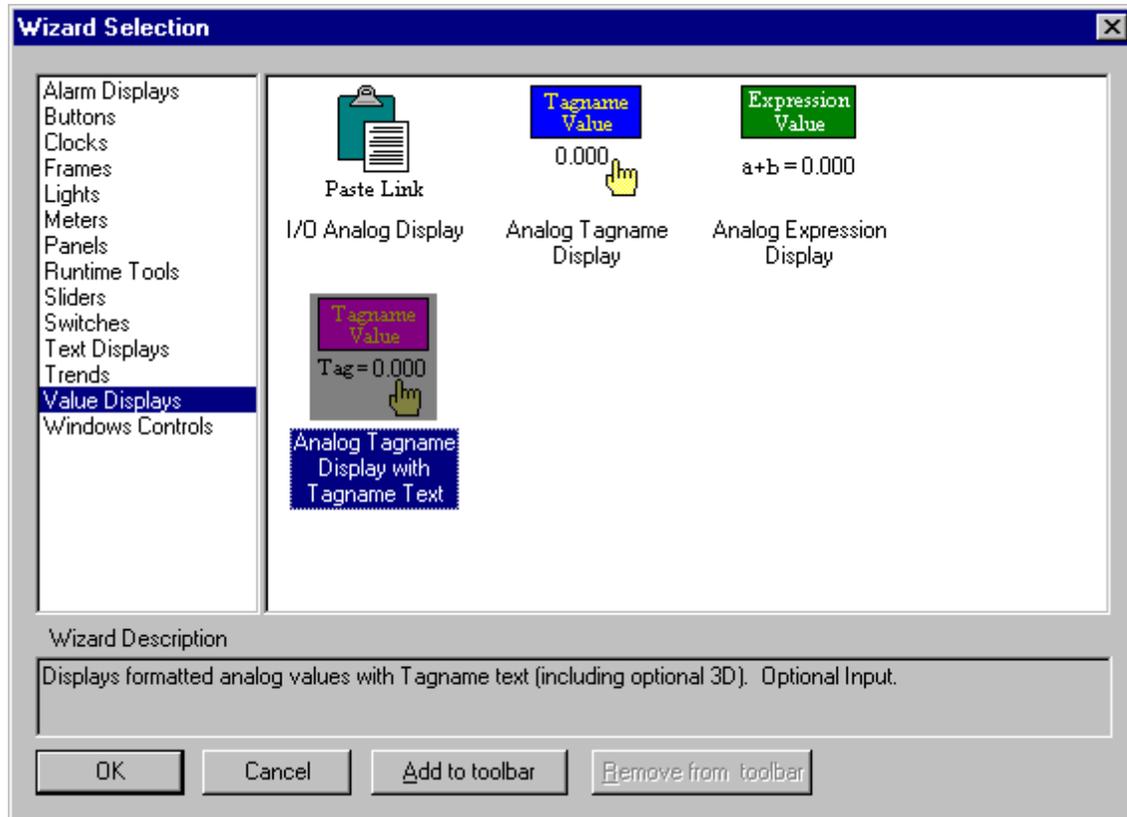


Choose the "Value Displays" and from the possible selections pick "Analog Tagname
Display with Tagname Text" followed by **OK**.

Position the cursor in the application window where you want the Value Display and left click the mouse to place it. Intouch generates a temporary Tagname for the Value Display, which must be changed.
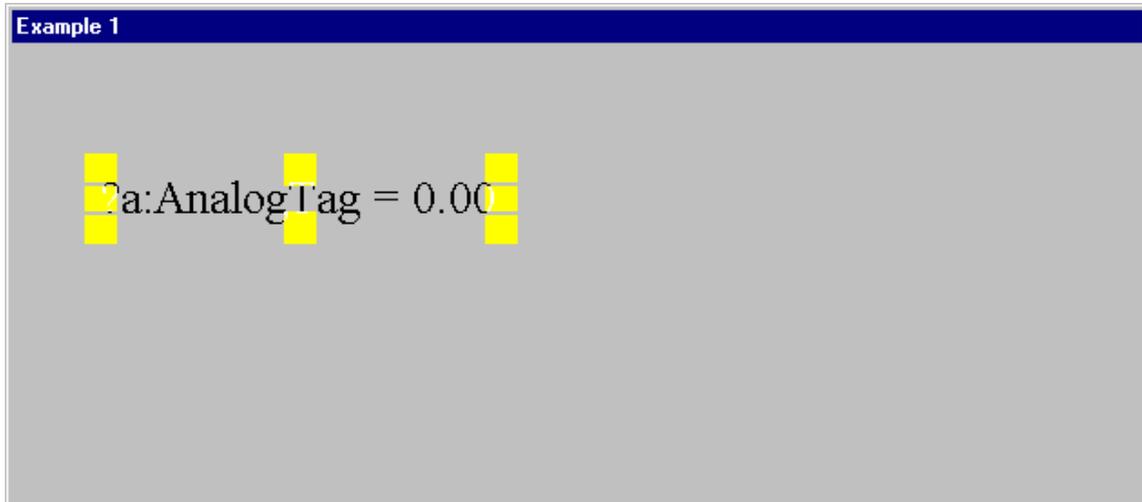


Double click on the temporary Tagname to bring up the following window. Change the temporary tagname to "Data_In", (previously entered into the SMSI Server). The Tagnames are case sensitive and must be entered in **exactly** as they were in the SMSI Server. Otherwise, communication between Intouch and the SMSI Server is impossible. Change the "Number Format" to an integer value by clicking on the pull down menu arrow and scrolling up to the single "0". Select it and click on OK.

A question will pop up asking if you want to define the Tagname "Data_In".  Click OK to
bring up the Tagname Dictionary.

**Tagname Dictionary**

○ Main   ● Details   ○ Alarms   ○ Details & Alarms   ○ Members

| New | Restore | Delete | Save | << | Select... | >> | Cancel | Close |

Tagname:  Data_In          Type: ...  Memory Real

Group: ...   $System          ○ Read only  ● Read Write

Comment:

☐ Log Data   ☐ Log Events          ☐ Retentive Value ☐ Retentive Parameters

Initial Value:  0          Eng Units:
Min Value:  -32768          Deadband:  0
Max Value:  32767          Log Deadband:  0

Click on the **Type** button to define the Tag Type.  Click on I/O Integer followed by **OK**.

**Tag Types**

☐ Memory Discrete
☐ I/O Discrete
☐ Indirect Discrete
☐ Memory Integer
☑ I/O Integer
☐ Memory Real
☐ I/O Real
☐ Indirect Analog
☐ Memory Message
☐ I/O Message
☐ Indirect Message
☐ Group Var
☐ Hist Trend
☐ Tag ID

| OK | Cancel | Details | Select All | Clear All |

Notice that additional information is now available at the bottom of the Tagname
Dictionary window.  The Access Name is used by Intouch to reference realtime I/O data
from a server such as the SMSI Server.



Click on the Access Name button and when the "Access Names" window appears, click
on "Add" to bring up the "Add Access Names" window. Type in "COM-DNS" as the
Access Name.  This name can be arbitrary, but try to make it meaningful by describing
what the Tagnames are linked to.  Fill in the Application Name with "smsiserver" and the
Topic Name with "board0" followed by OK.  Unlike the Tagnames, the Application
Name and the Topic Names are not case sensitive.

Back at the Access Names window click on **Close** to return to the "Tagname Dictionary" window.  Click on the checkbox, "Use Tagname as Item Name" and note that the Item is filled in automatically with the Tagname from above.

Click on Close to accept all the entries and to exit the Tagname Dictionary.  Now notice that the Tagname pre-assigned by Intouch to the Value Display now reflects the "Data_In" text.

**Example 1**

Data_In = 0
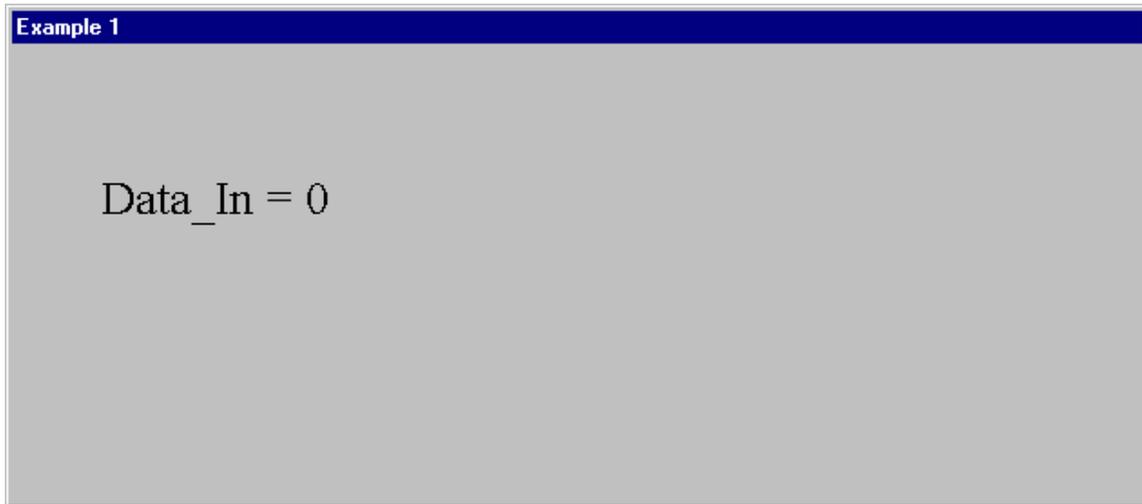
Create another Value Display as before but give it the Tagname "Data_Out".  The Access name in the Tagname Dictionary should already indicate COM-DNS, so no further setup is required.  When finished, the application window should look as follows.  From the File menu, select Save Window to save the application.  Now exit Window Maker.  The application can now be transferred to the Runtime Workstation using the techniques previously described.

A batch file must be set up to automatically launch the SMSI Server and WindowViewer.

**Example 1**

Data_In = 0                    Data_Out = 0

**350012(B)**