



## **Setup Examples**

---

# **NetArrays Signal Validation**

## NetArrays Signal Validation Example

© 2005, 2007, 2008, 2009 RTP Corporation

Not for reproduction in any printed or electronic media without express written consent from RTP Corp.

All information, data, graphics and statements in this document are proprietary intellectual property of RTP Corp. unless otherwise indicated and are to be considered RTP Corp. confidential. This intellectual property is made available solely for the direct use of potential or licensed RTP Corp. customers in their application of RTP Corp. products, and any other use or distribution is expressly prohibited. If you have received this publication in error, immediately delete, discard or return all copies to RTP Corp.

RTP Corporation  
1834 SW 2<sup>nd</sup> Street  
Pompano Beach, FL 33069  
Phone: (954) 974-5500  
Fax: (954) 975-9815  
Internet: <http://www.rtpcorp.com>

**File Name: Signal Validation Example.pdf**  
**Last Updated: 1/21/09**

## NetArrays Signal Validation

This example demonstrates the function of the NetArrays Signal Validation feature. The inputs from three redundant analog input cards will be configured in the Signal Validation display that is part of the I/O Configuration Form. The hardware used to complete this example includes three 3015/00 32-Channel Scanning Single Ended Analog Input cards installed in a Triple Redundant 3000 TMR Chassis.

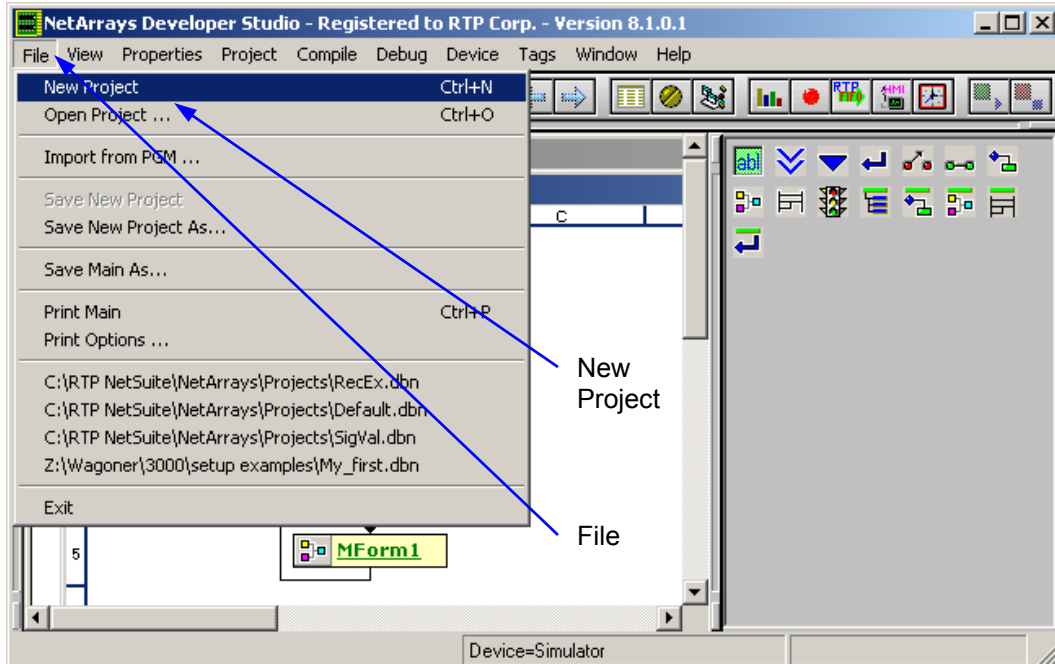
**Note:** This example is valid for any type of input I/O cards. The only condition is that each of the inputs measure the same physical condition. Redundant inputs may be installed on the same card, or on different cards in the same rack or on different cards in different racks.

No physical hardware is required for this example. The RTP NetSuite software package includes a pc based control engine simulator that permits the user to verify logic solving without the use of hardware.

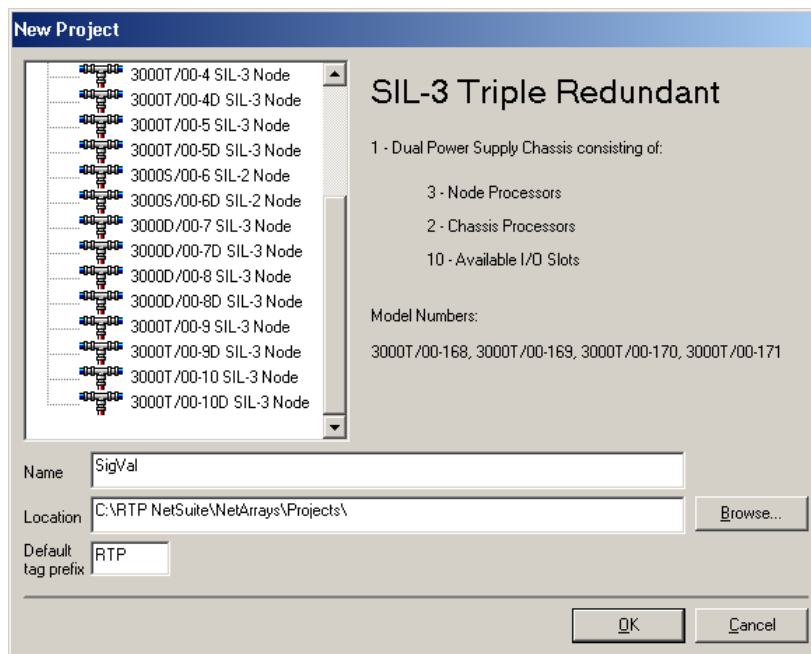
## Start NetArrays

### Start NetArrays

- On your PC, press  **Start** and select **Programs** ▶ **RTP NetSuite** ▶ **NetArrays**.
- Click **File**, then **New Project**.



- Select the **3000T/00-9D SIL-3 Node**
- Change the project name to SigVal
- Click OK.

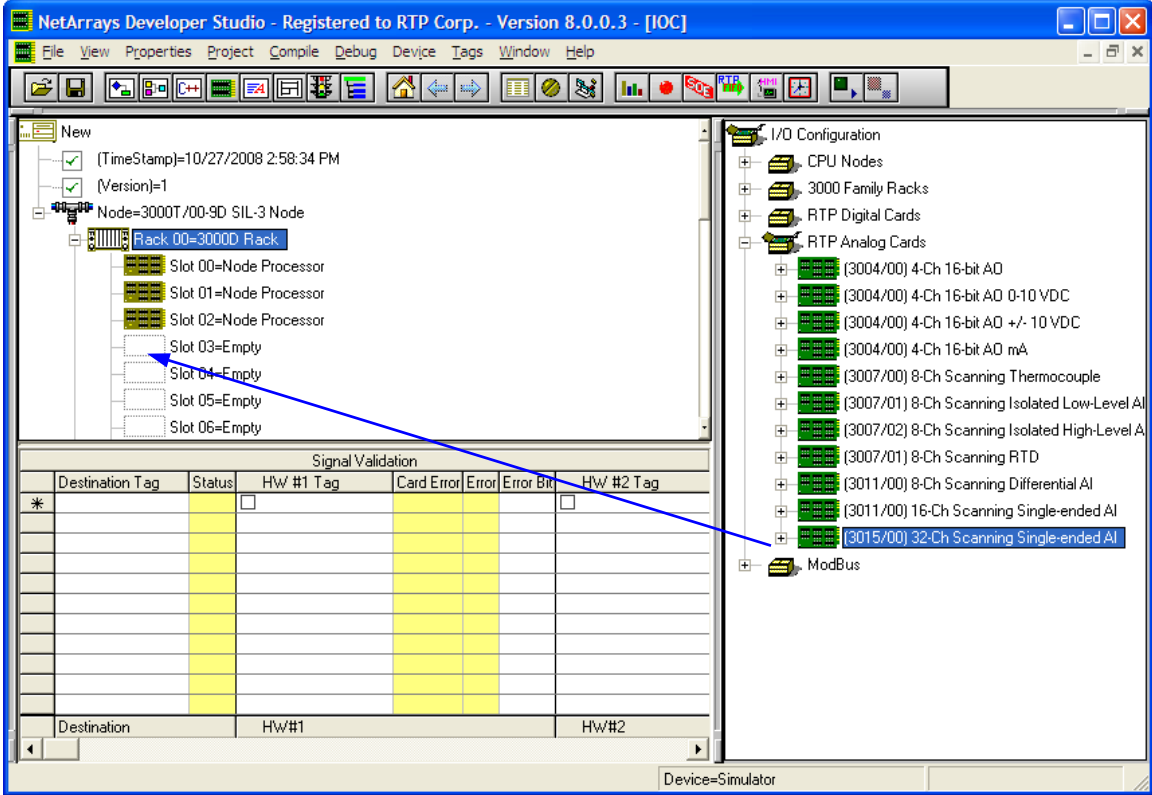




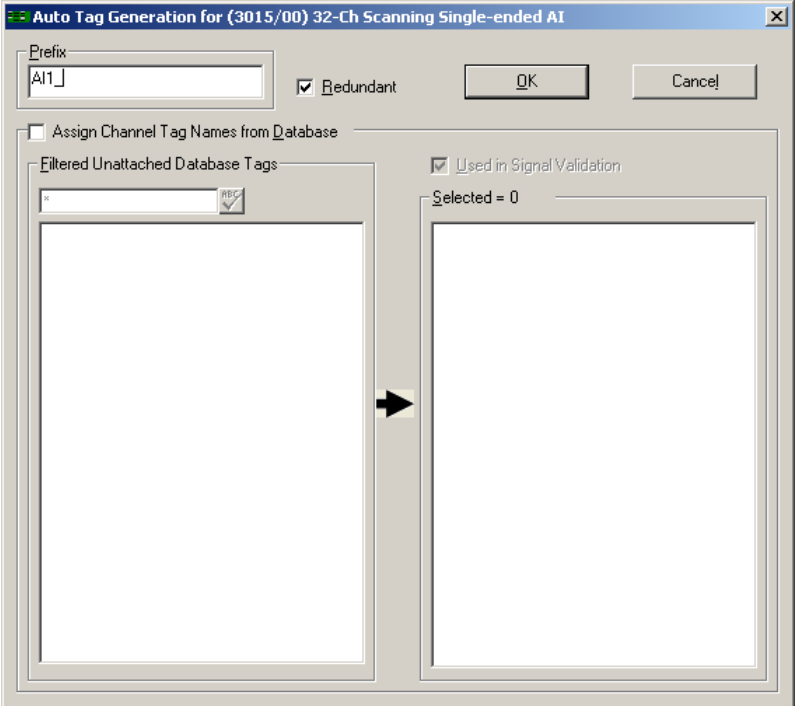
NetArrays Signal Validation Example

I/O Configuration Form

I/O Configuration Toolbox



- In the Auto Tag Generation window, enter the Prefix of **AI1\_** and select **Redundant**. Then click the **OK** button.

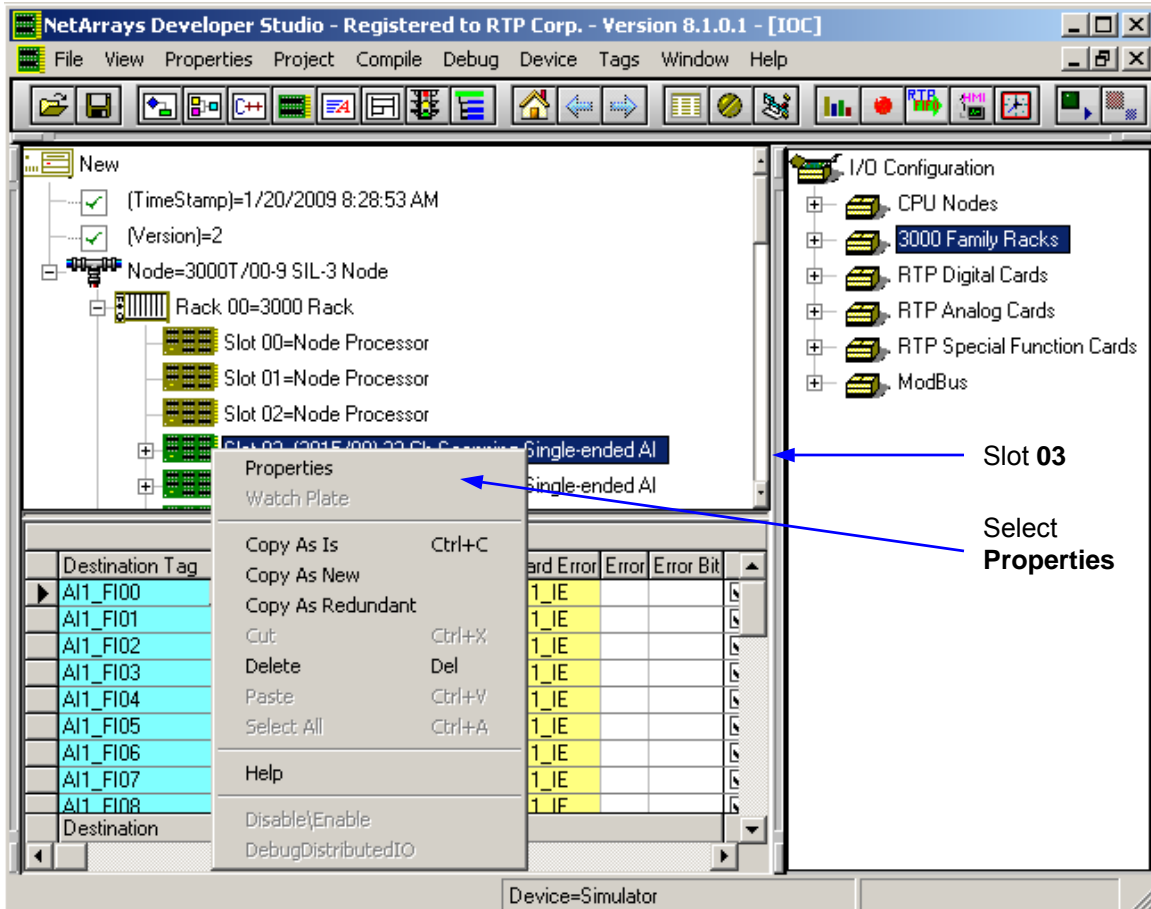




## NetArrays Signal Validation Example

### First I/O Card Property

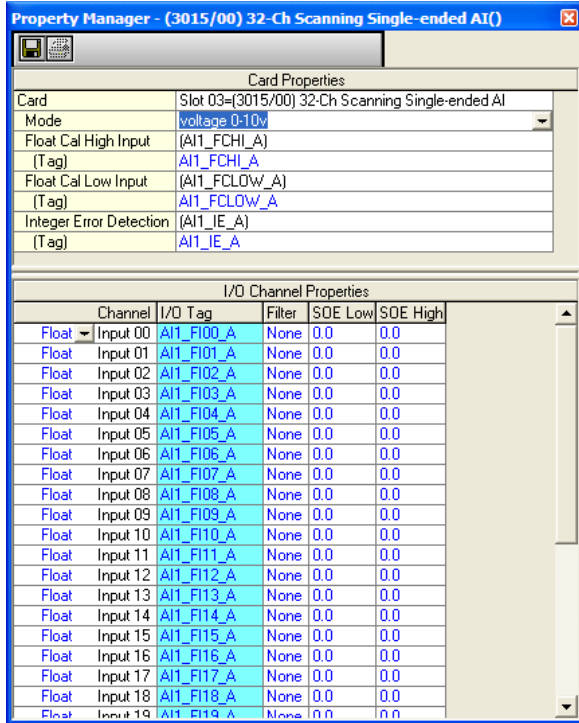
- Right-click on the **Slot 03** Analog Input card's icon to display the menu and select **Properties** to open the card's **Property Manager** window.




- Observe that the Postfix **A** has been added for each input parameter's tagname. The I/O card properties including diagnostics are listed at the top, and the I/O channel information is listed below.



## NetArrays Signal Validation Example



- Close the Property Manager window by clicking on .
- Similarly observe that the Redundant Analog Input cards in **Slot 04** and **Slot 05** have the Postfix **B** and **C** respectively.

### Signal Validation Configuration

Redundant inputs are voted using the signal validation table located in the IO ConfigurationForm.

- Click on the first Analog Input card's icon (in **Slot 03**) and drag it to the **HW#1** section of the Signal Validation display. Carefully position the cursor over the first **Tag** field and release the mouse button.
- After releasing the mouse button, NetArrays will fill in the Signal Validation display with the **HW#1** card's configured parameters as shown in the next figure.

## NetArrays Signal Validation Example

Node=3000T/00-9D SIL-3 Node  
Rack 00=3000D Rack  
Slot 00=Node Processor  
Slot 01=Node Processor  
Slot 02=Node Processor  
Slot 03=(3015/00) 32-Ch Scanning Single-ended AI  
Slot 04=(3015/00) 32-Ch Scanning Single-ended AI  
Slot 05=(3015/00) 32-Ch Scanning Single-ended AI  
Slot 06=Empty  
Slot 07=Empty  
Slot 08=Empty  
Slot 09=Empty  
Slot 10=Empty  
Slot 11=Empty

Drag and drop first AI card in **Slot 03** into first row of **HW#1** column.

Signal Validation													
Destination Tag	Status	Hw/ #1 Tag	Card Error	Error	Error Bit	Hw/ #2 Tag	Card Error	Error	Error Bit	Hw/ #3 Tag	Card Error	Error	Error Bit
AI1_FI00	AI1_FI00_S	AI1_FI00_A	AI1_IE_A										
AI1_FI01	AI1_FI01_S	AI1_FI01_A	AI1_IE_A										
AI1_FI02	AI1_FI02_S	AI1_FI02_A	AI1_IE_A										
AI1_FI03	AI1_FI03_S	AI1_FI03_A	AI1_IE_A										
AI1_FI04	AI1_FI04_S	AI1_FI04_A	AI1_IE_A										
AI1_FI05	AI1_FI05_S	AI1_FI05_A	AI1_IE_A										
AI1_FI06	AI1_FI06_S	AI1_FI06_A	AI1_IE_A										
AI1_FI07	AI1_FI07_S	AI1_FI07_A	AI1_IE_A										
AI1_FI08	AI1_FI08_S	AI1_FI08_A	AI1_IE_A										
AI1_FI09	AI1_FI09_S	AI1_FI09_A	AI1_IE_A										
AI1_FI10	AI1_FI10_S	AI1_FI10_A	AI1_IE_A										
AI1_FI11	AI1_FI11_S	AI1_FI11_A	AI1_IE_A										
AI1_FI12	AI1_FI12_S	AI1_FI12_A	AI1_IE_A										
AI1_FI13	AI1_FI13_S	AI1_FI13_A	AI1_IE_A										
AI1_FI14	AI1_FI14_S	AI1_FI14_A	AI1_IE_A										
AI1_FI15	AI1_FI15_S	AI1_FI15_A	AI1_IE_A										
AI1_FI16	AI1_FI16_S	AI1_FI16_A	AI1_IE_A										
AI1_FI17	AI1_FI17_S	AI1_FI17_A	AI1_IE_A										
Destination		HW#1				HW#2				HW#3			

- The Destination Variable fields are automatically populated without the Postfix. This **Destination Tag** variable stores the results of the voted inputs or the user configured default value. The destination **Status** variable reports the quality of the logical vote.
- Drag and drop the second redundant AI card from **Slot 04** into the **HW#2** column.
- Drag and drop the third redundant AI card from **Slot 05** into the **HW#3** column.
- For the channel with the Designation Tag **AI1\_FI00**, click in the **Type** field and then click on the ▼ to designate the signal validation algorithm for the channel.

Signal Validation															
Destination Tag	Status	Hw/ #1 Tag	Card Error	Error	Err	Hw/ #2 Tag	Card Error	Error	Err	Hw/ #3 Tag	Card Error	Error	Err	Defa Value	Type
AI1_FI00	AI	AI1_FI00_A	AI1_IE			AI1_FI00_B	AI1_IE			AI1_FI00_C	AI1_IE				3-2-1-0 / F / A
AI1_FI01	AI1	AI1_FI01_A	AI1_IE			AI1_FI01_B	AI1_IE			AI1_FI01_C	AI1_IE				3-2-1-0 / F / A
AI1_FI02	AI1	AI1_FI02_A	AI1_IE			AI1_FI02_B	AI1_IE			AI1_FI02_C	AI1_IE				3-2-1-0 / F / A
AI1_FI03	AI1	AI1_FI03_A	AI1_IE			AI1_FI03_B	AI1_IE			AI1_FI03_C	AI1_IE				3-2-1-0 / F / A
AI1_FI04	AI1	AI1_FI04_A	AI1_IE			AI1_FI04_B	AI1_IE			AI1_FI04_C	AI1_IE				3-2-1-0 / F / A
AI1_FI05	AI1	AI1_FI05_A	AI1_IE			AI1_FI05_B	AI1_IE			AI1_FI05_C	AI1_IE				3-2-1-0 / F / A
AI1_FI06	AI1	AI1_FI06_A	AI1_IE			AI1_FI06_B	AI1_IE			AI1_FI06_C	AI1_IE				3-2-1-0 / F / A
AI1_FI07	AI1	AI1_FI07_A	AI1_IE			AI1_FI07_B	AI1_IE			AI1_FI07_C	AI1_IE				3-2-1-0 / F / A
AI1_FI08	AI1	AI1_FI08_A	AI1_IE			AI1_FI08_B	AI1_IE			AI1_FI08_C	AI1_IE				3-2-1-0 / F / A
AI1_FI09	AI1	AI1_FI09_A	AI1_IE			AI1_FI09_B	AI1_IE			AI1_FI09_C	AI1_IE				3-2-1-0 / F / A
AI1_FI10	AI1	AI1_FI10_A	AI1_IE			AI1_FI10_B	AI1_IE			AI1_FI10_C	AI1_IE				3-2-1-0 / F / A
AI1_FI11	AI1	AI1_FI11_A	AI1_IE			AI1_FI11_B	AI1_IE			AI1_FI11_C	AI1_IE				3-2-1-0 / F / A
AI1_FI12	AI1	AI1_FI12_A	AI1_IE			AI1_FI12_B	AI1_IE			AI1_FI12_C	AI1_IE				3-2-1-0 / F / A
AI1_FI13	AI1	AI1_FI13_A	AI1_IE			AI1_FI13_B	AI1_IE			AI1_FI13_C	AI1_IE				3-2-1-0 / F / A
AI1_FI14	AI1	AI1_FI14_A	AI1_IE			AI1_FI14_B	AI1_IE			AI1_FI14_C	AI1_IE				3-2-1-0 / F / A
AI1_FI15	AI1	AI1_FI15_A	AI1_IE			AI1_FI15_B	AI1_IE			AI1_FI15_C	AI1_IE				3-2-1-0 / F / A
AI1_FI16	AI1	AI1_FI16_A	AI1_IE			AI1_FI16_B	AI1_IE			AI1_FI16_C	AI1_IE				3-2-1-0 / F / A
AI1_FI17	AI1	AI1_FI17_A	AI1_IE			AI1_FI17_B	AI1_IE			AI1_FI17_C	AI1_IE				3-2-1-0 / F / A
AI1_FI18	AI1	AI1_FI18_A	AI1_IE			AI1_FI18_B	AI1_IE			AI1_FI18_C	AI1_IE				3-2-1-0 / F / A
AI1_FI19	AI1	AI1_FI19_A	AI1_IE			AI1_FI19_B	AI1_IE			AI1_FI19_C	AI1_IE				3-2-1-0 / F / A

## NetArrays Signal Validation Example

### Signal validation options

First determine the Signal Validation scheme from the available algorithms:

Chose a Type:

Type 3-2-1-0: A single valid input may determine the logical vote.

Type 3-2-0: Minimum of 2 valid inputs must agree to determine the logical vote.

Choose an action if no valid inputs are available:

/F- Freeze the value of the logical destination tag.

/V- Use the specified value of the default tag for the logical destination tag.

Chose a calculation type. Use the following only for integer and float input tags:

/A- Average the analog input values.

/L- Select the lowest of the analog input values.

/H- Select the highest of the analog input values.


- Select **3-2-1-0 /V /A** for input channel **AI1\_F100**. This specifies: A single valid input may determine the logical vote. If no valid inputs are available, use the specified value of the default tag for the logical destination tag **AI1\_F100**. Average the analog input values.
- Enter value of **999** for the default value of **Default Tag AI1\_F100\_F**.
- Enter a value of **.1** for the delta value **Default Tag AI1\_F100\_D** in **Value**. This will exclude any input that is more than 0.1 from the median value.

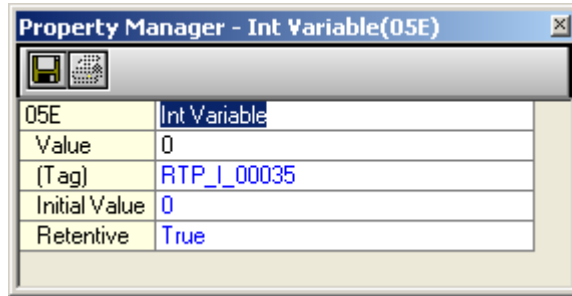
Signal Validation												
HW #2 Tag	Card Error	Error	Error Bit	HW #3 Tag	Card Error	Error	Error Bit	Default Tag	Value	Type	Delta Tag	Value
<input checked="" type="checkbox"/> AI1_F100_B	AI1_IE_B			<input checked="" type="checkbox"/> AI1_F100_C	AI1_IE_C			AI1_F100_F	999	3-2-1-0 /V /A	AI1_F100_D	.1
<input checked="" type="checkbox"/> AI1_F101_B	AI1_IE_B			<input checked="" type="checkbox"/> AI1_F101_C	AI1_IE_C					3-2-1-0 /F /A	AI1_F101_D	0
<input checked="" type="checkbox"/> AI1_F102_B	AI1_IE_B			<input checked="" type="checkbox"/> AI1_F102_C	AI1_IE_C					3-2-1-0 /F /A	AI1_F102_D	0
<input checked="" type="checkbox"/> AI1_F103_B	AI1_IE_B			<input checked="" type="checkbox"/> AI1_F103_C	AI1_IE_C					3-2-1-0 /F /A	AI1_F103_D	0
<input checked="" type="checkbox"/> AI1_F104_B	AI1_IE_B			<input checked="" type="checkbox"/> AI1_F104_C	AI1_IE_C					3-2-1-0 /F /A	AI1_F104_D	0
<input checked="" type="checkbox"/> AI1_F105_B	AI1_IE_B			<input checked="" type="checkbox"/> AI1_F105_C	AI1_IE_C					3-2-1-0 /F /A	AI1_F105_D	0
<input checked="" type="checkbox"/> AI1_F106_B	AI1_IE_B			<input checked="" type="checkbox"/> AI1_F106_C	AI1_IE_C					3-2-1-0 /F /A	AI1_F106_D	0
<input checked="" type="checkbox"/> AI1_F107_B	AI1_IE_B			<input checked="" type="checkbox"/> AI1_F107_C	AI1_IE_C					3-2-1-0 /F /A	AI1_F107_D	0
<input checked="" type="checkbox"/> AI1_F108_B	AI1_IE_B			<input checked="" type="checkbox"/> AI1_F108_C	AI1_IE_C					3-2-1-0 /F /A	AI1_F108_D	0
<input checked="" type="checkbox"/> AI1_F109_B	AI1_IE_B			<input checked="" type="checkbox"/> AI1_F109_C	AI1_IE_C					3-2-1-0 /F /A	AI1_F109_D	0
<input checked="" type="checkbox"/> AI1_F110_B	AI1_IE_B			<input checked="" type="checkbox"/> AI1_F110_C	AI1_IE_C					3-2-1-0 /F /A	AI1_F110_D	0
<input checked="" type="checkbox"/> AI1_F111_B	AI1_IE_B			<input checked="" type="checkbox"/> AI1_F111_C	AI1_IE_C					3-2-1-0 /F /A	AI1_F111_D	0
<input checked="" type="checkbox"/> AI1_F112_B	AI1_IE_B			<input checked="" type="checkbox"/> AI1_F112_C	AI1_IE_C					3-2-1-0 /F /A	AI1_F112_D	0
<input checked="" type="checkbox"/> AI1_F113_B	AI1_IE_B			<input checked="" type="checkbox"/> AI1_F113_C	AI1_IE_C					3-2-1-0 /F /A	AI1_F113_D	0
<input checked="" type="checkbox"/> AI1_F114_B	AI1_IE_B			<input checked="" type="checkbox"/> AI1_F114_C	AI1_IE_C					3-2-1-0 /F /A	AI1_F114_D	0
<input checked="" type="checkbox"/> AI1_F115_B	AI1_IE_B			<input checked="" type="checkbox"/> AI1_F115_C	AI1_IE_C					3-2-1-0 /F /A	AI1_F115_D	0
<input checked="" type="checkbox"/> AI1_F116_B	AI1_IE_B			<input checked="" type="checkbox"/> AI1_F116_C	AI1_IE_C					3-2-1-0 /F /A	AI1_F116_D	0
<input checked="" type="checkbox"/> AI1_F117_B	AI1_IE_B			<input checked="" type="checkbox"/> AI1_F117_C	AI1_IE_C					3-2-1-0 /F /A	AI1_F117_D	0
HW#2	<> HW #3			Default				Algorithm				

- That completes the I/O Configuration portion of this example.
- The next step is to create a Module Form that will demonstrate the function of the Signal Validation feature.

## Module Form “MForm1”

### Add Objects to the Module Form and Configure Their Properties

- Place the destination and status variables into Mform1 by selecting the input channels in the Signal Validation table. In the I/O Configuration Studio, highlight the first row in the Signal Validation table. Press and hold the shift key and select the last row within the Signal Validation table.
- Right click the mouse and choose **Copy As Is**.
- Return to the Main Form by clicking on the  button in the Main Toolbar.
- Double-click on the **MForm1** object in the Main Form to display the empty Module Form.
- In the top left corner of the Module Form, right click the mouse and select **Paste**.
- Drag and drop three integer variable objects into the form from the Object Toolbox. The Auto Tag Generator window will appear.

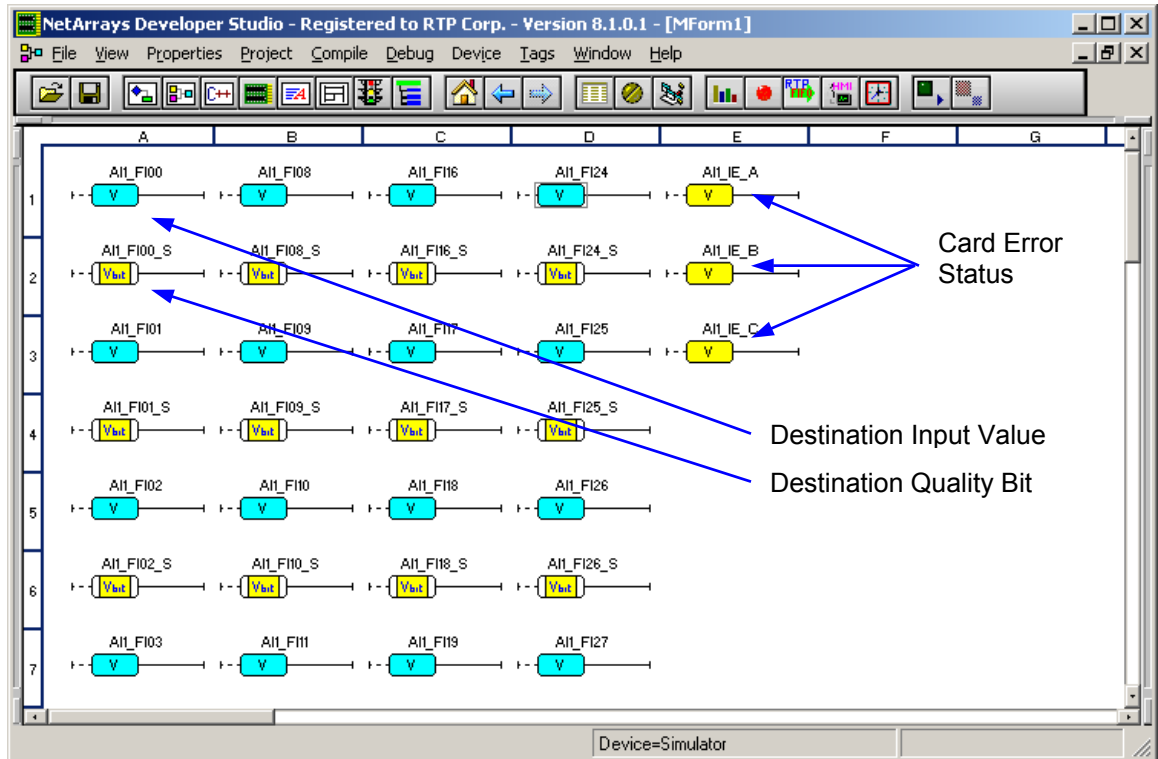


- For the objects, enter the tags for these variables as listed in the following table.

Cell	Object	Properties
E1	Int Variable	(Tag) = AI1_IE_A
E2	Int Variable	(Tag) = AI1_IE_B
E3	Int Variable	(Tag) = AI1_IE_C

- The finished MForm1 will look like the following picture.

## NetArrays Signal Validation Example



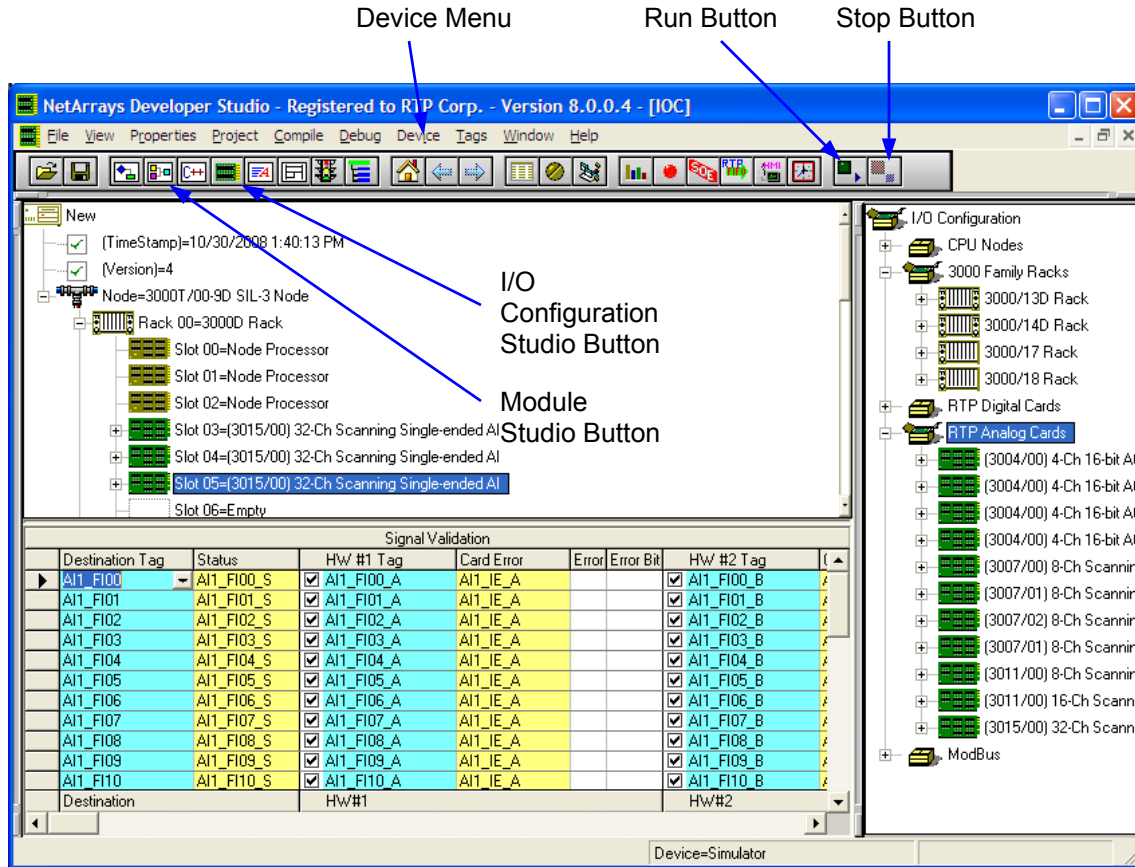
- In the NetArrays **File** menu, select **Save SigVal.dbn**.

The next step is to verify the project by downloading it to the target node and running it in Debug mode.

In this example, we will use the simulator as the device.

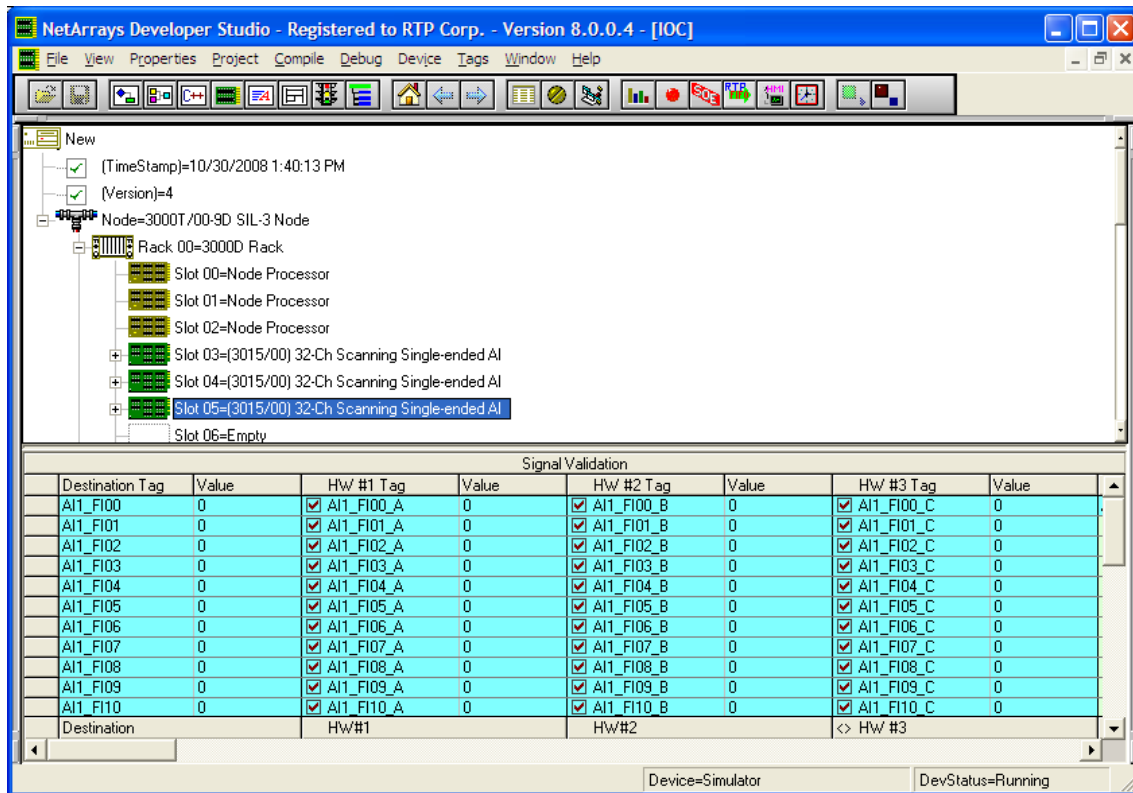
## Verification

Now that the Project Program is completed, we will compile it and download it to the simulator device for verification.

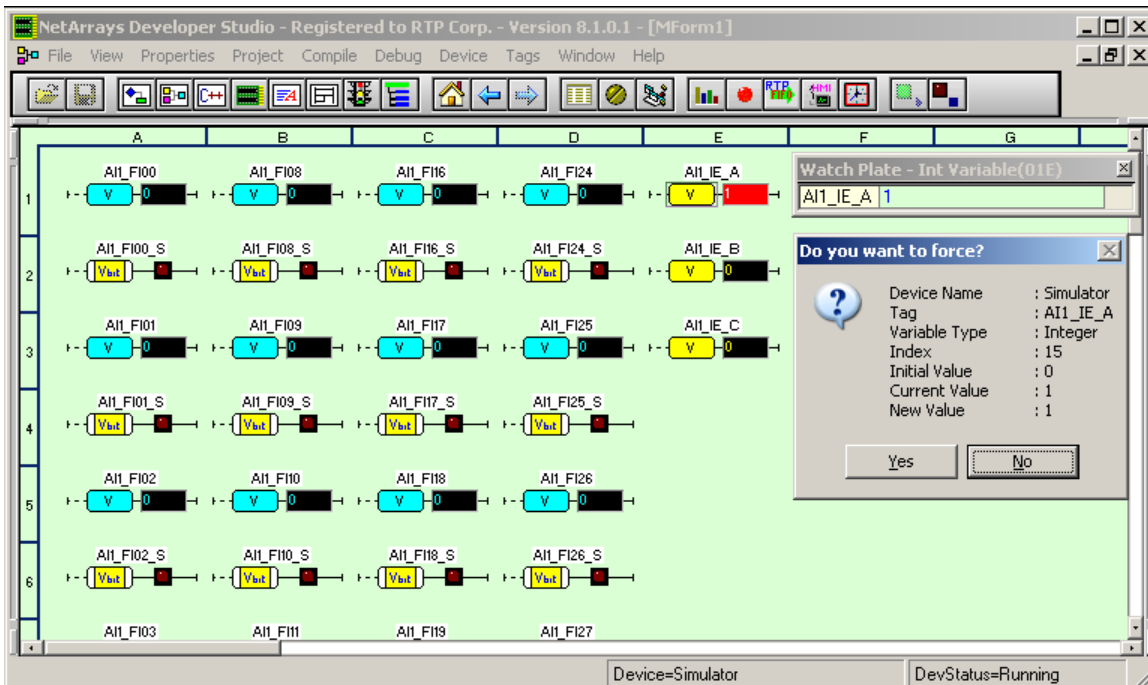


- Click on the **I/O Configuration Studio** button in the Main Toolbar to switch to the I/O Configuration Form.
- In the **Device** menu, select **Select ▶ Simulator**.
- Click on the **Run** button in the Main Toolbar. Enter "rtp" for the **Debug Password** and select **OK**. Enter "rtp" for the **Download Password** and select **OK**. Select **Yes** to the question, **Do you want to overwrite?** This will compile and download the Project Program to the target node. Wait until the download is complete.
- Observe the Signal Validation display. It should look like the one shown below:

## NetArrays Signal Validation Example

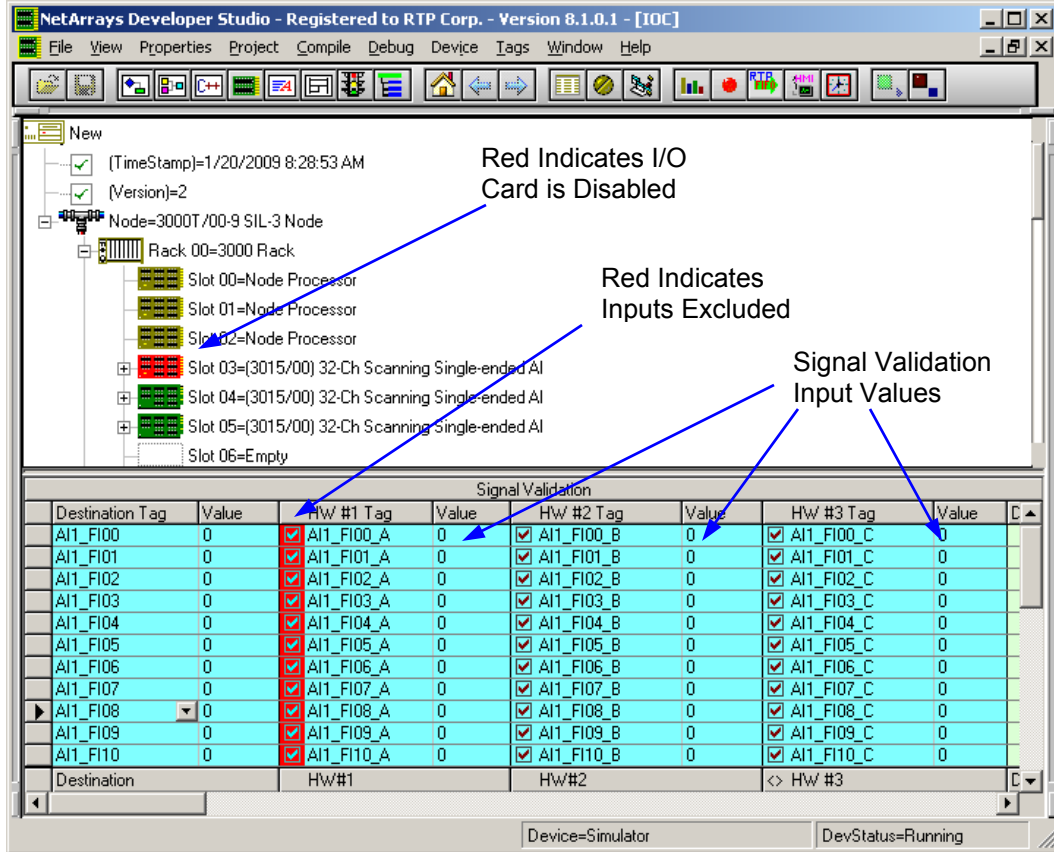


- Click on the **Module Studio** button in the Main Toolbar to switch to the Module Form **MForm1**.
- Click on the output of the integer variable **AI1\_IE\_A** and force the value to 1. Click on **Yes** to confirm the force. This causes the **HW#1** AI card to be Disabled.



### NetArrays Signal Validation Example

- Click on the **I/O Configuration Studio** button in the Main Toolbar to switch to the I/O Configuration Form. Observe that the card in slot **03** is **Red** (Disabled) and that the **HW1#** variables are **Red** (Excluded).



- Click on the **Module Studio** button in the Main Toolbar to switch to the Module Form **MForm1**. Force the integer variable **AI1\_IE\_B** to a value of **1**.
- Click on the **I/O Configuration Studio** button in the Main Toolbar to switch to the I/O Configuration Form. Observe that the card in slot **04** is Disabled and **HW#2** tags are Excluded.
- Click on the **Module Studio** button in the Main Toolbar to switch to the Module Form **MForm1**. Force **AI1\_IE\_C** to a value of **1**.
- Click on the **I/O Configuration Studio** button in the Main Toolbar to switch to the I/O Configuration Form. Observe that all 3 cards are Disabled and all inputs are Excluded. The default value of **999** is shown for **AI1\_FI00** in **Value** to the right of the **Destination Tag**.



### NetArrays Signal Validation Example

NetArrays Developer Studio - Registered to RTP Corp. - Version 8.1.0.1 - [IOC]

File View Properties Project Compile Debug Device Tags Window Help

New

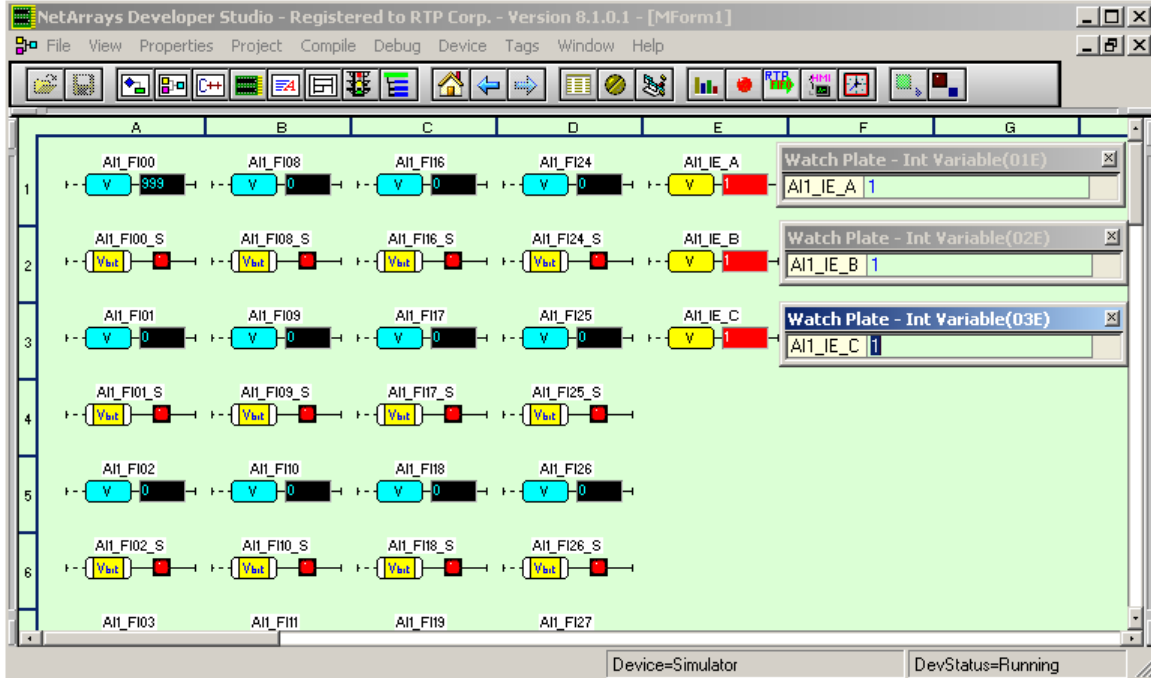
- (TimeStamp)=1/20/2009 8:28:53 AM
- (Version)=2
- Node=3000T/00-9 SIL-3 Node
  - Rack 00=3000 Rack
    - Slot 00=Node Processor
    - Slot 01=Node Processor
    - Slot 02=Node Processor
    - Slot 03=(3015/00) 32-Ch Scanning Single-ended AI
    - Slot 04=(3015/00) 32-Ch Scanning Single-ended AI
    - Slot 05=(3015/00) 32-Ch Scanning Single-ended AI

Signal Validation							
Destination Tag	Value	HW #1 Tag	Value	HW #2 Tag	Value	HW #3 Tag	Value
AI1_FI00	999	AI1_FI00_A	0	AI1_FI00_B	0	AI1_FI00_C	0
AI1_FI01	0	AI1_FI01_A	0	AI1_FI01_B	0	AI1_FI01_C	0
AI1_FI02	0	AI1_FI02_A	0	AI1_FI02_B	0	AI1_FI02_C	0
AI1_FI03	0	AI1_FI03_A	0	AI1_FI03_B	0	AI1_FI03_C	0
AI1_FI04	0	AI1_FI04_A	0	AI1_FI04_B	0	AI1_FI04_C	0
AI1_FI05	0	AI1_FI05_A	0	AI1_FI05_B	0	AI1_FI05_C	0
AI1_FI06	0	AI1_FI06_A	0	AI1_FI06_B	0	AI1_FI06_C	0
AI1_FI07	0	AI1_FI07_A	0	AI1_FI07_B	0	AI1_FI07_C	0
AI1_FI08	0	AI1_FI08_A	0	AI1_FI08_B	0	AI1_FI08_C	0
Destination		HW#1		HW#2		HW #3	

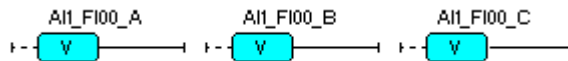
Device=Simulator      DevStatus=Running

- Click on the **Module Studio** button in the Main Toolbar to switch to the Module Form **MForm1**. Observe that the default value of **999** is displayed for **AI1\_FI00**. Also observe that the destination bit status variables **AI1\_FI00\_S** through **AI1\_FI31\_S** are **True** indicating bad quality.

## NetArrays Signal Validation Example



- 
- To remove the forced values, change the value of integer variables **AI1\_IE\_A**, **AI1\_IE\_B**, and **AI1\_IE\_C** back to **zero** and close each watch plate. Each time click **Yes** to remove the force. All card and variable indicators will return to their normal conditions.
- Analog input values may be simulated just as easily by adding additional float variables on **Mform1**, assigning them to AI channels and forcing them in the same manner. For example, channel 00 of each input card would be indicated as shown.



**Congratulations!** You have successfully configured the NetArrays Signal Validation feature and tested it by running your NetArrays Project Program in the simulator.