Fieldbus

NI-FBUS™ Configurator User Manual

NATIONAL INSTRUMENTS™

July 2000 Edition
Part Number 321423C-01

Worldwide Technical Support and Product Information

ni.com

National Instruments Corporate Headquarters

11500 North Mopac Expressway   Austin, Texas 78759-3504   USA   Tel: 512 794 0100

Worldwide Offices

Australia 03 9879 5166, Austria 0662 45 79 90 0, Belgium 02 757 00 20, Brazil 011 284 5011,
Canada (Calgary) 403 274 9391, Canada (Ontario) 905 785 0085, Canada (Québec) 514 694 8521,
China 0755 3904939, Denmark 45 76 26 00, Finland 09 725 725 11, France 01 48 14 24 24,
Germany 089 741 31 30, Greece 30 1 42 96 427, Hong Kong 2645 3186, India 91805275406,
Israel 03 6120092, Italy 02 413091, Japan 03 5472 2970, Korea 02 596 7456, Mexico (D.F.) 5 280 7625,
Mexico (Monterrey) 8 357 7695, Netherlands 0348 433466, New Zealand 09 914 0488, Norway 32 27 73 00,
Poland 0 22 528 94 06, Portugal 351 1 726 9011, Singapore 2265886, Spain 91 640 0085,
Sweden 08 587 895 00, Switzerland 056 200 51 51, Taiwan 02 2528 7227, United Kingdom 01635 523545

For further support information, see the *Technical Support Resources* appendix. To comment on the documentation, send e-mail to techpubs@ni.com

Important Information

Warranty

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this document is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of National Instruments Corporation.

Trademarks

FieldPoint™, Lookout™, National Instruments™, NI-FBUS™, and ni.com™, are trademarks of National Instruments Corporation.

Product and company names mentioned herein are trademarks or trade names of their respective companies.

WARNING REGARDING USE OF NATIONAL INSTRUMENTS PRODUCTS

(1) NATIONAL INSTRUMENTS PRODUCTS ARE NOT DESIGNED WITH COMPONENTS AND TESTING FOR A LEVEL OF RELIABILITY SUITABLE FOR USE IN OR IN CONNECTION WITH SURGICAL IMPLANTS OR AS CRITICAL COMPONENTS IN ANY LIFE SUPPORT SYSTEMS WHOSE FAILURE TO PERFORM CAN REASONABLY BE EXPECTED TO CAUSE SIGNIFICANT INJURY TO A HUMAN.

(2) IN ANY APPLICATION, INCLUDING THE ABOVE, RELIABILITY OF OPERATION OF THE SOFTWARE PRODUCTS CAN BE IMPAIRED BY ADVERSE FACTORS, INCLUDING BUT NOT LIMITED TO FLUCTUATIONS IN ELECTRICAL POWER SUPPLY, COMPUTER HARDWARE MALFUNCTIONS, COMPUTER OPERATING SYSTEM SOFTWARE FITNESS, FITNESS OF COMPILERS AND DEVELOPMENT SOFTWARE USED TO DEVELOP AN APPLICATION, INSTALLATION ERRORS, SOFTWARE AND HARDWARE COMPATIBILITY PROBLEMS, MALFUNCTIONS OR FAILURES OF ELECTRONIC MONITORING OR CONTROL DEVICES, TRANSIENT FAILURES OF ELECTRONIC SYSTEMS (HARDWARE AND/OR SOFTWARE), UNANTICIPATED USES OR MISUSES, OR ERRORS ON THE PART OF THE USER OR APPLICATIONS DESIGNER (ADVERSE FACTORS SUCH AS THESE ARE HEREAFTER COLLECTIVELY TERMED "SYSTEM FAILURES"). ANY APPLICATION WHERE A SYSTEM FAILURE WOULD CREATE A RISK OF HARM TO PROPERTY OR PERSONS (INCLUDING THE RISK OF BODILY INJURY AND DEATH) SHOULD NOT BE RELIANT SOLELY UPON ONE FORM OF ELECTRONIC SYSTEM DUE TO THE RISK OF SYSTEM FAILURE. TO AVOID DAMAGE, INJURY, OR DEATH, THE USER OR APPLICATION DESIGNER MUST TAKE REASONABLY PRUDENT STEPS TO PROTECT AGAINST SYSTEM FAILURES, INCLUDING BUT NOT LIMITED TO BACK-UP OR SHUT DOWN MECHANISMS. BECAUSE EACH END-USER SYSTEM IS CUSTOMIZED AND DIFFERS FROM NATIONAL INSTRUMENTS' TESTING PLATFORMS AND BECAUSE A USER OR APPLICATION DESIGNER MAY USE NATIONAL INSTRUMENTS PRODUCTS IN COMBINATION WITH OTHER PRODUCTS IN A MANNER NOT EVALUATED OR CONTEMPLATED BY NATIONAL INSTRUMENTS, THE USER OR APPLICATION DESIGNER IS ULTIMATELY RESPONSIBLE FOR VERIFYING AND VALIDATING THE SUITABILITY OF NATIONAL INSTRUMENTS PRODUCTS WHENEVER NATIONAL INSTRUMENTS PRODUCTS ARE INCORPORATED IN A SYSTEM OR APPLICATION, INCLUDING, WITHOUT LIMITATION, THE APPROPRIATE DESIGN, PROCESS AND SAFETY LEVEL OF SUCH SYSTEM OR APPLICATION.

Conventions

The following conventions are used in this manual:

»

The » symbol leads you through nested menu items and dialog box options to a final action. The sequence **File»Page Setup»Options** directs you to pull down the **File** menu, select the **Page Setup** item, and select **Options** from the last dialog box.

💡

This icon denotes a tip, which alerts you to advisory information.

📝

This icon denotes a note, which alerts you to important information.

⚠

This icon denotes a caution, which advises you of precautions to take to avoid injury, data loss, or a system crash.

bold

Bold text denotes items that you must select or click on in the software, such as menu items and dialog box options. Bold text also denotes parameter names.

Fieldbus

The generic term *fieldbus* refers to any bus that connects to field devices. This includes Foundation Fieldbus, CAN, DNET, and Profibus. In this manual, the term *Fieldbus* refers specifically to the Foundation Fieldbus.

italic

Italic text denotes variables, emphasis, a cross reference, or an introduction to a key concept. This font also denotes text that is a placeholder for a word or value that you must supply.

`monospace`

Text in this font denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames and extensions, and code excerpts.

`monospace bold`

Bold text in this font denotes the messages and responses that the computer automatically prints to the screen. This font also emphasizes lines of code that are different from the other examples.

`monospace italic`

Italic text in this font denotes text that is a placeholder for a word or value that you must supply.

Contents

Chapter 1
NI-FBUS Configurator Overview

Chapter 2
Using the NI-FBUS Configurator

Chapter 3
NI-FBUS Configurator Tutorials

Chapter 4
Parameters

Appendix A
Error Messages and Warnings

Appendix B
Troubleshooting and Common Questions

Appendix C
Cascade and Remote Cascade Setups

Appendix D
NI-FBUS Dialog Utility

Appendix E
Technical Support Resources

Glossary

Index

1

NI-FBUS Configurator Overview

This chapter introduces the NI-FBUS Configurator, lists some of its main features, and describes the main windows of the application.

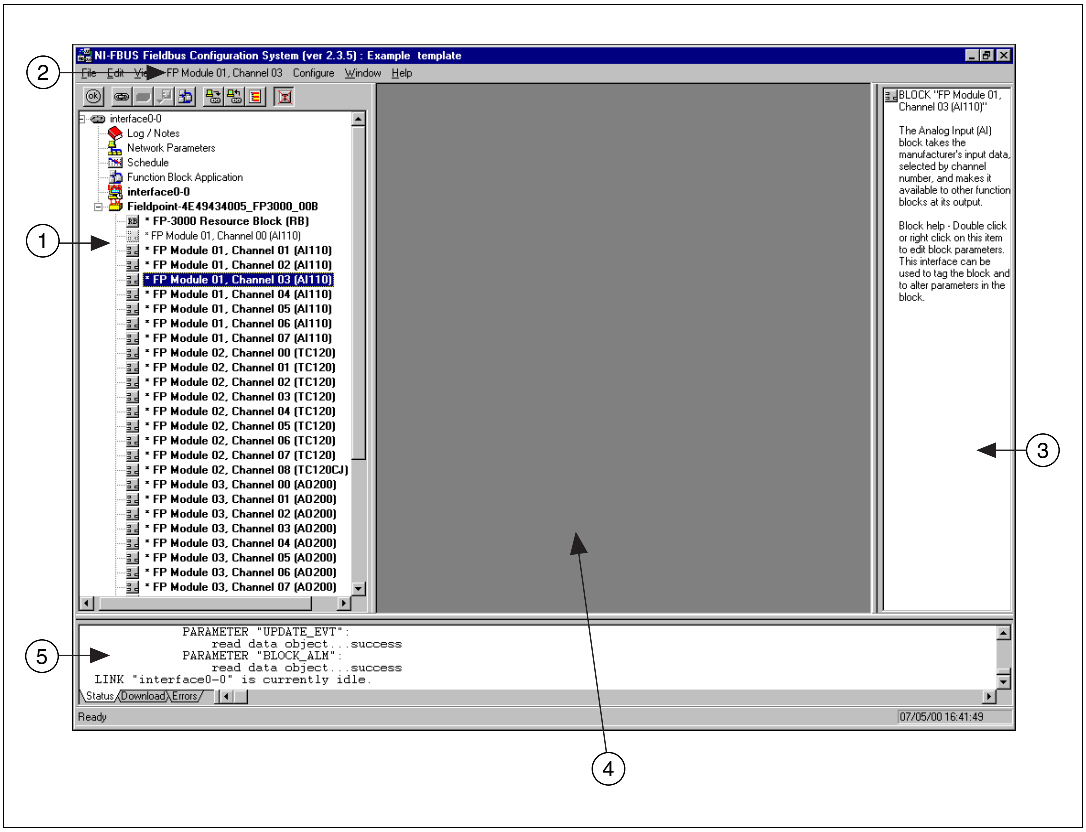
This manual assumes that you are already familiar with Windows 2000/NT/98/95 and have read the *FOUNDATION Fieldbus Overview* document or are otherwise familiar with using FOUNDATION Fieldbus.

Introduction to the NI-FBUS Configurator

You can use the NI-FBUS Configurator to configure a FOUNDATION Fieldbus network and keep track of your configuration changes. The NI-FBUS Configurator is a graphical environment for creating linkages, loops, and a schedule based on the concepts described in the *FOUNDATION Fieldbus Overview* document.

NI-FBUS Configurator Windows

The NI-FBUS Configurator has three resizable windows within the main window: the configuration tree, help window, and status window. Additional specialty windows can be opened in the middle frame to configure your function block applications, change parameters, and update the schedule. You can also open separate windows for block parameter adjustment on top of the main window. The following figure shows elements of the NI-FBUS Configurator main window.


NI-FBUS Fieldbus Configuration System (ver 2.3.5) : Example template
File Edit View FP Module 01, Channel 03 Configure Window Help
interface0-0
Log / Notes
Network Parameters
Schedule
Function Block Application
interface0-0
Fieldpoint-4E49434005_FP3000_00B
* FP-3000 Resource Block (RB)
* FP Module 01, Channel 00 (AI110)
* FP Module 01, Channel 01 (AI110)
* FP Module 01, Channel 02 (AI110)
* FP Module 01, Channel 03 (AI110)
* FP Module 01, Channel 04 (AI110)
* FP Module 01, Channel 05 (AI110)
* FP Module 01, Channel 06 (AI110)
* FP Module 01, Channel 07 (AI110)
* FP Module 02, Channel 00 (TC120)
* FP Module 02, Channel 01 (TC120)
* FP Module 02, Channel 02 (TC120)
* FP Module 02, Channel 03 (TC120)
* FP Module 02, Channel 04 (TC120)
* FP Module 02, Channel 05 (TC120)
* FP Module 02, Channel 06 (TC120)
* FP Module 02, Channel 07 (TC120)
* FP Module 02, Channel 08 (TC120CJ)
* FP Module 03, Channel 00 (AO200)
* FP Module 03, Channel 01 (AO200)
* FP Module 03, Channel 02 (AO200)
* FP Module 03, Channel 03 (AO200)
* FP Module 03, Channel 04 (AO200)
* FP Module 03, Channel 05 (AO200)
* FP Module 03, Channel 06 (AO200)
* FP Module 03, Channel 07 (AO200)
Status Download Errors
Ready
07/05/00 16:41:49
1
2
3
4
5

BLOCK "FP Module 01, Channel 03 (AI110)"

The Analog Input (AI) block takes the manufacturer's input data, selected by channel number, and makes it available to other function blocks at its output.

Block help - Double click or right click on this item to edit block parameters. This interface can be used to tag the block and to alter parameters in the block.

```
            PARAMETER "UPDATE_EVT":
                read data object...success
            PARAMETER "BLOCK_ALM":
                read data object...success
LINK "interface0-0" is currently idle.
```

1	Configuration tree	3	Help window
2	Object menu (changes based on object selected in configuration tree)	4	Middle frame
		5	Status window

At the upper-right corner of the screen, you may see two sets of resize buttons. The outer set of buttons controls the NI-FBUS Configurator, and the inner set controls whatever window is currently active in the middle frame. To see other windows in the middle frame, you may need to minimize the active window in the middle frame.

Configuration Tree

The configuration tree, which appears to the left of the middle frame, displays the configurable objects of the link(s) connected to the NI-FBUS Configurator. When you select an object in the configuration tree, its own menu appears as the main menu bar Object menu item. You can also view the items on this menu by right-clicking on the object.

To configure or view an object's settings, double-click on its icon in the configuration tree.

You can click on the **Show/Hide Transducers and Device IDs** button to toggle between showing or hiding the transducer blocks and device IDs in the configuration tree.

During the initial scan of the bus, the following state icons sometimes overlap the configuration tree icons:

- Updating—This icon appears when the NI-FBUS Configurator is reading or writing to the object.
- Error—This icon appears when the NI-FBUS Configurator detects or encounters an error with the object.
- Invalid—This icon appears when the indicated object is not responding to the NI-FBUS Configurator scan.

Note It is normal for objects to go invalid briefly during the initial scan of the bus or when changing the device's address. This is because the device suspends communication while setting its address.

Help Window

The help window, which appears to the right of the middle frame, displays help information. To view help information for a particular topic, move the mouse cursor over the relevant object or parameter. Some objects require you to click on the object to view the help. To toggle between showing or hiding the help window, select **Window»Help Window**.

Status Window

The status window, which appears below the middle frame, displays the current status of the NI-FBUS Configurator. The status window contains three tabs. To toggle between showing or hiding the status window, select **Window»Status Window**.

Status Tab

The **Status** tab shows live updates of what the NI-FBUS Configurator does on the Fieldbus. The **Status** tab automatically appears on top when you start the NI-FBUS Configurator. The **Status** tab displays the steps that the NI-FBUS Configurator takes to accomplish a task when you read, write, scan, or download data to the bus.

Download Tab

The **Download** tab shows all the status information for the last configuration download to the bus. The NI-FBUS Configurator retains this information until the next download. The **Download** tab automatically appears on top when you begin to download your configuration. The status information on the **Download** tab can also be read in the log/notes window. The **Download** tab is updated only when a configuration is downloaded to the bus.

Errors Tab

The Errors tab displays all recent I/O errors, project configuration errors, and warnings in the current project. To fix an error, double-click on the error to open the window showing the cause of the error. For a list of possible errors, refer to Appendix A, *Error Messages and Warnings*.

Use the drop-down list attached to the upper-left corner of the Errors tab to filter the types of errors shown. The **All Errors** selection shows all I/O and project errors, and the **Project Errors** selection shows only project errors.

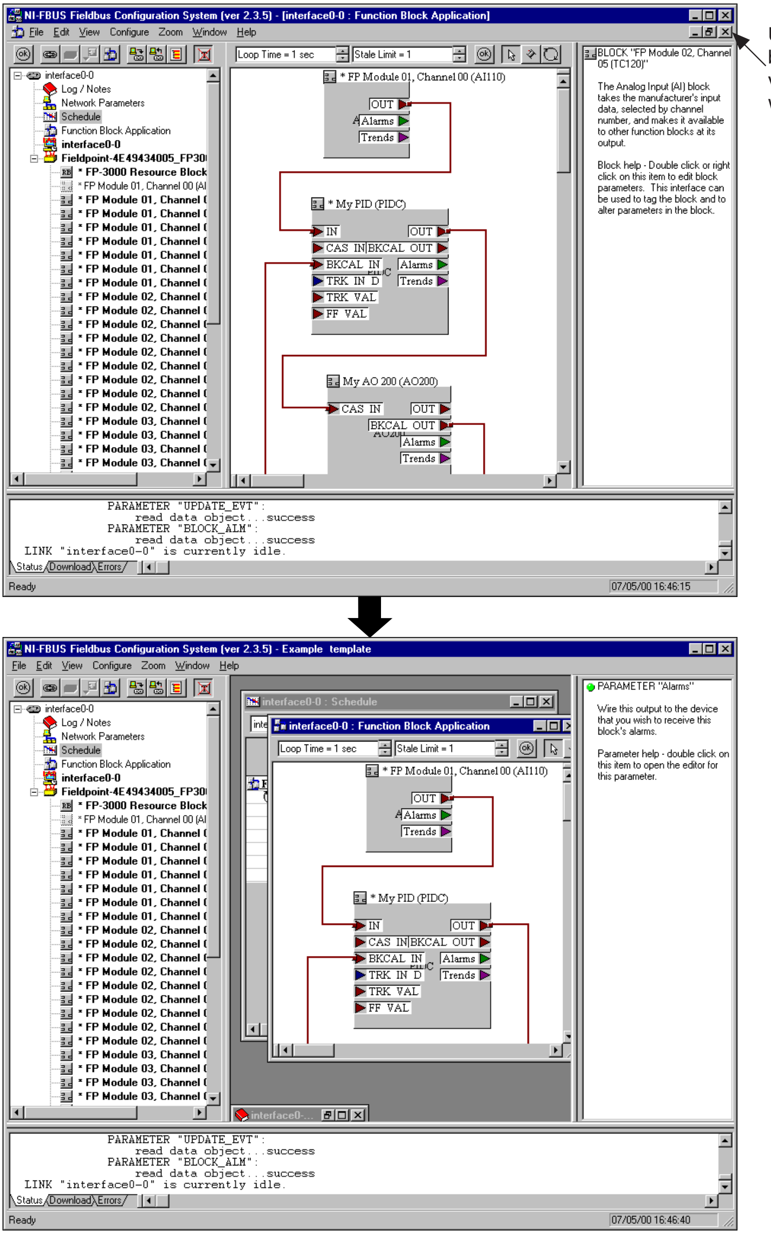
When you download your configuration, the NI-FBUS Configurator checks for errors in your project. If you have a project error, the NI-FBUS Configurator warns you and allows you to cancel the download.

The NI-FBUS Configurator automatically regenerates error information each time you change the project. However, you can force the NI-FBUS Configurator to revalidate the project by clicking on the **Check Project for Errors** toolbar button.

Middle Frame

The contents of the middle frame are determined by what you choose to open there. Double-clicking on certain items in the configuration tree causes a corresponding window to open in the middle frame. The items in the configuration tree that open a window in the middle frame are Function Block Application, Schedule, Log/Notes, or any device.

Double-clicking on a second item in the configuration tree opens another window on top of the first window in the middle frame. To retrieve the first window, you should minimize, restore, or close the second window using the resize buttons shown in the following figure.



BLOCK "FP Module 02, Channel 05 (TC120)"

The Analog Input (AI) block takes the manufacturer's input data, selected by channel number, and makes it available to other function blocks at its output.

Block help - Double click or right click on this item to edit block parameters. This interface can be used to tag the block and to alter parameters in the block.

```
            PARAMETER "UPDATE_EVT":
                read data object...success
            PARAMETER "BLOCK_ALM":
                read data object...success
   LINK "interface0-0" is currently idle.
```

PARAMETER "Alarms"

Wire this output to the device that you wish to receive this block's alarms.

Parameter help - double click on this item to open the editor for this parameter.

```
            PARAMETER "UPDATE_EVT":
                read data object...success
            PARAMETER "BLOCK_ALM":
                read data object...success
   LINK "interface0-0" is currently idle.
```

Use these buttons to view additional windows

Function Block Application Editor Window

The Function Block Application Editor is the graphical interface you use to create your FOUNDATION Fieldbus control strategy (also known as the block diagram) in the NI-FBUS Configurator.

You can open the Function Block Application Editor by double-clicking on the **Function Block Application** icon in the configuration tree. The Function Block Application Editor appears in the middle frame of the NI-FBUS Configurator. You can minimize and maximize this window using the resize buttons in the upper-right corner. When this window is maximized, its resize buttons appear under the main window resize buttons, above the help window.

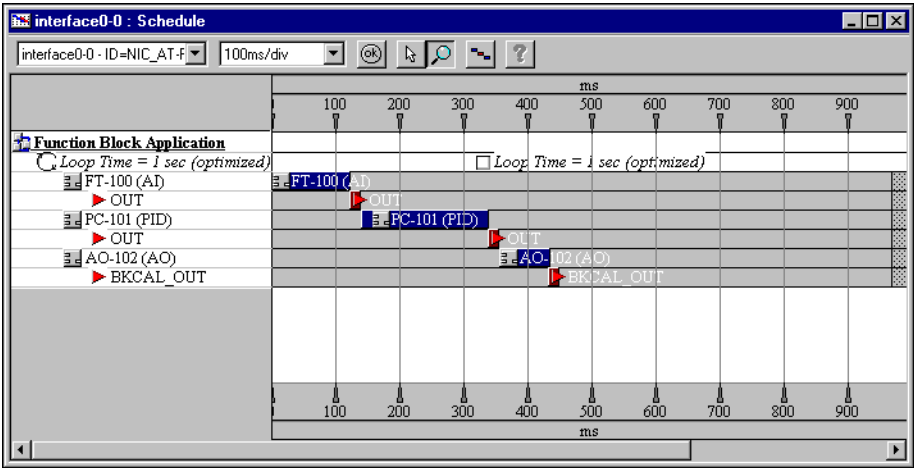
To connect blocks, use the Wiring tool. For help using this tool, see *Wiring Blocks Manually* section in Chapter 2, *Using the NI-FBUS Configurator*.

To switch between the pointer, wiring, and loop tools in the Function Block Application Editor window, press <Tab>. To switch between the pointer and the wiring tool, press the spacebar.

You can drag and drop block or device icons from one window into another. This is especially helpful when dragging function blocks from the configuration tree into the Function Block Application Editor window.

Schedule Window

The NI-FBUS Configurator combines the link active schedule and function block schedules into one diagram, as shown in the following figure.



To view or edit the link active schedule, double-click on the **Schedule** icon in the configuration tree. The schedule window appears in the middle frame.

In the schedule window, a key to the scheduled objects appears on the left. The timing diagram on the right displays the execution times of the function blocks, and when data is transmitted on the bus. The blue bars on the screen correspond to the function block execution times and the red bars correspond to the data transmission times, which consist of LAS notification time and transmission time.

Tip With only one device, you will not see data transmission since there are no parameters that need to be transmitted to other devices. Also, for some fast devices (such as the FP-3000), the blue block execution bars are so short that they are entirely hidden behind the block icons themselves.

Note An optimized schedule can show that the transmission time overlaps the end of function block execution. This is because the transmission time includes the time that the LAS uses to tell the device to publish the data.

To change the execution time for a loop, click on the execution time next to the loop icon and type in the new time.

Log/Notes Window

The log/notes window keeps track of the last downloaded configuration status output. To open the log/notes window, double-click on the **Log/Notes** icon in the configuration tree. The log/notes window appears in the middle frame.

Device Window

The device window shows the device ID and address of the host device or field device you select. From this window, you can view and change advanced parameters. You can view the values in hex or decimal representation.

You can open the device window in the following ways:

- Double-click on the host icon to open the device window for a host device, which is the device that hosts the NI-FBUS Configurator (the PC).

- Double-click on the device icon to open the device window for a field device. The icon is followed by the device's tag and its unique serial identifier.

Network Parameters Window

The network parameters window lets you change the low-level (advanced) configuration parameters for the link. To open the network parameters window, double-click on the **Network Parameters** icon in the configuration tree. The network parameters window appears floating over the main window.

Caution Do *not* modify these parameters without good reason. If you must modify parameters for certain devices, the device manufacturer will recommend settings. Modifying these parameters can have an adverse affect on data throughput rates. If settings are incorrectly modified, some devices may disappear off the bus.

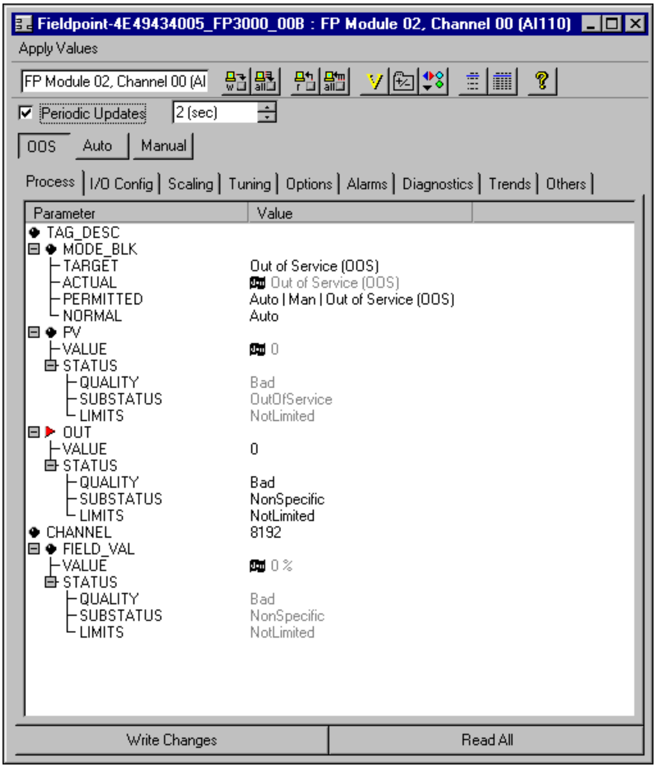
Block Configuration Window

Each function block in your application has a block configuration window that you can use to change block parameters and other settings. When you read or write data in the block configuration window, the **Status** tab in the status window shows the status of the transactions.

To open the block configuration window for a block, do one of the following:

- Double-click on the name of the function block under the appropriate device in the configuration tree.
- Double-click on a function block in the Function Block Application Editor window.

The block configuration window appears floating over the main window. The following figure shows a block configuration window.



☑ Periodic Updates 2 (sec)

Parameter	Value
TAG_DESC	
MODE_BLK	
TARGET	Out of Service (OOS)
ACTUAL	Out of Service (OOS)
PERMITTED	Auto | Man | Out of Service (OOS)
NORMAL	Auto
PV	
VALUE	0
STATUS	
QUALITY	Bad
SUBSTATUS	OutOfService
LIMITS	NotLimited
OUT	
VALUE	0
STATUS	
QUALITY	Bad
SUBSTATUS	NonSpecific
LIMITS	NotLimited
CHANNEL	8192
FIELD_VAL	
VALUE	0 %
STATUS	
QUALITY	Bad
SUBSTATUS	NonSpecific
LIMITS	NotLimited

Changing Parameter Settings

You can change parameter settings either by clicking on the desired field and typing the new value or by selecting the new value from a drop-down menu, if provided. For more information, refer to *Editing Block Parameters* section in Chapter 2, *Using the NI-FBUS Configurator*.

Yellow highlighting in the block configuration window indicates that a parameter's setting has been changed and does not match the parameter value stored in the device.

Block Configuration Window Icons

As shown in the previous figure, an icon to the left of each parameter name color-codes and symbolically represents the class of the parameter. Table 1-1 describes the classes of the parameters.

Table 1-1. Block Configuration Window Icons

Shape	Color	Description
Circle	Green—Alarms	Represents contained parameters. Contained parameters cannot be linked to other parameters; they are contained in the block.
	Light blue—Tuning	
	Black—Others	
Right arrow	Red	Represents output that can be sent to another block.
Left arrow	Blue	Represents input that can be obtained from another block.

Block Configuration Window Tabs

The block configuration window groups the parameters by functional category. For example, parameters related to alarms appear on the **Alarms** tab, and parameters related to tuning appear on the **Tuning** tab. Uncategorized parameters appear on the **Others** tab. Some parameters appear on multiple tabs. The tabs you see and their contents will vary, depending on the type of block you select.

You can also customize the tabs of the block configuration window. You can add or delete tabs, change the order in which they appear in the window, and add or remove any block parameters from an individual tab. For more information, refer to *Categorizing Block Parameters* section in Chapter 2, *Using the NI-FBUS Configurator*.

Note When you customize a tab of the block configuration window, that customization applies to all blocks of the same type, even in other projects.

Block Configuration Window Units

Some information in the block configuration window is presented in time, however units are not explicitly shown. The units are 1/32 of a millisecond. For example, if `Execution Time` is listed as 736, you should divide 736 by 32 to get the execution time of 23.8 milliseconds.

2

Using the NI-FBUS Configurator

Start the NI-FBUS Configurator and NIFB Process

1. Install and configure your software and any FOUNDATION Fieldbus interfaces, as instructed in your getting started manual.
2. Install device descriptions, as instructed in your getting started manual.
3. Install and wire your FOUNDATION Fieldbus device(s), as instructed in the documentation that came with your device(s).
4. Select **Start»Programs»National Instruments FBUS»NI-FBUS Configurator** to start the NI-FBUS Configurator and when prompted, click on **Yes** to start **NIFB.EXE**.

   OR

   Select **Start»Programs»National Instruments FBUS»NIFB** to start the NIFB process, then select **Start»Programs»National Instruments FBUS»NI-FBUS Configurator**. Refer to your getting started manual for more information about the NIFB process.

Create a New Project

1. When you start the NI-FBUS Configurator, the **Add Links** dialog box appears automatically. If you want to create a new project at any other time, select **File»New**.

Tip Close any open projects before opening or creating another. The multiple window structure of the NI-FBUS Configurator quickly becomes confusing with multiple open projects.

2. In the **Add Links** dialog box, each configured interface will appear. Select the link(s) (ports on your FOUNDATION Fieldbus interface) you want to include in your project in the **Added Link(s)** checklist.
3. Select the type of project to create (usually, this will be online).
4. Click on **OK**.

The NI-FBUS Configurator scans the Foundation Fieldbus network and checks addresses and object tags. If a device is missing either an address or a tag, the NI-FBUS Configurator assigns one. If the NI-FBUS Configurator detects duplicate tags, it renames the device or block, appending a numeric identifier to the original device or block tag.

After you complete the preceding steps, the configuration tree appears, displaying all the Fieldbus objects on the link or links you chose in Step 3.

During the initial scan of the bus, the following state icons sometimes overlap the configuration tree icons:

- Updating—This icon appears when the NI-FBUS Configurator is reading or writing to the object.
- Error—This icon appears when the NI-FBUS Configurator detects or encounters an error with the object.
- Invalid—This icon appears when the indicated object is not responding to the NI-FBUS Configurator scan.

Note It is normal for objects to go invalid briefly during the initial scan of the bus or when changing the device's address. This is because the device suspends communication while setting its address.

Open an Existing Project

To open an existing project, click on the **Cancel** button in the **Add Links** dialog box, then choose **File»Open**. Browse to the file you want to edit, and click on **Open**.

If you want to edit saved configuration files without connecting to the Fieldbus, click on **No** when you are prompted to reconnect to the Fieldbus link(s).

Upload Project Configuration

Using the **Upload Project** button (or selecting **Configure»Upload Configuration**), you can cause the NI-FBUS Configurator to read a configuration from a device that has already been configured and overwrite the currently open project with that information. To prevent overwriting an existing project, close all projects, then select **File»New** before uploading a configuration. The function block parameters, schedules, linkages for publisher/subscriber, alarms, and trends will be uploaded and placed in the project. Note that the LAS schedule is not uploaded. The LAS schedule will be created from the list of linkages and function block schedules by the NI-FBUS Configurator's usual scheduling algorithm. When the upload is complete, the project can be modified, saved, or downloaded just as if you had created it by hand.

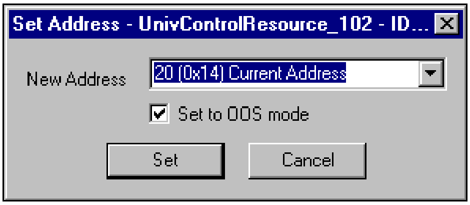
Set Device Addresses

The NI-FBUS Configurator automatically attempts to set addresses for the devices it detects. Information on manually setting device addresses is provided in case you want to change the addresses assigned by the NI-FBUS Configurator when the device is initially brought on the bus.

Note Devices may disappear off the bus while their addresses are being set.

Complete the following steps to set the device address:

1. In the configuration tree, right-click on the device whose address you want to set and select **Set Address**. The **Set Address** dialog box appears.
2. In the **New Address** field, select the desired address from the drop-down list, as shown in the following figure.



Set Address - UnivControlResource_102 - ID...

New Address: 20 (0x14) Current Address

☑ Set to OOS mode

3. Check the **Set to OOS Mode** checkbox. Doing so sets the block to `OOS` mode, which prevents the device from operating in its current control system.

Caution When you set the device address, the device loses all linkage and communication configuration information and loses control of the process. You should not change a device's address after you have configured and are running your process.

4. Click on the **Set** button.

   The NI-FBUS Configurator sets the device resource block to out of service (`OOS`) mode before setting the new address. If the device does not go to `OOS` mode, the NI-FBUS Configurator notifies you and you must determine whether to continue to set the address.

5. After you successfully set the address, you can set the resource block to the desired mode.

If you need to expand the range of available addresses shown in the **New Address** field, use the network parameters window to modify the number of polled addresses.

Function Block Instantiation and Deletion

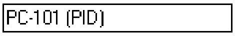
The NI-FBUS Configurator detects whether a given device supports function block instantiation (creation) and deletion. If a device supports instantiation and deletion, **Instantiate new block** appears in the device's object menu. (This menu can be accessed by right-clicking on the device, or from the menu bar when that device is selected). When you select **Instantiate new block**, a dialog box appears that shows the available block types for this device, as read from the device description. You may choose the block type and the number of blocks of that type to instantiate. To delete a block from a device which supports deletion, click on the block and press the <Delete> key.

Note Most devices do not support instantiation and deletion of function blocks. For these devices, all the available function blocks are pre-programmed and appear automatically in the configuration tree under the device.

Set Device or Block Tags

Note If you have multiple host machines, setting a device or block tag may affect how other host machines on an operating network access a device.

1. There are three ways to change a tag:
   - In the configuration tree, right-click on the device or block whose tag you want to change and select **Set Tag**.
   - Select the object's menu item from the main menu bar and select **Set Tag**.
   - Double-click on the device or block in the configuration tree to launch the block configuration window. Then, click on the Device Tag field in the block configuration window.



PC-101 (PID)

2. In the **New Tag** field, type the tag you want to assign.
3. Make sure the **Set to OOS Mode** checkbox is selected.

Caution When you set the device tag, the device loses all linkage and communication configuration information and loses control of the process. You should not change device tags after you have configured your process.

4. Click on the **Set** button. The NI-FBUS Configurator sets the block or device to `OOS` mode before setting the new tag. If the device or block does not go to `OOS` mode, the NI-FBUS Configurator notifies you and you must determine whether to continue to set the tag. You should not try to set the tag without putting the device in `OOS` mode.

Note It may take a few minutes for the tag to set.

5. After you successfully set the tag, double-click on the block icon to launch the block configuration window.
6. On the **Process** tab, change the block from `OOS` mode to a desired mode (usually `Auto`) by selecting the desired `Target` mode from the `MODE_BLK` parameter. For a description of modes, see the *Function Block Modes* section.

Function Block Modes

The most common operational modes for function blocks on devices are Automatic (`Auto`), Cascade (`Cas`), Manual (`Man`), and Out Of Service (`OOS`).

- In Automatic mode, the block is running under its normal automatic control, using a local setpoint value in the normal block algorithm to determine the output value.
- In Cascade mode, the block is receiving its setpoint value from another function block for use in the normal block algorithm to determine the output value. A linkage object makes the connection between the two blocks. The linkage object is created implicitly when you wire the output of one function block to the input of another as described in the *Create and Edit Function Block Applications* section. For example, a PID function block receives its setpoints from an upstream block.
- In Out Of Service mode, the block is not running at all. Normally, this mode is used during block configuration. Some devices require that the function block be in Out Of Service mode when changing certain parameters.
- In Manual mode, the block output is not being calculated by the normal block algorithm. The operator writes the output of the block directly.

The MODE_BLK Parameter

The MODE_BLK parameter for a function block is the parameter that contains information on the modes of the block. It has four fields: TARGET, ACTUAL, PERMITTED, and NORMAL. PERMITTED and NORMAL are defined by the device manufacturer.

- PERMITTED contains a list of all allowable modes for that block.
- NORMAL is the mode the device manufacturer expects the block to be in during normal operation.
- ACTUAL is the current operating mode of the function block on the device.
- TARGET is a field that is writable by the user. Writing this field tells the device to change to the specified mode. The device will attempt to change the mode. If it is successful, the ACTUAL mode changes to reflect the TARGET mode.

Note If the block is not currently scheduled, it will always remain in Out Of Service (OOS) mode, regardless of any writes to the TARGET field. A block is scheduled by placing it in a Function Block Application Editor window and downloading the project to the device. Refer to the *Add Blocks to the Function Block Application* and *Download a Project Configuration* sections for more information.

For more information on all the allowable modes, refer to the *MODE_BLK (Diagnostic, Process)* parameter description in Chapter 4, *Parameters*.

Configure Block Parameters

Categorizing Block Parameters

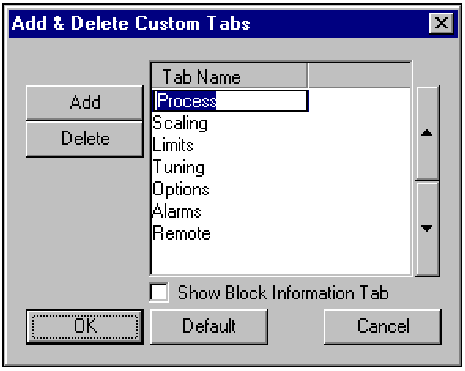
The NI-FBUS Configurator lets you customize the parameter categorization. You can add or delete custom tabs on the block configuration window.

All custom changes that you make will be saved on this computer, and associated with this block type, so all blocks of this type in all projects will use your custom setup.

If you do not want to change the default categorization, skip to the *Editing Block Parameters* section.

Adding Tabs

1. Click on the **Add or Delete Custom Tab** button in the block configuration window toolbar. The **Add & Delete Custom Tabs** dialog box appears.



2. Click on the section where you want to add the tab.
3. Click on the **Add** button. Click on the newly created blank field and type the name of the new tab.
4. The **Block Information** tab contains additional read-only information on each block, such as execution time. Checking the **Show Block Information Tab** checkbox will add this tab to your view.
5. Click on the **OK** button to generate the new tab.

   On the new tab in the block configuration window, right-click and select **Customize Parameters**. Follow the steps in *Customizing Parameters on a Tab* to configure the parameters for your new tab.

Customizing Parameters on a Tab

You can customize parameters on a tab by using the **Customize Parameters** button. Check the boxes next to the parameters you want to show on this tab, or uncheck the boxes to remove the parameters from this tab.

Removing Tabs

To remove a tab from the block configuration window, complete the following steps:

1. Click on the **Add or Delete Custom Tab** button in the block configuration window toolbar. The **Add & Delete Custom Tabs** dialog box appears.
2. In the **Add & Delete Custom Tabs** dialog box, click on the name of the tab that you want to remove.
3. Click on the **Delete** button.
4. Click on **OK**.

Modify Block Configuration Window View

You may need to resize the block configuration window to view all buttons and parameters.

To update the screen, click on the **Simple Verify** button. In the dialog box that appears, click on **Verify**, then click on **Done**.

To add space between parameter listings, click on the **Spaced View** button.

To display the **Parameter**, **Value**, **Type & Range**, and **Help** columns, click on the **Detailed View** button. To return to displaying only the **Parameter** and **Value** columns, click on the **Detailed View** button again.

Editing Block Parameters

The default tab groupings in the block configuration window are by functions. Parameters with a plus sign beside them are records or arrays (structures) that contain more than one entry. To expand the record or array, click on the plus sign, or to collapse the structure, click on the minus sign. The sub-elements of a record are called fields.

The drop-down box for editing enumerated types on blocks is sized according to the width of the **Value** column. You may need to resize the **Value** column to view the entire entry in a drop-down box.

Chapter 4, *Parameters*, includes complete descriptions of the parameters and their values.

To edit a block's parameter value, complete the following steps:

1. Double-click on a function block to open the block configuration window.
2. By default, records are expanded. If for some reason they are collapsed, click on the plus sign to see the values of the record or array parameters.
3. Use the cursor to select the parameter value you want to change. In many cases, you can select the desired setting from a drop-down menu, or type in the new value. If a parameter value is grayed out, it is read-only and you cannot change it. After you change a parameter value, an asterisk (*) appears by the parameter name and the field turns yellow.
4. To make your changes effective immediately, click on the **Write changes** button on the block configuration window toolbar. If the device accepts the new setting, the asterisk and yellow color will clear. If the device does not accept the new setting, you are probably trying to write a setting that is invalid for your device. Check the help information for that parameter or your device documentation to identify valid settings.

Tip If you decide that you do not want to write the changed value to the device, you can undo your change to the block configuration window by reading the parameter value stored in the device as described in the *Manually Update Block Parameters* section.

5. If you do not click on the **Write changes** button, you can make all your changes take effect when downloading the configuration by checking the **Write Contained Block Parameters** checkbox.

You can also click on the **Write All** button to write all parameters, including changed block tags, to the block on the device.

Note Some devices will require that you set their mode to OOS mode before writing certain parameters. A dialog box will appear to inform you of this if you attempt to write such a parameter in any other mode.

Updating Block Parameters Values

You can update block parameter values in the block configuration window manually or automatically, as described in the following sections.

Manually Update Block Parameters

To update one particular block parameter value manually, select the parameter and click on the **Read Selected** button in the block configuration window toolbar.

To update all the block parameter values, click on the **Read All** button.

Automatically Update Block Parameters

To set the NI-FBUS Configurator to update the block parameter values periodically, complete the following steps:

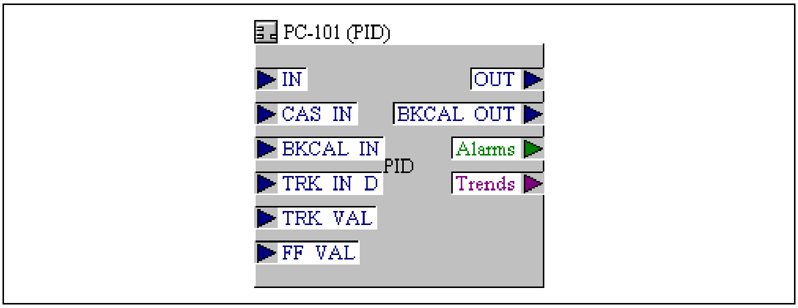
1. At the top of the block configuration window, check the **Periodic Updates** checkbox. When this box is checked, the NI-FBUS Configurator periodically checks the bus for any changes in values for the parameters and refreshes their values in the window.

   An icon at the upper-left corner of the block configuration window blinks green when you select the **Periodic Updates** checkbox.
2. Select the poll rate from the drop-down list next to the **Periodic Updates** checkbox. The NI-FBUS Configurator will update all block parameter values at that rate.
3. An icon at the upper-left corner of the block configuration window blinks green to indicate you selected **Periodic Updates**.

Note If you select a fast update rate, the reads from the bus might not be able to occur at that rate due to bus traffic, especially if you are already running a process.

Create and Edit Function Block Applications

Function blocks contain an algorithm and multiple parameters to control a process. The names of input and output parameters of a function block are displayed on the block diagram, as shown in the following figure.



The pointers on the outside of the parameters show the data flow direction of that parameter. By default, the block tag is at the top, the inputs are on the left, and the outputs are on the right.

You can resize the block and move the block's tag relative to the block.

Add Blocks to the Function Block Application

1. To open the Function Block Application Editor window, double-click on the **Function Block Application** icon in the configuration tree.
2. Drag blocks from the configuration tree to the Function Block Application Editor until all the blocks you want to use are in the Function Block Application Editor.

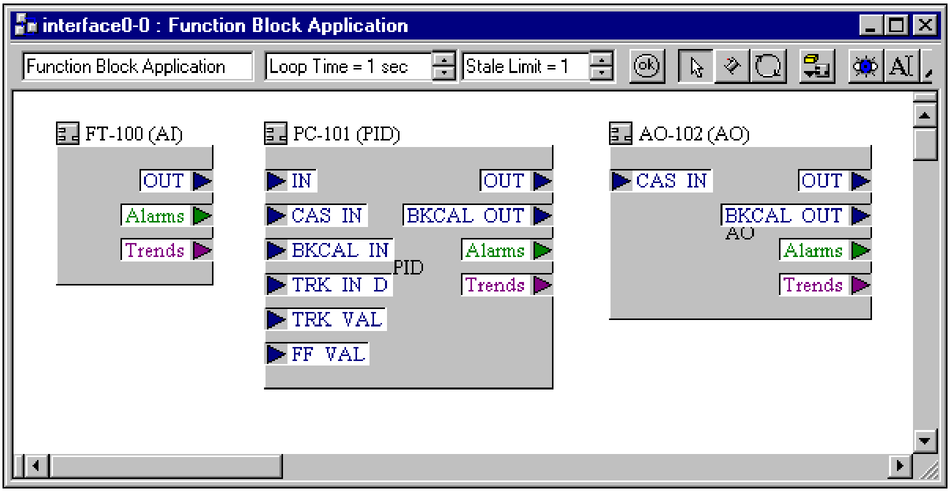
Tip You can drag host devices from the configuration tree into the Function Block Application Editor. Dragging host devices allows you to configure alarms and trends, as described in the *Configuring Alarms* and *Configuring Trends* sections.

You can also add blocks as follows:

1. Right-click on the background of the Function Block Application Editor or click on the **Block** button in the toolbar of the Function Block Application Editor.
2. Select the device or interface from the menu that appears, and then select the block you want to place on the screen from the menu of function blocks.

Note You can use a function block in a Function Block Application Editor window only once. After the block has been dragged over, it will be grayed out on the menu and in the configuration tree. To use one block's output as inputs to multiple downstream blocks, wire the output to both locations. To use one block's output in multiple loops, refer to the *Defining Loops* section.

The following figure shows how the Function Block Application Editor looks with function blocks dragged onto it.



Connecting Blocks

To send data from one block to another, you must connect the outputs and inputs. You can wire blocks manually or use a template. The help window on the right side of the screen describes the input and output parameters and how they can be connected.

All the necessary arrangements for broadcasting data over the bus are automatically generated when you wire the function blocks together. When you later download your project to the bus, this scheduling and linkage information will be downloaded to all the link masters on the link.

Wiring Blocks Manually

To wire blocks manually, complete the following steps:

1. Click on the **Wiring** tool on the Function Block Application Editor toolbar.
2. Click on one terminal you want to connect.

Tip You do not have to connect points in any particular order. You can start at the input and wire to the output, or vice-versa.

3. To tack down the wire at different points, click on the background of the Function Block Application Editor window.

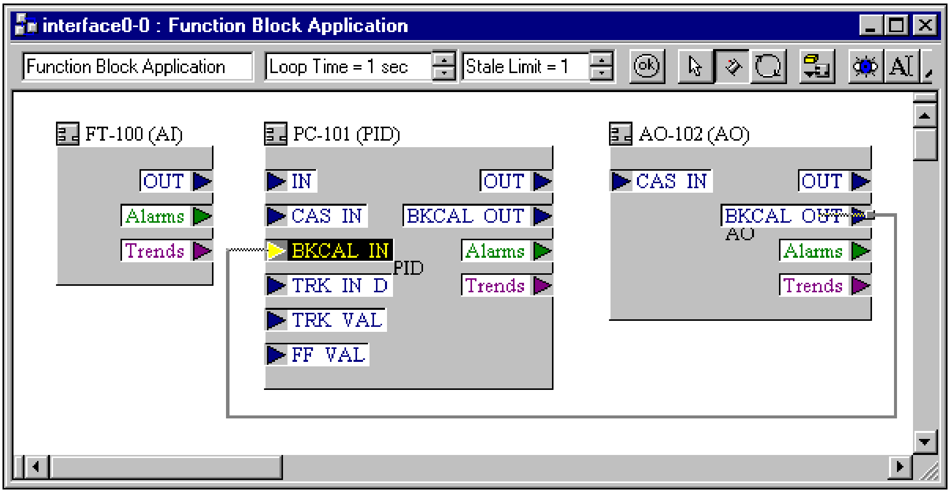
Tip To toggle the direction of the connection you are dragging, press the spacebar while wiring.

Tip To zoom in and out, click on the magnifying glass icon. To zoom in, drag a rectangle around the area you want to enlarge. To zoom out, right-click on the background of the Function Block Application Editor window. You can also select different zoom settings from the **Zoom** menu.

4. Click on the terminal to which you want to wire.

When you click on the Wiring tool, the NI-FBUS Configurator performs type checking and highlights valid inputs and outputs when you move the cursor over them. For instance, it will not allow you to wire an analog output to a discrete input.

The following figure shows the `BKCAL OUT` output of the AO block being wired to the `BKCAL IN` input of the PID block.



interface0-0 : Function Block Application

Function Block Application

Loop Time = 1 sec

Stale Limit = 1

Checking for Function Block Application Errors

Click on the **Check Project for Errors** button in the Function Block Application Editor window to perform error checking, regenerate the schedule of the application, and remove bad wires.

Customizing Your Function Block Application

Click on the **Text** button to type additional information about the application and change text in fields.

Click on the **Fill** button to fill objects on the screen with a color.

Click on the **Fill Color** button to select a color to be used with the **Fill** tool.

Using Templates

Templates are pre-defined, pre-wired control loops that you can drop into the Function Block Application Editor to use as a model for your function block application. Templates can help you by providing the wiring for common control strategies, such as PID loops. You can also create your own templates to reduce time required for configuring multiple "copies" of your own strategies.

Tip Templates store parameter values as well as wiring connections, saving you the time-consuming operation of configuring function block parameters when you have multiple copies of the same type of control strategy.

The NI-FBUS Configurator includes the following templates:

- PID Feedback Control
- Manual Control with PV Indication
- PD Feedback Control
- Track Feedforward Control
- Cascade Control
- Ratio Control
- Override Control

National Instruments-defined templates are provided for wiring use only; the values in those templates will be overridden by your function block settings. The values stored in user-defined templates will be applied or not applied, at your discretion, when you use the template. It is *not* recommended that you accept the template values for templates you did not create; the parameter settings may not be appropriate for your device(s).

Inserting Templates

To access templates, do one of the following:

- Right-click on the background of the Function Block Application Editor and select the desired template from **FBAP Templates**.
- Select **File»Open Template**.

A dialog box appears asking how to apply the template's parameter values to the function block that will eventually replace the template. You can choose one of four different options:

- Click on the **No** button to keep the function block's current values. This will ignore any values stored in the template. If you did not create this template with parameter setting appropriate for the device(s) you plan to use in it, this is the appropriate option.
- Uncheck the **Use *Template* CHANNEL values?** checkbox and click on the **Yes**, **use values** button to change the parameter values of the function blocks used to fill the template to match the template's parameters (except the `CHANNEL` parameter). This is useful if you have set up a PID loop for one set of input and output devices and would like to use the same loop with the same settings for other sets of identical input and output devices. In this case, you would save your initial PID loop as a template. Then, you would insert the template, uncheck the **Use *Template* CHANNEL values?** checkbox, and choose **Yes, use values**.

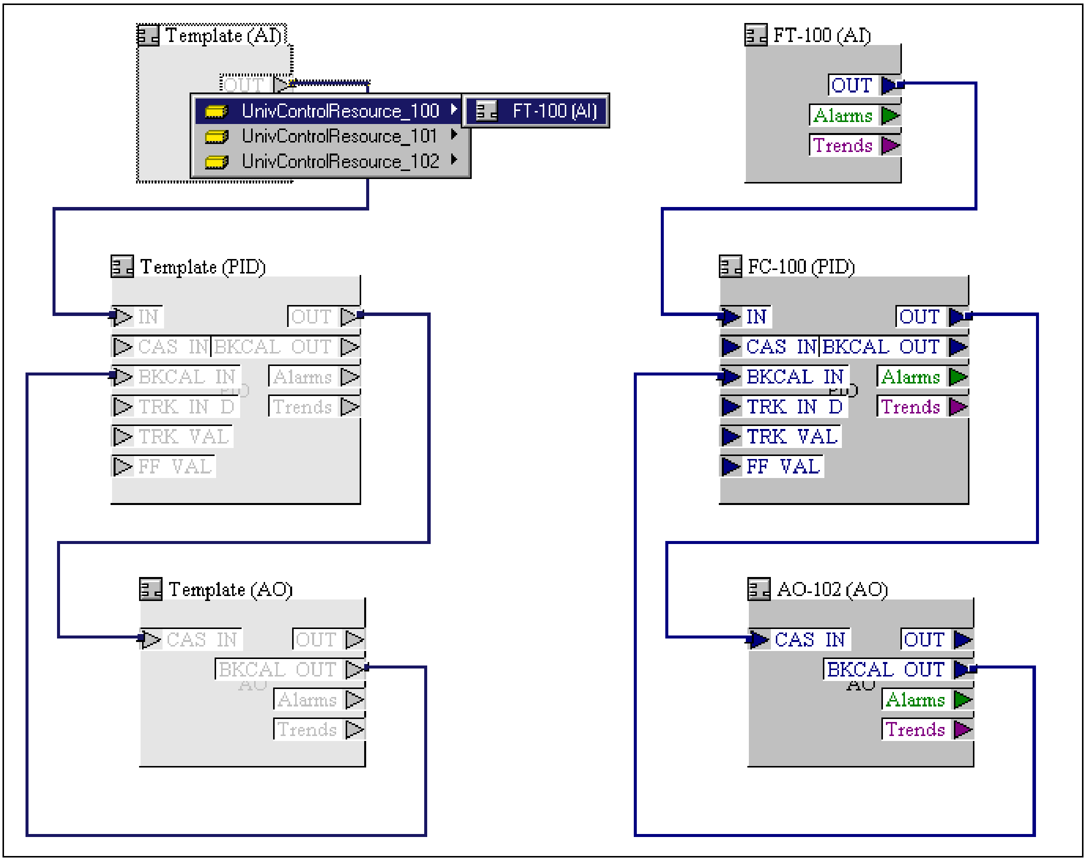
  In this case, any function block used to fill the template will take on the saved parameters, except it will keep its own `CHANNEL` parameters (that is, device input or output channel that is set to control). See the *Creating Templates* section for more information on creating templates.
- Check the **Use *Template* CHANNEL values?** checkbox and choose **Yes, use values** to change the function block's parameter values to match all the template's parameters, including the `CHANNEL` parameters.
- Click on the **Decide Later** button to do nothing now, but ask the same question for each individual function block when you actually fill the template.

Tip If you select the **Decide Later** option, you can later browse the template's parameter values by right-clicking on the template's block outline and choosing **Open**, or by double-clicking on the block.

While the template is highlighted in the Function Block Application Editor, you can drag it to the desired location.

Tip Moving the mouse just outside of the Function Block Application Editor window while dragging blocks will cause it to scroll.

The following figure shows an example of a template, both when initially placed and when filled. The template on the left shows how a template appears when initially placed, and the template on the right shows how it looks after it has been filled in.



When templates are initially placed, the template blocks are grayed out. This is because you have not assigned a function block to the template block. To assign a function block to the template block, complete the following steps:

1. Double-click on the template block to view all the blocks that match this block type in your project.
2. Select the desired block from the list that appears.

You must replace all the template blocks with function blocks from your project. The **Errors** tab lists errors warning you of unassigned templates.

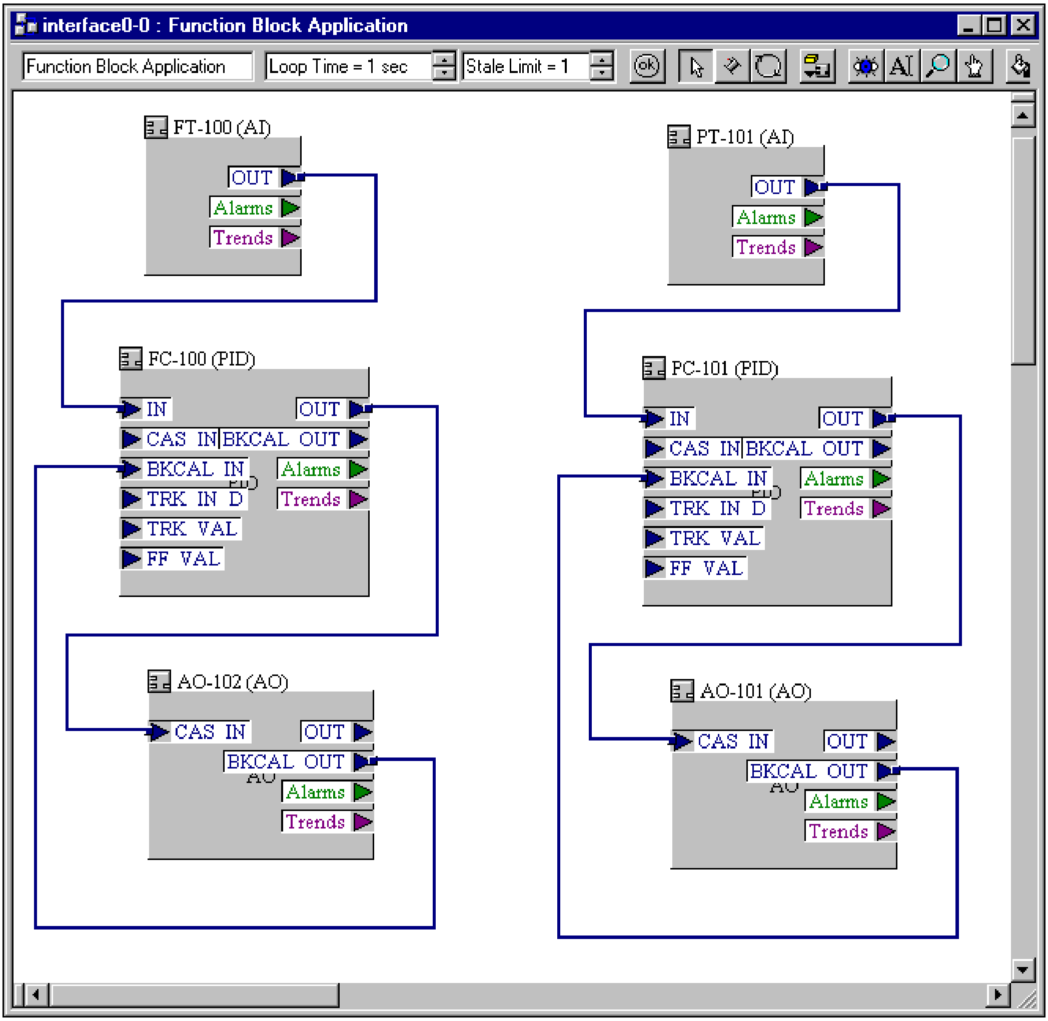
Creating Templates

After you have set up the function block parameters and connections for the first unit, you can save that setup as a template for later use. If you want to save only part of your function block application as a template, select that part (drag the pointer around the desired blocks and wires OR use the Control key while clicking on the desired blocks and wires). With the Function Block Application Editor window open, select **File»Save Template**. You can choose to save only selected items or the entire Function Block Application. By default, the NI-FBUS Configurator saves all templates in the `\fcs\templates` directory under your installation of NI-FBUS. If a template is in that directory, it automatically appears as an available template in the template menu.

Defining Loops

A *loop* is a group of connected function blocks that execute at a specified rate. You can define multiple loops in one Function Block Application.

The following figure shows an example of two separate loops running at the same rate. Loops running at the same rate can be placed in a Function Block Application without using a loop structure.

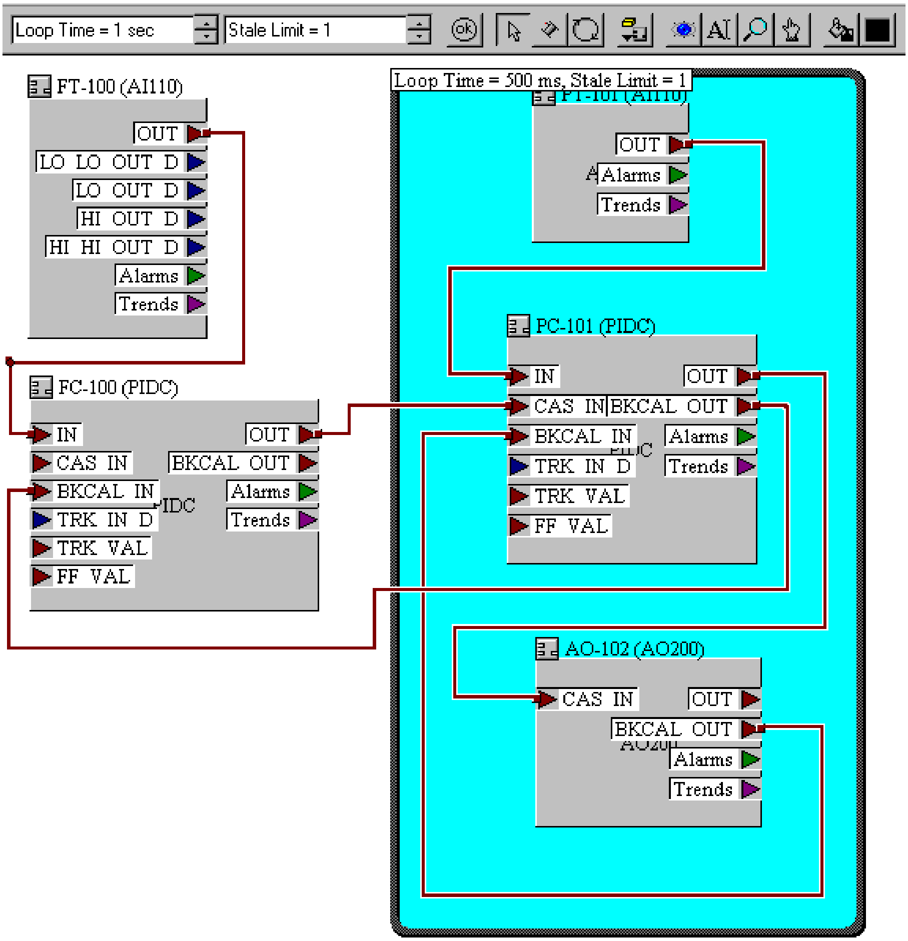


Defining Multiple Loops Running at Different Rates

The loop execution time shown on the Function Block Application Editor toolbar is the default execution time for all function blocks on the window *not* isolated in a loop structure. You can use the loop structure to isolate a group of function blocks to run at a different execution rate than the rate on the Function Block Application Editor toolbar.

For example, in the following figure, the AI and PID blocks to the left are one loop and execute at a rate of once every second, as shown in the toolbar.

The AI-PID-AO loop to the right executes once every 500 ms, as shown at the top of the loop structure.



Loop Time = 1 sec

Stale Limit = 1

Loop Time = 500 ms, Stale Limit = 1

To create a loop structure, complete the following steps:

1. Click on the **Loop** tool on the Function Block Application Editor toolbar.
2. Click at the desired location for one corner of the loop and drag the loop around the blocks you want to include in the loop.
3. To change the loop execution time, double-click on the box at the upper left of the loop, and type the new time.

Determining Function Block Execution Time

Because function blocks must execute serially, there is a trade-off between how many PID loops you can run and how fast you can run them. The maximum execution time for each block (that is, the worst-case execution time) is specified by the device manufacturer. To view the maximum execution time for a block, do the following.

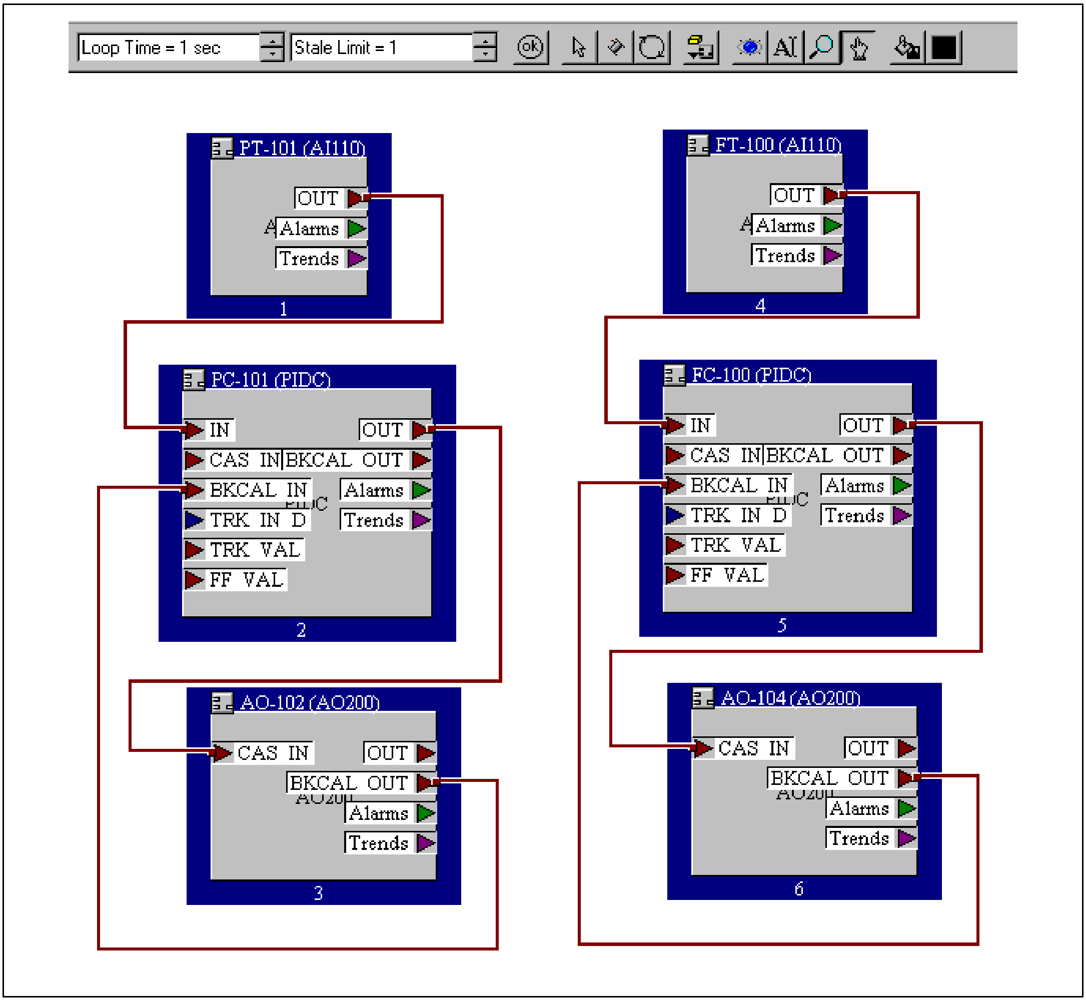
1. If you have not already done so, select **View»Preferences** and click on the **Block View** tab.
2. Select the **Show Block Information** checkbox, and click on **OK**. Now when you double-click on a function block, a new tab called **Block Information** will appear.

Look at the parameter `EXECUTION_TIME` on the **Block Information** tab for the function block. If the number is a hexadecimal number, you can change it to decimal by right-clicking and unchecking Hexadecimal Data. The decimal value of this parameter is the execution time in 1/32 ms, as required by the FOUNDATION Fieldbus specification (that is, dividing this number by 32 will give the maximum execution time, in milliseconds). For example, the maximum execution time for a National Instruments FP-3000 AI-110 block (with firmware revision 2) is listed as 192. Dividing this by 32 gives 6 ms maximum execution time.

Changing the Block Execution Order

To change the execution order of the blocks, complete the following steps:

1. In the Function Block Application Editor, click on the **Execution Order** button. Numbers appear on the blocks, indicating their current order, as shown in the following figure.



2. Click on the blocks in the order that you want them to execute.

3. When you are done, click on the background of the Function Block Application Editor or select another tool. The change will be reflected in the schedule. For more information, refer to the *View and Edit a Schedule* section.

Setting the Stale Limit

Stale Limit = 1

Click on the **Stale Limit** field to adjust the number of consecutive duplicate values that a function block accepts before it labels the input data as stale (old data).

Using Menus and Methods

The NI-FBUS Configurator supports user methods, which are a sub-set of the FOUNDATION Fieldbus Menus and Methods specification.

User methods are the methods you explicitly invoke from the NI-FBUS Configurator. Typically this would be a method that is called "Set up device" or "Calibrate valve"—something that represents a specific task you want to perform.

If you double-click on a function block in the configuration tree and the block has menus and methods associated with it, a menu bar appears above the block configuration window toolbar and the **Methods** tab appears in the block configuration window.

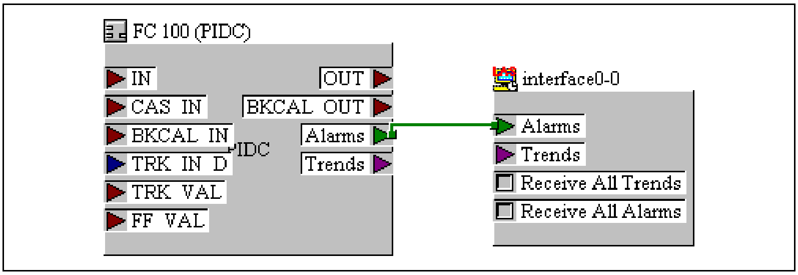
The first menu item in the block configuration window is normally **Apply values**, which shows up for all blocks. Any additional menus defined by the device description will appear to the right of this menu. These menus can contain parameter names or the names of methods, which can be invoked by selecting them. If there are any methods detected in the device description that are not already on a menu, a menu titled `Methods` will be shown, with an entry for each method.

If you select a parameter name from the menu, an edit window for that parameter will appear. To invoke a method, select it from the menu bar. The block configuration window will switch to the **Methods** tab, which displays all output from the method. In addition, when the method requires you to enter input, you will be prompted to type it in a box in the **Methods** tab. Methods will run until they encounter an error or complete. Error information is also displayed in the **Methods** tab. You cannot close the block configuration window while a method is still running.

Configuring Alarms

You can configure a device, such as an interface in a PC, to receive alarms generated by function blocks.

1. Drag the configuration tree icon for the device or interface to which you want to send the alarm to the Function Block Application Editor, if you have not done so already.
2. Either wire the **Alarms** output of the block that you want to generate an alarm to the device you want to receive the alarm, or check the **Receive All Alarms** box, if the device has one. The following figure shows an example of a configured alarm.

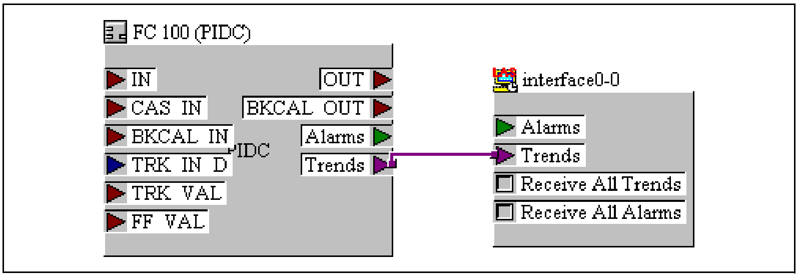


3. Double-click on the function block that you want to generate the alarm. The block configuration window appears.
4. In the block configuration window, click on the **Alarms** tab to view or change alarm information, such as the high and low limits of an alarm condition, alarm priorities, and so on.

Configuring Trends

You can configure a device, such as an interface in a PC, to receive trends from function blocks.

1. Drag the configuration tree icon for the device or interface to which you want to send the trend to the Function Block Application Editor, if you have not done so already.
2. Either wire the **Trends** output of the block to the device you want to receive the trend, or check the **Receive All Trends** box, if the device has one. The following figure shows an example of a configured trend.



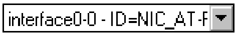
3. Double-click on the function block from which you want to send a trend. The block configuration window appears.
4. In the block configuration window, click on the **Trends** tab.
5. Check the **Trend** column checkbox next to the parameter you want to trend.

6. Set the sample **Interval** of the trend in loops/sample. The interval should be a multiple of any loop macrocycle time in which this block is involved. For example, if the interval is 2 loops/sample, and the loop macrocycle time is 1 sec, the trend will acquire a value every 2 sec.
7. Set the **Sample Type**. Select **Instant** or **Average** from the **Sample Type** drop-down list. **Instant** trends the actual value. **Average** trends all samples.

Note If you want to view trends on a host computer, that computer must have software. The NI-FBUS Configurator does not have trend-viewing functionality.

View and Edit a Schedule

The NI-FBUS Configurator automatically determines the execution order and timing of the schedule based on your control strategy, network and device factors, and the loop execution rate you choose. The NI-FBUS Configurator also automatically optimizes the schedule.


interface0-0 - ID=NIC_AT-F

The Link Active Scheduler (LAS) is a device that dictates both scheduled and unscheduled communication according to the schedule downloaded to it. The **Primary LAS** field on the schedule window displays the name of the device that is the primary LAS for the system.

Change the Link Active Schedule

You can change the schedule by completing the following steps:

1. Double-click on the **Schedule** icon in the configuration tree. The schedule window appears.

100ms/div

2. Click on the **Scale** field to display or change the time and divisions on the timing diagram.
3. Use the cursor to drag the function blocks in the schedule as desired. Dragging bars in the schedule shuts off the automatic optimization.

   If you want to reoptimize, click on the **Optimize Schedule** button on the schedule window toolbar.
4. During the time between scheduled and unscheduled communications, the LAS allows unscheduled communications to occur. Unscheduled communications are always secondary to scheduled communications. The ratio of scheduled to unscheduled time can have an affect on data throughput rates.

5. To verify the schedule, click on the **Errors** button in the schedule window.

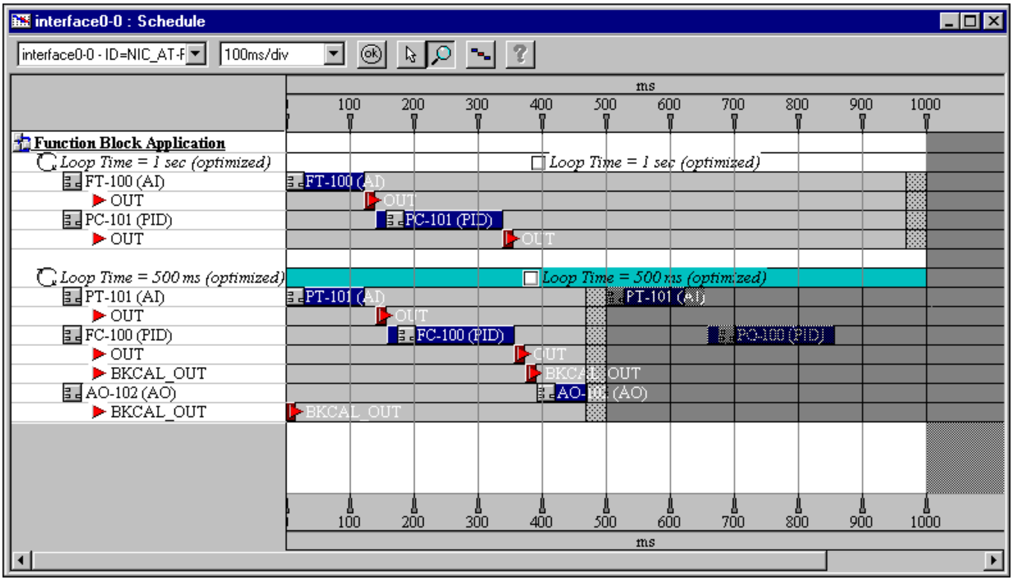
Tip To zoom in and out, click on the magnifying glass icon in the schedule window.

Tip With only one device, you will not see data transmission since there are no parameters that need to be transmitted to other devices. Also, for some fast devices (such as the FP-3000), the blue blocks representing execution times are so short that they are entirely hidden behind the block icons.

Note An optimized schedule can show that the transmission time overlaps the end time for the function block because the transmission time includes the time that the LAS uses to tell the device to send the data.

Multiple Loop Representation

The following figure shows how the NI-FBUS Configurator displays multiple loops in the schedule window.



The loops are separated by white dividers. If you would like to lock the current schedule for a given loop, click on the **Loop Time** checkbox shown above the colored bars. This will prevent you and the NI-FBUS Configurator from changing the schedule until you uncheck the box.

Set Network Parameters

To change network parameters such as the Link Active Scheduler (LAS) and primary time master, double-click on the **Network Parameters** icon in the configuration tree.

The **Network Parameters** dialog box appears. This dialog box lists all devices capable of being link master for the link. Every device you select in this dialog box receives the Link Active Schedule when you download your configuration. Thus, if the primary LAS is unavailable, one of these devices will step in and continue executing the Link Active Schedule. This is how the Foundation Fieldbus implements redundancy.

Advanced link settings control the interaction of devices over the network. They are set on a port-by-port basis.

Caution Do *not* modify advanced parameters without good reason. If you must modify parameters for certain devices, the device manufacturer will recommend settings. Modifying these parameters can have an adverse affect on data throughput rates. If settings are incorrectly modified, some devices may disappear off the bus.

To revert to the default network parameters, double-click on Network Parameters. Click on the Link Settings tab. Press the **Press to See Suggested Values** pushbutton. The next window shows the original settings.

Change Language

At the device manufacturer’s discretion, names and help strings can be provided in multiple languages.

1. Select **View»Preferences»General tab**.
2. In the **DD Language Codes** field, enter the ISO standard two-letter language abbreviation.

Save a Project Configuration

To save a configuration, select **File»Save**. This saves the entire project, including the configuration tree, Function Block Application, schedule, and notes/log.

Download a Project Configuration

Up to this point, the only information sent to the devices has been the parameters written from the block information window. Before function blocks will execute on the device, the rest of the configuration must be downloaded to the devices. Function block linkages must be downloaded to allow dataflow from one block to another. Also, the schedules for function block execution and communication have not yet been provided to the devices. When you download a configuration, the NI-FBUS Configurator sends the following information over the bus:

- Communication schedule for the LAS
- Function block execution schedule for devices
- Function block linkages
- Alarm configuration
- Trend configuration
- Static parameters

To download your project configuration, complete the following steps:

1. Click on the **Download Project** button in the configuration tree toolbar or select **Configure»Download Configuration**. The **Download Configuration** dialog box appears.
2. Select the configuration object to download.
3. If you have not done so already, click on the **Link Masters** button to set link masters to receive the primary Link Active Scheduler, the primary time master of the link, and the Link Active Schedule. The **Network Parameters** dialog box appears.
4. (Optional) Check the appropriate checkboxes for the download options. Refer to the *Download Options* section for more information. In general, you should use **Clear Devices** and **Automatic Mode Handling** on all downloads.
5. To download your configuration changes to the network, click on the **Download** button. The **Download Status** progress bar shows what percentage of the configuration is complete.

   While downloading, the NI-FBUS Configurator displays and updates the **Download** tab in the status window. It shows what device is currently being configured, its settings, and whether it was successfully configured.

6. If you did not check the **Automatic Mode Handling** checkbox, click on the **Set All Targets to OOS** button to switch the function blocks to `OOS` mode. After the configuration downloads, you must set the blocks back to their normal mode of operation to begin operating.

Download Options

There are several options in the download configuration dialog box for downloading a configuration.

Write Contained Block Parameters

Write Contained Block Parameters writes all contained parameters (parameters that are internal to the function block, such as the scaling and alarm limit settings) from the block configuration windows. This is especially useful if you are changing parameters in multiple function blocks and do not want to take the time to write them block by block.

Note Per the FOUNDATION Fieldbus specification, some contained parameters have default values that are not legal values. This is done to encourage users to change these parameters to values appropriate for their process. Often, device manufacturers will allow their devices to operate with uninitialized parameter values, but any attempts to rewrite the uninitialized value to a parameter will result in an error. As a result, using **Write Contained Block Parameters** generates errors if any parameters are left uninitialized. Such errors can be cleared by writing a legal value to the parameter or by reading into the NI-FBUS Configurator the value from the device. Generally, such errors are non-fatal and the download can continue successfully.

Clear Devices

Clear Devices clears the existing configuration in the devices before downloading. It is good practice to use this option. Using this feature ensures that the linkages you see in the Function Block Application Editor window are the only linkages in the device. Otherwise, linkages from a previous download may persist.

Automatic Mode Handling

Automatic Mode Handling automatically takes the blocks Out Of Service (`OOS`) as needed during the download process and returns them to their normal modes of operations (as read from `MODE_BLK.NORMAL` for each block) after the configuration is downloaded. This is a time-saving and convenience feature. If this option is not checked, you will be prompted to take the blocks to `OOS` as needed and will have to set them back to `NORMAL` by hand after the download completes.

If you do not select **Automatic Mode Handling**, the NI-FBUS Configurator shows a table of all the block modes at the start of configuration. You have the ability to change any of the modes from this table, or to set them all to Out of Service using the **Set All Targets to OOS** button. The table appears again at the end of configuration, and allows you to set the modes by hand, or click the **Set All to Normal** button to put all the blocks into their normal modes. Note that the modes you have selected will not be written to the bus until you click the **Continue** button.

Verify & Diff

This option causes the NI-FBUS Configurator to upload the actual configuration from the devices after a download. It will then compare the uploaded information to the configuration file in the NI-FBUS Configurator (refer to *Compare Two Configurations* for instructions). This will report any differences between the configuration file and the actual configuration of the devices. It is useful to compare a configuration file to the bus in cases that the exact configuration on the bus is unknown or to verify that the devices accepted all assignments.

Bringing a Block Online

If you did not select **Automatic Mode Handling** during the download process, you will have to manually change the function blocks back to their normal mode of operation.

1. Go to the MODE_BLK parameter of the resource block and set the TARGET to Auto. Wait a few seconds and re-read the MODE_BLK parameter. The ACTUAL field should go to Auto. If it does not, examine the parameters on the **Diagnostics** tab to identify the problem.
2. Go to the MODE_BLK parameter of the function block you are interested in and set the TARGET to Auto.
3. Re-read the MODE_BLK parameter. The ACTUAL field should go to Auto. If it does not, examine the parameters on the **Diagnostics** tab to identify the problem, or refer to your device documentation.

Once the block goes to its NORMAL mode, it is executing on the device.

Monitoring Parameters

After downloading your configuration, you can display a graphical representation of the updates to input, parameters, and mode parameters of function blocks.

1. On the Function Block Application Editor toolbar, click on the **Monitor** button, or select **Configure»Monitor**. This option is available only when the Function Block Application Editor is active.
2. In the monitor window, select the types of items you want the NI-FBUS Configurator to monitor and display.
3. Modify the update rate as desired.
4. Click on the **Start Monitoring** button to monitor the application.

The mode box for a block will be outlined in red if the actual mode for the block does not match the target mode, which assists in quickly diagnosing system problems.

Verify a Configuration

Quick Verification

You can perform a quick check to verify that the parameters on the bus match those in the project for a particular link, device or function block.

1. Right-click on the desired object in the configuration tree and select **Simple Verify Configuration**.
2. Click on the **Verify** button. The NI-FBUS Configurator compares the project settings to the actual settings for the selected object and displays the result in the **Result** field.
3. If the object settings in the device do not match the project settings, you can click on the **Upload Changes** button to upload the actual device settings to the project.

⚠ **Caution** Selecting **Upload Changes** modifies the currently open project to match the devices. It is not possible to undo this modification.

4. Click on the **Done** button.

Compare Two Configurations

You can upload your current bus configuration and compare it against a currently open, saved project. To do so, click on the **Verify and Diff** button in the configuration tree toolbar. You can also select **Configure»Verify and Diff Configuration**.

To filter the resulting report, select the desired boxes in the **Filter Options** menu.

With Verify and Diff, the NI-FBUS Configurator creates a difference report between the contents of your currently loaded project and the bus. The difference report includes device parameters, function block parameters, function block schedules, function block application linkages, and alarm and trend connections. When the difference report is completed, any errors that occurred will first be reported. If any errors are reported, the related portions of the difference report may be incomplete due to the errors. The report will be displayed in its own window as a tree with branches that can be expanded by clicking the "+". The report is color coded on each line, with yellow representing items that are "new" (items that appear on the bus, but not in the current project). Blue represents items that are "missing" (items that appear in the project, but not on the bus). Orange represents values that differ between the project and the bus.

Check for Errors

Click on the **Check Project for Errors** button in the configuration tree to check for errors in the project.

View and Edit a Log

The log/notes window in the project keeps track of the last downloaded configuration status output. The log/notes window shows the same output as the **Download** tab of the status window plus the time and date of the last change and previous downloads of the current project. To open the log/notes window, double-click on the **Log/Notes** icon in the configuration tree.

Double-click on a log entry to see the download log for that session. You can also add notes at the bottom of the download log for each session, and the NI-FBUS Configurator automatically saves them with the project for future reference.

Add Devices

Click on the **Add New Device** button to add another device to the project.

Add An Additional Function Block Application

Click on the **Add New FBAP** button to add another Function Block Application Editor window to the project. To change the name of the second Function Block Application Editor window, right-click on it in the configuration tree and select **Set Tag**.

Replace Devices

A device's settings (parameters, tags, and linkages) can be transferred to another device, whether the original device is functioning or not. Follow these steps to transfer a device's settings to a replacement field device.

1. Connect the new device to bus.
2. From the **Configure** menu, select **Replace Devices** or right-click on the device to be replaced and select **Replace Devices**. The original device you selected will appear in the **Missing Device** field.
3. In the **Replacement Device** field, select the new device from the drop-down list to replace the **Missing Device**.
4. After you select the desired devices, click on the **OK** button.
5. Download the configuration with the **Clear Devices** checkbox checked.

The NI-FBUS Configurator replaces the Missing Device with the Replacement Device. When you download your changes, the original device's configuration, including linkages, tags, and parameter values, is written to the Replacement Device.

Reset a Device to Factory Defaults

You can clear a field device so that it is reset to a state appropriate for shipping. To do so, right-click on the device name in the configuration tree and select **Set to Factory Defaults** from the menu that appears.

The NI-FBUS Configurator clears the device and function block tags, along with the device address. In addition, it clears all communication information in the device.

Set Preferences

To set your preferences in the NI-FBUS Configurator, select **View»Preferences**. The **Preferences** dialog box appears, containing the following tabs: **Fonts**, **Block View**, **General**, and **Links**.

Descriptions of these tabs follow:

- **Fonts**—Use this tab to change the font types and sizes on the display.
- **Block View**—Use this tab to configure the default view for parameters:
  - **Show Spaced View**—Click on this checkbox to add space between parameters.
  - **Show Detailed View**—Click on this checkbox to show additional type and help strings for the view.
  - **Show Block Information**—Click on this checkbox to show the **Block Information** tab, which contains additional read-only information on each block, such as execution time.
  - **Use Descriptive Labels on Parameters**—Click on this checkbox to show descriptive information about parameters in the block configuration window.
- **General**—Use this tab to select which windows appear when you start the NI-FBUS Configurator.
- **Links**—Use this tab to change the software key, as described in the following section, *Changing the Software Key*.

Changing the Software Key

The NI-FBUS Configurator, by default, is licensed for use on one machine with up to four Fieldbus links. If you would like to use more than four links (ports on your boards), you need a link upgrade for each link in excess of four. To purchase additional licenses, contact National Instruments at `ni.com/fieldbus` to obtain an order form for link upgrade(s).

850283-0	Part number for a one-link upgrade of the NI-FBUS Configurator.

The current software key can be found under **Help»About FCS**. You will need to provide the software serial number and the current software key when ordering. If you have not previously purchased any link upgrades, your key will be blank.

After you obtain a new key, complete the following steps to change the key:

1. Select **View»Preferences**. The **Preferences** dialog box appears.
2. Click on the **Links** tab.
3. On the **Links** tab, click on the **Show About Dialog** button. The **About** dialog box appears.
4. Click on the **Set SW Key** button, enter the new key, and click on the **OK** button. Click on the **OK** button to close the **About** dialog box.

   The number of **Links Allowed** on the **Links** tab changes based on the new software key.
5. Click on the **OK** button to close the **Preferences** dialog box.

Print

You can print your configured project in the form of text, graphical diagrams, and text explanations of the diagrams.

To print your project, select **File»Print**. From the **Print** dialog box, you can select what you want to print.

- **Entire Configuration**—Prints the entire configuration
- **Specific Configuration Items**—Lets you select specific items from the configuration tree to print.
- **Print Minimum Amount of Configuration Information**—Prints only the minimum amount of information necessary to describe the printed items. If you check this item, the NI-FBUS Configurator prints only the function blocks that are used in a function block application of the project and does not print unused blocks.
- **Print Graphical Diagrams**—Prints the function block application and graphical representation of the schedule.
- **Print Text Interpretations of Graphical Diagrams**—Prints a text version of the function block application and schedule.
- **Page Format**—Lets you select printing styles for the graphical schematics, page titles, and numbering.

3

NI-FBUS Configurator Tutorials

These tutorials will give you first-hand experience with the NI-FBUS Configurator.

Before you begin these tutorials, make sure you have successfully installed all Fieldbus hardware and software and have properly imported the device descriptions for the devices you want to use.

In these tutorials, you will learn about the following types of control applications:

- Basic connections to AI and AO signals.
- PID control.

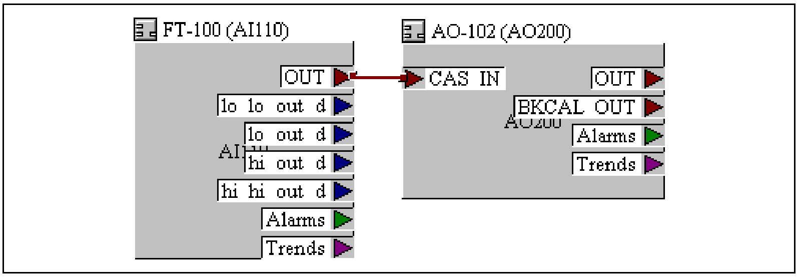
Tutorial 1: Connect to Network and Hardware

1. Launch the NI-FBUS Configurator by selecting **Start»Programs» National Instruments FBUS»NI-FBUS Configurator**.
2. A dialog box appears informing you that NIFB.EXE has not been launched and asking if you want to launch it now. Select **YES**. It takes a few moments for NIFB.EXE to launch and for the NI-FBUS Configurator to load.
3. In the **Add Links** dialog box that appears, select **Interface0-0**, and click on **OK**.

The NI-FBUS Configurator automatically detects all valid devices connected to the network. If the connected devices are registered by the Fieldbus Foundation as compliant devices and the device description files have been imported, the NI-FBUS Configurator detects both the device and the function blocks contained within the device. This process can take a few minutes. When complete, the configuration tree will display all devices and function blocks connected to your link.

Tutorial 2: Create Basic Connections to AI and AO Blocks

1. Double-click on **Function Block Application** in the configuration tree. This opens a blank window to the right of the configuration tree, referred to as the Function Block Application Editor.
2. Drag one of each of the following types of function blocks from the configuration tree to the Function Block Application Editor window:
   - AI block
   - AO block
3. Use the wiring tool to connect data together to pass the data value from one function block to the next. Connect the AI and AO modules together, as shown in the following figure. Use the pointer tool to arrange the blocks or nudge wires to make the Function Block Application Editor diagram look more organized. The Zoom menu has several options to make the code within the Function Block Application Editor more manageable.



4. At this point, you could download the configuration so that the function blocks can begin executing. However, if the device is controlling a real-world signal, it is generally better to set the parameters appropriately before downloading to avoid unexpected output values resulting from unconfigured parameters. The function blocks will not go into `Auto` mode until after they have been scheduled to execute by placing them in the Function Block Application Editor window and downloading. If you would like to download the configuration now, proceed to Step 8.
5. Fine-tune your application by adjusting the scaling. For the AI block, you will set as many as three types of scaling—`XD_SCALE`, `OUT_SCALE`, and `L_TYPE`. For more information about scaling, refer to *Scaling Parameters* section in Chapter 4, *Parameters*.

6. To change the scaling parameters, double-click on the AI function block.
7. In the block configuration window, select the **Scaling** tab.
8. Confirm the scaling parameters in particular and check that the other parameters are reasonable.
9. To change a parameter, place your mouse over the existing value and click. Then, enter values appropriate for your device. An example is shown below:

```
XD_SCALE
    EU_100       50
    EU_0         -50
    Units_Index  mV
    Decimal      3

OUT_SCALE
    EU_100       400
    EU_0         -100
    Units_Index  °C
    Decimal      3

L_TYPE           indirect
```

Note Selecting an L_TYPE of direct results in no further scaling of the transducer value (that is, OUT_SCALE is ignored). Normally, some type of output scaling is desirable, in which case one of the indirect L_TYPES should be chosen. Refer to the *L_TYPE (Scaling)* parameter description in Chapter 4, *Parameters*, for more information.

10. When you are ready to write all the new values, click on the **Write Changes** button at the bottom of the window.

    If you have already downloaded the configuration, you might get a warning telling you that the block is in the wrong mode for the request. This is because some devices allow certain parameters to be written only in OOS mode. Acknowledge the warning and click the **OOS** button. Allow a few seconds for the device to get into OOS mode before clicking **Write Changes** again. After the changes have been written, change the block back to its original mode. Another common cause for warnings when writing changes is trying to write an invalid value to a parameter. Devices will refuse values that are out of range. To correct this problem, note the parameter that caused the warning and change it to an appropriate value.
11. Modify the scaling of the AO module. Double-click on the AO function block. Scaling for the AO block is similar to that of the AI block, only in the reverse direction. PV_SCALE and PV for an AO block

correspond to the OUT_SCALE and OUT of the AI block—they are the values that are seen on the bus.

```
PV_SCALE
    EU_100      400
    EU_0        -100
    Units_Index °C
    Decimal     3
XD_SCALE
    EU_100      50
    EU_0        -50
    Units_Index mV
    Decimal     3
L_TYPE          indirect
```

12. Adjust the macrocycle of your configuration. The macrocycle is the least common multiple of all the loop times on a given link. To edit the schedule, double-click on the **Schedule** icon from the configuration tree. If you are using a fast device, you might be able to decrease the Loop Time to .1 second for this configuration. If you decrease the macrocycle too much, a warning will appear on the **Errors** tab.

13. Download your configuration by clicking on the **Download Project** button.

    In the download configuration window that appears, select **Clear Devices** and **Automatic Mode Handling**.

    Click on the **Download** button when you are ready to download your configuration. Downloading can take several minutes to complete. When finished, a dialog box appears informing you that the download is complete.

14. Test your configuration.

    a. Clicking on the **Monitoring Mode** button in the Function Block Application Editor window.

    b. In the dialog box that appears, select **Monitor Block Outputs**, **Monitor Wired Parameters**, and set the rate to two seconds.

    c. Click on **Start Monitoring**.

    You can also monitor the configuration using the block configuration window for any individual function block. Observe the values and changes in the parameters. Check the **Periodic Updates** box to cause the parameters to be read at the specified rate when the block is running.

Tutorial 3: PID Control of a Simulated Process

This tutorial assumes you have already gone through Tutorial 2, which describes how to modify function blocks, write parameters, and test a configuration.

1. If necessary (that is, if your device supports function block instantiation), create a PID function block.
    a. Right-click on the device in the configuration tree and select **Instantiate New Block**.
    b. Select **PID Block** and click on **OK**.
    c. Right-click on the new tag and select **Set Tag**. Change the name of the tag to something like **FP-PID**.
2. Open the Function Block Application Editor window.
3. Create a PID Feedback control application using the control templates that come with the NI-FBUS Configurator.
    a. Right-click in an empty area of the Function Block Application Editor window.
    b. Select **FBAP Templates»PID Feedback Control**. A dialog box asks if you want to use the parameters saved in the template. Since this is a generic template, your function block's parameters are probably more suitable than the template's parameters. Choose **No**. This operation places a PID template on the Function Block Application Editor window. Notice the blocks are grayed out—a visual indication that the blocks are merely shells.
4. Assign function blocks to fill in the template blocks.
    a. Right-click on an AI template block, highlight **Replace With**, and select an AI function block. This block will provide the process value to the PID block.
    b. Right-click on the AO template block, highlight **Replace With**, and select an AO function block. This block is the output that will be controlled by the PID block.
    c. Right-click on the PID template block, highlight **Replace With**, and select a PID function block.

5. In the PID block configuration window, select the **Tuning** tab. Enter parameters appropriate for your process. An example is shown below:

```
GAIN            10

RESET           3.5 sec

BAL_TIME        0 sec
```

6. Set the PID scaling parameters. The PV_SCALE of the PID block (which tells the PID the range of its IN parameter) should match the OUT_SCALE of the AI block (the range of the value that the AI is passing to the PID). Likewise, the OUT_SCALE of the PID block (the range of the value that the PID is passing to the AO) should match the PV_SCALE of the AO block (which gives the AO the range of its CAS_IN parameter).

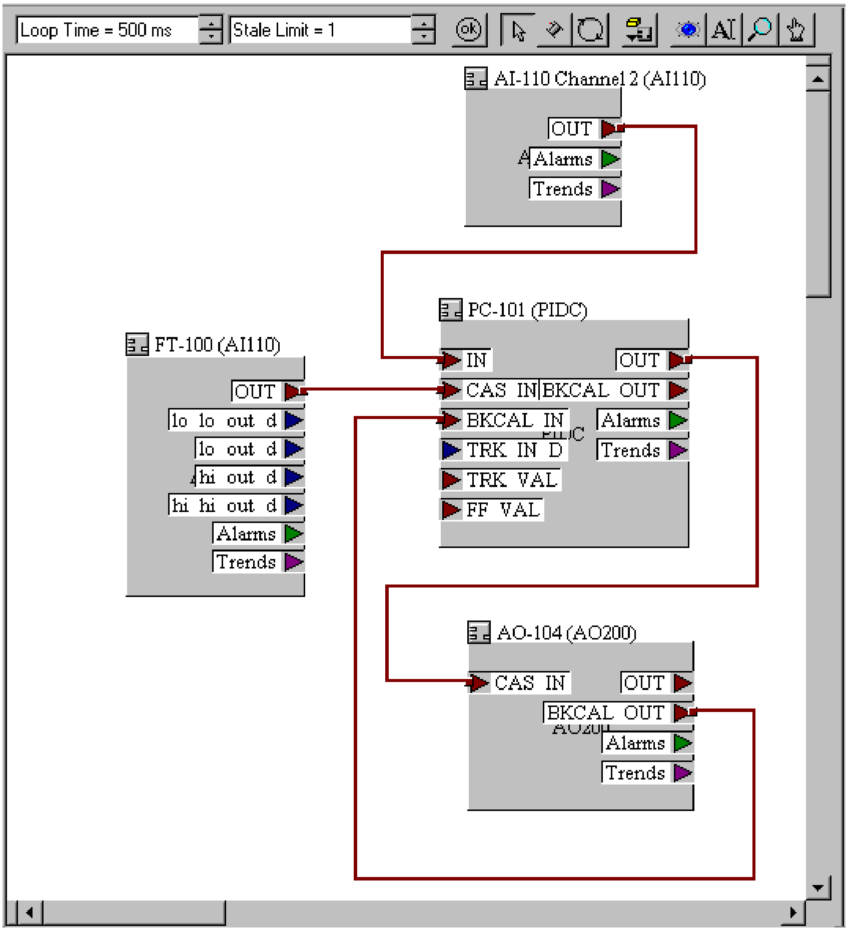
```
PV_SCALE
   EU_100       400
   EU_0         -100
   Units_Index  °C
   Decimal      3

OUT_SCALE
   EU_100       400
   EU_0         -100
   Units_Index  °C
   Decimal      3
```

7. If desired, use a second AI block to specify the setpoint of the PID.

   a. Drag another AI block onto your Function Block Application Editor window.

   b. Connect the OUT of the new AI block to the CAS_IN of the PID block. The OUT_SCALE of the new AI block should match the PV_SCALE of the PID block. Adjust the AI and PID parameters as needed. The following figure shows how the final Function Block Application Editor window should look.



Loop Time = 500 ms

Stale Limit = 1

8. Decrease the macrocycle if you have not done so already so that the loop is more responsive.
9. Download this new configuration and test by changing the setpoint through the second AI block and observing the `SP` (setpoint) and `PV` (process value) parameters of the PID function block. You may want to confirm the modes of each of the blocks for proper operation. If any of the modes are incorrectly set, first try to use the mode buttons within the function block configuration windows.

   The proper modes should be:
   - First AI —`Auto`
   - Second AI—`Auto`
   - PID—`Cas`
   - AO—`Cas`

4

Parameters

This chapter is divided into two parts. The first part explains which parameters you might need to modify, grouped by functionality. The second part of the chapter lists parameters and their descriptions alphabetically.

Alarming Parameters

Alarming parameters let you configure alarms and events. Collectively, alarms and events are referred to as alerts. You use alert notifiers to communicate the condition to other devices.

Table 4-1. Alarming Parameters

ACK_OPTION	DV_HI_PRI	LIM_NOTIFY
ALARM_HYS	DV_LO_ALM	LO_ALM
ALARM_SUM	DV_LO_LIM	LO_LIM
ALERT_KEY	DV_LO_PRI	LO_LO_ALM
BLOCK_ALM	HI_ALM	LO_LO_LIM
CONFIRM_TIME	HI_HI_ALM	LO_LO_PRI
DISC_ALM	HI_HI_LIM	LO_PRI
DISC_LIM	HI_HI_PRI	MAX_NOTIFY
DISC_PRI	HI_LIM	WRITE_ALM
DV_HI_ALM	HI_PRI	WRITE_PRI
DV_HI_LIM		

Diagnostic Parameters

Diagnostic parameters help you find problems in the operation of the device. They also provide general information about the device and its capabilities.

Table 4-2. Diagnostic Parameters

BLOCK_ALM	MEMORY_SIZE
BLOCK_ERR	MIN_CYCLE_TIME
DD_RESOURCE	MODE_BLK
DD_REV	NV_CYCLE_T
DEV_REV	RESTART
DEV_TYPE	RS_STATE
FEATURE_SEL/FEATURES	ST_REV
FREE_SPACE	TAG_DESC
FREE_TIME	UPDATE_EVT
MANUFAC_ID	

Faultstate Parameters

FOUNDATION Fieldbus output function blocks (AO, DO, and CDO) support a faultstate to deal with the case where the control of the output block has been lost while the block is in Cascade mode. If the block is in Cascade mode, and CAS_IN has a bad quality for longer than the time specified in the FSTATE_TIME parameter, the block enters faultstate. If the Faultstate Use Value option is set in the IO_OPTS parameter, the block uses the value in FSTATE_VAL as its output value. If the option is not set, it holds the value it had at the time the input went bad until the cascade can be reestablished.

The faultstate is a predefined state that outputs will go to if certain conditions are met. For example, if a valve's analog output block loses communication with the block that is giving it information, it can be set to close automatically.

Table 4-3. Faultstate Parameters

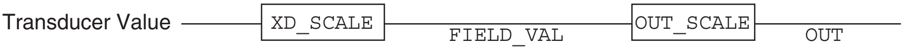
CLR_FSTATE	FSTATE_VAL_D
FAULT_STATE	SET_FSTATE
FSTATE_TIME	STATUS_OPTS
FSTATE_VAL	

Scaling Parameters

The scaling parameters differ depending on the type of function block. For example, scaling on an AI will be different from scaling on an DI. For an Analog Input (AI) block, the most important scaling parameters are:

- L_TYPE—The linearization type used in taking the physical transducer value to the block output.
- XD_SCALE—Used to take values from the physical transducer for the I/O channel to a percent of scale. This percent of scale value is called FIELD_VAL.
- OUT_SCALE—Used in combination with L_TYPE to take the percent of scale to an output value. This output value is called OUT.

The following diagrams will assist you in determining the parameters you need to change to obtain the correct scaling for AI, AO, or PID blocks. Each box indicates the application of the parameter it contains.



Range:	XD_SCALE	0 to 100%	Dependent on L_TYPE and OUT_SCALE
What it Does:	The transducer in the device produces some physical signal such as Volts, Amps, etc.	XD_SCALE specifies the transducer scaling (field range) of the physical signal. From this information, the physical signal value is converted to a percentage of the field range.	OUT_SCALE specifies the scaling from FIELD_VAL (% of scale) to output units. It is used in conjunction with L_TYPE. The type of output scaling is specified by L_TYPE. The options are no scaling (direct), linear scaling (indirect), or square root scaling (indirect square root).
Sample Settings:		For example, if the physical signal can range from –50mV to +50mV, you would enter the following: ⊟ ● XD_SCALE EU_100 50 EU_0 –50 UNITS_INDEX mV DECIMAL 3	For example, if –50mV at the transducer corresponds to a physical temperature of –100˚C, +50mV corresponds to 400˚C, and the scaling should be linear, you would enter the following: ⊟ ● OUT_SCALE EU_100 400 EU_0 –100 UNITS_INDEX ˚C DECIMAL 3 ● L_TYPE Indirect

Calculations:

$$\texttt{FIELD\_VAL} = \frac{(\texttt{transducer\_value} - \texttt{XD\_SCALE.EU0})}{\texttt{XD\_SCALE.EU100} - \texttt{XD\_SCALE.EU0}}$$

$$\texttt{OUT}_{\text{Direct}} = \texttt{transducer\_value}$$

$$\texttt{OUT}_{\text{Indirect}} = \frac{\texttt{FIELD\_VAL}}{100} * (\texttt{OUT\_SCALE.EU100} - \texttt{OUT\_SCALE.EU0}) + \texttt{OUT\_SCALE.EU0}$$

$$\texttt{OUT}_{\text{Indirect Square Root}} = \sqrt{\frac{\texttt{FIELD\_VAL}}{100}} * (\texttt{OUT\_SCALE.EU100} - \texttt{OUT\_SCALE.EU0}) + \texttt{OUT\_SCALE.EU0}$$

Figure 4-1. AI Scaling Procedure

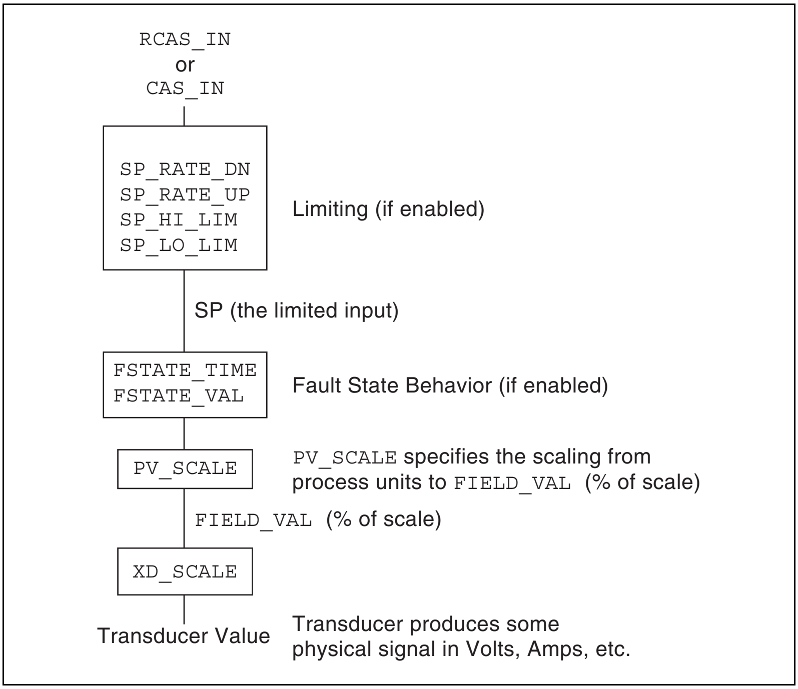
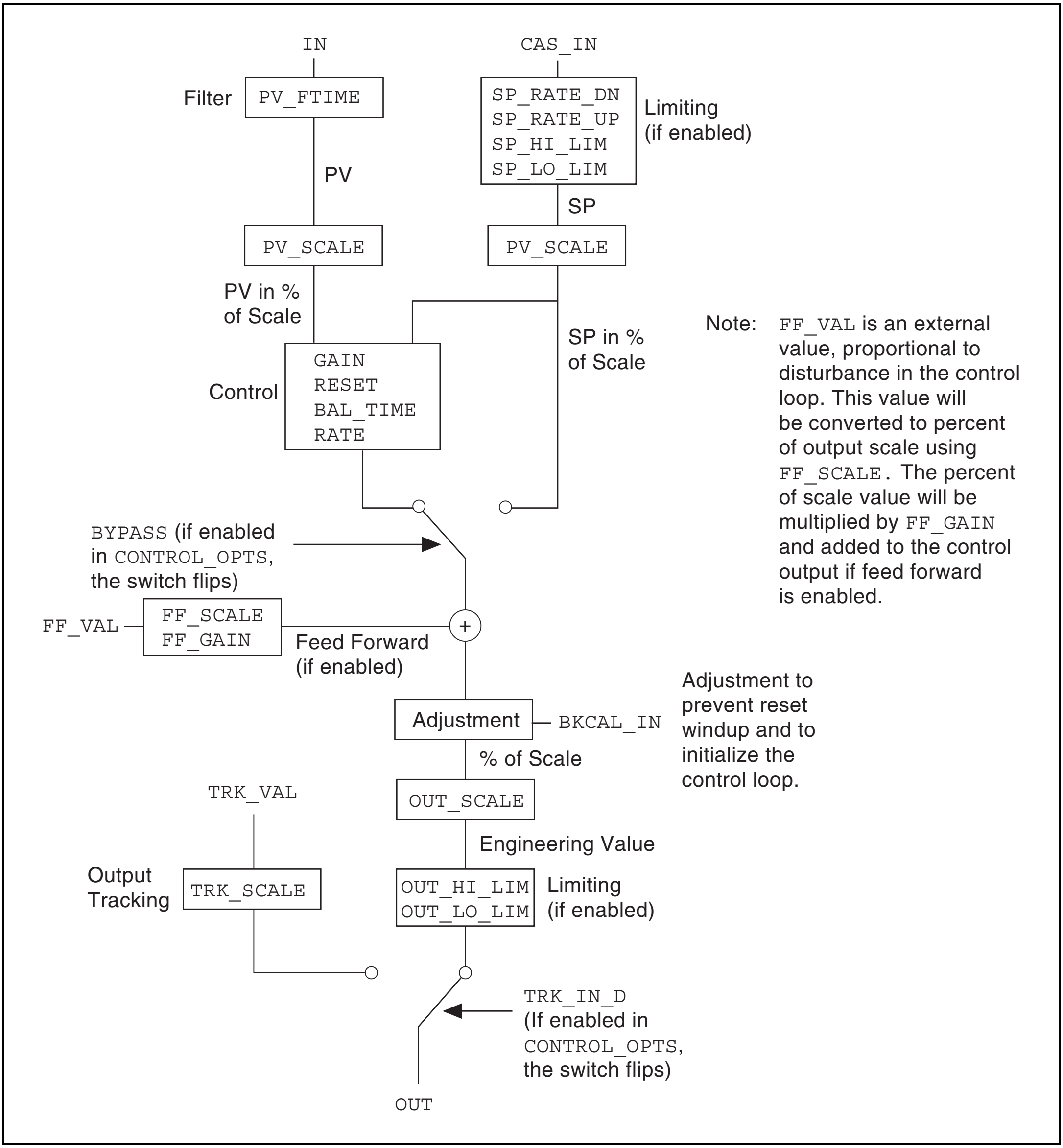


Figure 4-2. AO Scaling Procedure



Note: FF_VAL is an external value, proportional to disturbance in the control loop. This value will be converted to percent of output scale using FF_SCALE. The percent of scale value will be multiplied by FF_GAIN and added to the control output if feed forward is enabled.

Figure 4-3. PID Scaling Procedure

Table 4-4. Scaling Parameters

BYPASS	I/O_OPTS	READBACK
CONTROL_OPTS	L_TYPE	READBACK_D
FF_GAIN	LOW_CUT	SEL_1 through SEL_3
FF_SCALE	OUT	SEL_TYPE
FF_VAL	OUT_SCALE	TRK_IN_D
FIELD_VAL	PV	TRK_SCALE
FIELD_VAL_D	PV_FTIME	TRK_VAL
IN	PV_SCALE	XD_SCALE
IN_1		

I/O Parameters

I/O parameters are related to the physical I/O. The I/O parameters are:

CHANNEL
HARD_TYPES
IO_OPTS
LOW_CUT

Limiting Parameters

Limiting is to prevent a calculated value (from a PID block, for example) from rising above or dropping below a certain value.

Table 4-5. Limiting Parameters

BKCAL_HYS	SP_LO_LIM
OUT_HI_LIM	SP_RATE_DN
OUT_LO_LIM	SP_RATE_UP
SP_HI_LIM	STATUS_OPTS

Mode Shedding Parameters

Mode shedding parameters handle communication failures with a host.

For remote modes in which the cascade input is periodically written by a host application, Foundation Fieldbus provides mode shedding to handle failure of the host application. For the RCas mode and ROut modes, there are two timeout parameters in the resource block: SHED_RCAS and SHED_ROUT. If the block is in a remote mode and the block input being used (RCAS_IN or ROUT_IN) is not updated within the timeout period, the block sheds to a higher priority mode. The action taken by the block when mode shedding occurs is defined by the SHED_OPT parameter, as described in the table below. The shed condition ends when the host writes the remote input parameter. If a normal return shed option is selected, the block attempts to enter the original remote mode. If a no return shed option is selected, the target mode is changed at the time the block sheds mode, and the block does not attempt to enter the original lower priority mode when the shed condition ends.

Table 4-6. Mode Shedding Parameters

RCAS_IN	SHED_OPT
RCAS_IN_D	SHED_RCAS
ROUT_IN	SHED_ROUT

Option Parameters

Table 4-7. Option Parameters

CLR_FSTATE	SET_FSTATE
CONTROL_OPTS	SHED_OPT
FAULT_STATE	SIMULATE
FEATURE_SEL/FEATURES	SIMULATE_D
FSTATE_TIME	SP_HI_LIM
FSTATE_VAL	SP_LO_LIM
FSTATE_VAL_D	SP_RATE_DN
GRANT_DENY	SP_RATE_UP

Table 4-7. Option Parameters (Continued)

IO_OPTS	STATUS_OPTS
LOW_CUT	WRITE_LOCK
RESTART	WRITE_PRI

Parameters That Must Be Initialized

Some parameters must be initialized to a legal value before the configuration can be downloaded without errors. This is because some parameters are considered too important to allow a user to overlook them.You must enter a legal value for these parameters.

ALERT_KEY
L_TYPE

Process Parameters

Table 4-8. Process Parameters

BKCAL_IN	OUT_STATE
BKCAL_OUT	PV
BKCAL_OUT_D	PV_D
CAS_IN	PV_STATE
CAS_IN_D	RCAS_IN
CHANNEL	RCAS_IN_D
FF_VAL	RCAS_OUT
FIELD_VAL	RCAS_OUT_D
FIELD_VAL_D	ROUT_IN
FREE_SPACE	ROUT_OUT
FREE_TIME	RS_STATE
HARD_TYPES	SEL_1 through SEL_3
IN	SEL_TYPE
IN_1	SP

Table 4-8. Process Parameters (Continued)

MIN_CYCLE_T	SP_D
MODE_BLK	TEST_RW
OUT	XD_STATE
OUT_D	

Tuning Parameters

Tuning parameters let you adjust the control settings.

Table 4-9. Tuning Parameters

BAL_TIME	IN_1
BIAS	LOW_CUT
BYPASS	OUT
CYCLE_SEL	PV
CYCLE_TYPE	PV_FTIME
FF_GAIN	RA_FTIME
FF_VAL	RATE
FIELD_VAL	READBACK
FIELD_VAL_D	READBACK_D
GAIN	RESET
IN	SEL_1 through SEL_3

Parameter Descriptions

ACK_OPTION (Alarming)

Allows alarms to be automatically acknowledged by the block with no outside intervention. This is useful if you are not interested in acknowledging certain alarms from a block.

ALARM_HYS (Alarming)

The amount a value must move off an alarm limit, in percent of scale, for the alarm to be considered clear. This helps prevent alarms from constantly “toggling” on and off when the process value is near the configured alarm limit.

ALARM_SUM (Alarming)

A summary of the status of alarms in the block. Allows alarms to be disabled.

ALERT_KEY (Alarming)

A user-assigned identification number reported in alarm messages from the block that allows HMI applications to sort and filter alarms and events. This parameter is set for each function block to indicate which physical unit the function block is associated with.

BAL_TIME (Tuning)

Time, in seconds, for the bias or ratio to change from the internal working value to the operator set value. Also, the time constant used by the integral term of the PID to obtain balance when the output is limited and the block is in `Auto`, `Cas`, or `RCas` mode.

BIAS (Tuning)

The bias value, in engineering units, used to calculate the function block output.

BKCAL_HYS (Limiting)

The amount a block’s output value must move off a limit, in percent of scale, for the limit status to be turned off.

BKCAL_IN (Limiting, Process)

Will be linked to a downstream block’s `BKCAL_OUT` or `SELECT_OUT` parameter. This is used to initialize a control loop through cascade initialization. Cascade initialization allows smooth transfer for a control block from manual to automatic mode. To bypass cascade initialization, this parameter can be left unwired and manually set to a status of `Good, non-cascade`.

BKCAL_OUT (Process)

A back-calculation value published to the `BKCAL_IN` of an upstream block in a control loop. The `BKCAL_OUT` parameter has the value of the block's current output. Before a cascade loop is initialized, the upstream block can use this value to smoothly transfer to loop control.

BKCAL_OUT_D (Process)

An output value published to an upstream discrete block. The upstream block can use this value to smoothly transfer to loop control.

BLOCK_ALM (Alarming, Diagnostic)

An alarm parameter used to report error conditions detected within the block, such as block Out of Service.

BLOCK_ERR (Diagnostic)

A list of error conditions for hardware and software components associated with the block.

Table 4-10. Error Codes

Error Code	Code	Description
Other	0x0001	Undefined block error condition.
Block Configuration Error	0x0002	The block has detected an error in its configuration. This usually indicates a static parameter has been left uninitialized.
Link Configuration Error	0x0004	The logical connection between this block and another block is misconfigured.
Simulate Active	0x0008	For I/O function blocks, this indicates that simulation is enabled. For the resource block, this indicates that the simulate jumper has been set, allowing simulation to be enabled in other blocks.
Local Override	0x0010	The block has locally overridden the output value. This can be the result of an interlock or faultstate.
Device Faultstate Set	0x0020	The block's faultstate behavior is active.

Table 4-10. Error Codes (Continued)

Error Code	Code	Description
Device Needs Maintenance Soon	0x0040	The device is reporting performance degradation that will soon require maintenance.
Input Failure/BAD PV Status	0x0080	Either the input transducer channel has reported a failure, or the input parameter from an upstream block has reported a failure. For an AI block, this could be caused by an open circuit being detected on the FP-AI-100 input module.
Output Failure	0x0100	The output transducer channel has reported a failure. For an AO block, this could indicate that the FP-AO-200 cannot drive the current request, perhaps due to an open circuit.
Memory Failure	0x0200	The storage for nonvolatile and static parameters was corrupted.
Lost Static Data	0x0400	The device was unable to restore the values of static parameters after a restart.
Lost NV Data	0x0800	The device was unable to restore the values of nonvolatile parameters after a restart.
Readback Check Failed	0x1000	The value read back from the output channel does not match the value the output channel was set to.
Device Needs Maintenance Now	0x2000	The device needs to be maintained now.
Power-Up	0x4000	The device has just powered up.
Out of Service	0x8000	The block is currently out of service.

BYPASS (Scaling, Tuning)

Allows the normal control algorithm to be bypassed. If control is bypassed, the PID uses its setpoint value in percent of scale as its output value and does not attempt to do any PID control.

Note The `CONTROL_OPTS Bypass Enable` option must be selected.

CAS_IN (Process)

A remote setpoint value. If used, will be linked to an output of an upstream block. In some blocks, may be left unlinked, and a local setpoint value (typically SP) will be used instead.

CAS_IN_D (Process)

A remote setpoint value. If used, will be linked to an output of an upstream block. In some blocks, may be left unlinked, and a local setpoint value (typically SP_D) will be used instead.

CHANNEL (I/O, Process)

Used by I/O function blocks to select a physical I/O channel. This mapping is defined by the manufacturer. In the FP-3000, this parameter is automatically updated when the FIELDPOINT_MODULE and FIELDPOINT_CHANNEL parameters are modified.

CLR_FSTATE (Faultstate, Option)

Writing Clear to this parameter causes the device-wide faultstate to be cleared and output blocks to resume normal execution. Also see SET_FSTATE and FAULT_STATE parameters.

CONFIRM_TIME (Alarming)

The lower bound on the time the device waits to send alert report messages if no confirmation is received from a host.

CONTROL_OPTS (Option, Scaling)

A list of options used to adjust the way control blocks, such as the PID block, operate.

Table 4-11. Control Options

Options	Description
Bypass Enable	If set, lets you set the BYPASS parameter and bypass the algorithm’s control.
SP-PV Track in Man	Causes the setpoint to track the process variable in Man.
SP-PV Track in ROut	Causes the setpoint to track the process variable in ROut.
SP-PV Track in LO or IMan	Causes the setpoint to track the process variable in LO or IMan.

Table 4-11. Control Options (Continued)

Options	Description
`SP Track Retained Target`	Causes the setpoint to track the input value of the retained target of the block. The retained target of the block is the lowest priority mode set in the target mode field of the `MODE_BLK` parameter of the block. For example, if the `RCas` bit is set in the target mode, the setpoint tracks `RCAS_IN`.
`Direct Acting`	Defines the relationship between changes to the `PV` and changes to the output. For example, consider a case with a fixed `SP` while the process variable varies. When `Direct Acting` is set, an increase in the process variable causes the control block's output value to be increased. When `Direct Acting` is clear, an increase in the process variable causes the control block's output value to be decreased.
`Track Enable`	Enables external tracking. If `Track Enable` is true, and `TRK_IN_D` is true, `TRK_VAL` overwrites the value at the output of the block except when `Man` is the target mode.
`Track in Manual`	Enables tracking in Manual mode.
`Use PV for BKCAL_OUT`	When set, uses the process variable as the value for `BKCAL_OUT`, instead of the setpoint.
`Obey SP Limits if Cas or RCas`	When set, confines the setpoint to values within `SP_HI_LIM` and `SP_LO_LIM`, even when the setpoint comes from another function block.
`No OUT Limits in Man`	Unused in FieldPoint.

CYCLE_SEL (Tuning)

Identifies the block execution methods available. Unused in National Instruments FP-3000.

CYCLE_TYPE (Tuning)

Used to select the block execution method. Unused in National Instruments FP-3000.

DD_RESOURCE (Diagnostic)

Unused in FieldPoint.

DD_REV (Diagnostic)

The revision of the device description used by the device.

DEV_REV (Diagnostic)

The revision of the device.

DEV_TYPE (Diagnostic)

The manufacturer’s model number for the device.

DISC_ALM (Alarming)

The current state of the discrete alarm, along with a time and date stamp.

DISC_LIM (Alarming)

The discrete input state in which an alarm should be generated.

DISC_PRI (Alarming)

Priority of the discrete alarm.

DV_HI_ALM (Alarming)

The current state of the deviation high alarm, along with a time and date stamp.

DV_HI_LIM (Alarming)

The deviation limit between the PID block setpoint and process value, in engineering units, beyond which the deviation high alarm is considered active.

DV_HI_PRI (Alarming)

The priority of the deviation high alarm.

DV_LO_ALM (Alarming)

The current state of the deviation low alarm, along with a time and date stamp.

DV_LO_LIM (Alarming)

The deviation limit between the PID block setpoint and process value, in engineering units, beyond which the deviation low alarm is considered active.

DV_LO_PRI (Alarming)

The priority of the deviation low alarm.

FAULT_STATE (Faultstate, Option)

The current status of the device faultstate. It can be set and cleared with `SET_FSTATE` and `CLR_FSTATE`. If it is set, all output blocks in the device initiate their own faultstate behavior.

FEATURE_SEL/FEATURES (Diagnostic, Option)

The `FEATURES` parameter lists features supported by the device. Use the `FEATURE_SEL` parameter to manually enable and disable the supported features listed in the `FEATURES` parameter.

Table 4-12. Feature Parameter Options

Option	Description
`Unicode`	The device supports strings in `Unicode` format. The FP-3000 does not support this feature.
`Reports`	The device supports event report messages for alarming. If this feature is not selected in the `FEATURE_SEL` parameter, the FP-3000 continues to detect alarms and events, but does not report them over the bus. In this case, the host must poll the alarm parameters to detect alarm conditions as they change.
`Faultstate`	The device supports `Faultstate` behavior for output blocks.
`Soft Write Lock`	The device supports locking of configuration of parameters with the `WRITE_LOCK` parameter in the resource block. With this feature selected and the `WRITE_LOCK` parameter written to `Set`, writes to all static configuration parameters are disallowed.
`Hard Write Lock`	The device supports locking of configuration parameters. For the FP-3000, a switch on the back of the device must also be set. If `Hard Write Lock` is enabled, the switch disallows writes to all configuration parameters in the device, including `FEATURE_SEL`.

Table 4-12. Feature Parameter Options (Continued)

Option	Description
`Out Readback`	The device provides a way for the action of output transducers to be verified through a readback. The FP-3000 does not support this feature.
`Direct Write`	The device provides a manufacturer-specific way to directly write to I/O channels. The FP-3000 does not support this feature.

FF_GAIN (Scaling, Tuning)

The gain by which the feed-forward input is multiplied before it is added to the output value of the control block.

FF_SCALE (Scaling)

The scaling parameter used by the feed-forward value of the block.

FF_VAL (Process, Scaling, Tuning)

The feed-forward value.

FIELD_VAL (Process, Scaling, Tuning)

The value from the input channel, in percent of scale.

FIELD_VAL_D (Process, Scaling, Tuning)

The value from the discrete input channel.

FREE_SPACE (Diagnostic, Process)

The percentage of memory available on the device. This can be used when instantiating (creating) blocks to determine the remaining capacity of the FP-3000. This value will be zero in preconfigured devices since they do not allow user configuration.

FREE_TIME (Diagnostic, Process)

Percentage of block processing time that is available to process additional blocks. Unused in FieldPoint.

FSTATE_TIME (Faultstate, Option)

Time (in seconds) to delay from the detection of loss of communications with the host for the output block remotes setpoint until the enaction of the fault state output.

FSTATE_VAL (Faultstate, Option)

The setpoint value to be used on failure.

Note The I/O option **Failsafe to value** must be selected.

FSTATE_VAL_D (Faultstate, Option)

The discrete setpoint value to be used on failure.

Note The I/O option **Failsafe to value** must be selected.

GAIN (Tuning)

The gain constant used by the PID in calculating the proportional component of the output.

GRANT_DENY (Option)

Allows HMI applications to determine access privileges for block parameters.

Note The device does not use this parameter to restrict parameter access itself. It is only for the benefit of host applications.

HARD_TYPES (I/O, Process)

A list of available channel types. As I/O modules are inserted and removed from the FP-3000 bank, bits in this field change to reflect the presence or absence of types of I/O channels.

Table 4-13. Hard Types

Bitmask	Description
`Analog Input`	This bit is set if the FP-3000 has analog input channels available.
`Analog Output`	This bit is set if the FP-3000 has analog output channels available.

Table 4-13. Hard Types (Continued)

Bitmask	Description
`Discrete Input`	This bit is set if the FP-3000 has discrete input channels available.
`Discrete Output`	This bit is set if the FP-3000 has discrete output channels available.

HI_ALM (Alarming)

The current state of the high alarm, along with a time and date stamp.

HI_HI_ALM (Alarming)

The current state of the high-high alarm, along with a time and date stamp.

HI_HI_LIM (Alarming)

The limit, in PV units, beyond which the high-high limit alarm is considered active.

HI_HI_PRI (Alarming)

The priority of the high-high limit alarm.

HI_LIM (Alarming)

The limit, in PV units, beyond which the high limit alarm is considered active.

HI_PRI (Alarming)

The priority of the high limit alarm.

IN (Process, Scaling, Tuning)

The primary input of the block.

IN_1 (Process, Scaling, Tuning)

The secondary input of the block.

IO_OPTS (I/O, Options, Scaling)

A bitmask used to adjust the way I/O blocks (AI, DI, AO, and DO) operate.

Table 4-14. Operation Bitmasks

Bitmask	Description
`Invert`	In discrete blocks, this maps a physical state of `Discret_State_0` to `Discret_State_1` and maps every other physical transducer state to `Discret_State_0`.
`SP-PV Track in Man`	Causes the setpoint to track the process variable in `Man`.
`SP-PV Track in LO or IMan`	Causes the setpoint to track the process variable in `LO` or `IMan`.
`SP Track Retained Target`	Causes the setpoint to track the input value of the retained target of the block. The retained target of the block is the lowest priority mode set in the target mode field of the `MODE_BLK` parameter of the block. For example, if the `RCas` bit is set in the target mode, the setpoint tracks `RCAS_IN`.
`Increase to Close`	Remaps the block's scaling so that as the input increases, the output decreases.
`Faultstate to Value`	When set, the block's faultstate behavior sets the output value to the value in `FSTATE_VAL`. When clear, the block's faultstate behavior leaves the output value at its current setting.
`Use Faultstate Value on Restart`	When set, causes the output value of output blocks to go to faultstate value immediately after a device restart. When clear, uses the value in nonvolatile memory.
`Target to Man if Faultstate Active`	When set, sets the target mode of the block to manual mode when faultstate goes active.
`Use PV for BKCAL_OUT`	When set, uses the process variable as the value for `BKCAL_OUT`, instead of the setpoint.
`Low Cutoff`	When set, enables the AI low cutoff parameter.

ITK_VER

The version of the Interoperability Test Kit with which this device was tested.

L_TYPE (Scaling)

The linearization type. This parameter affects the way the value from the transducer is linearized in the analog input block before it is presented as the block output. In all cases, the FIELD_VAL parameter behaves as follows:

```
FIELD_VAL = 100 * (transducer_value - XD_SCALE.EU0) /
(XD_SCALE.EU100 - XD_SCALE.EU0)
```

FIELD_VAL can be simply described as the percentage of span reading from the transducer, and therefore its units are percent.

Table 4-15. Linearization Types

Type	Description
Direct	The block output is directly taken from the transducer value: OUT = transducer_value
Indirect	The block output is scaled according to OUT_SCALE from the value in FIELD_VAL: OUT = OUT_SCALE.EU0 + ((FIELD_VAL/100) * (OUT_SCALE.EU100 - OUT_SCALE.EU0))
Indirect Square Root	The block output is scaled according to OUT_SCALE from the value in FIELD_VAL. Before the field value is rescaled, the square root is taken. OUT = OUT_SCALE.EU0 + (SQRT(FIELD_VAL / 100) * (OUT_SCALE.EU100 - OUT_SCALE.EU0))
Uninitialized	An invalid setting. The device reports a configuration error with an Uninitalized L_TYPE.

LIM_NOTIFY (Alarming)

A limit on the number of unconfirmed alarm/event notification messages the device can have active at once. This must be less than or equal to MAX_NOTIFY.

LO_ALM (Alarming)

The current state of the low alarm, along with a time and date stamp.

LO_LIM (Alarming)

The limit, in PV units, beyond which the low limit alarm is considered active.

LO_LO_ALM (Alarming)

The current state of the low-low alarm, along with a time and date stamp.

LO_LO_LIM (Alarming)

The limit, in PV units, beyond which the low-low limit alarm is considered active.

LO_LO_PRI (Alarming)

The priority of the low-low limit alarm.

LO_PRI (Alarming)

The priority of the low limit alarm.

LOW_CUT (I/O, Option, Scaling, Tuning)

With an `L_TYPE` of Indirect Square Root, this can be used to establish a floor (in percent of scale) for values from the transducer. Values below this floor are considered to be zero. This feature must first be enabled by setting `Low Cutoff` in the `IO_OPTS` parameter.

Note The `IO_Opts` parameter `Low Cutoff` must be enabled.

MANUFAC_ID (Diagnostic)

The ID of the manufacturer of the device. For National Instruments devices, it is 0x4E4943. The parameters `MANUFAC_ID`, `DEV_TYPE`, `DEV_REV`, and `DD_REV` are used in combination for a host tool to locate the Device Description for this device.

MAX_NOTIFY (Alarming)

The maximum number of unconfirmed alarm/event notification messages the device supports.

MEMORY_SIZE (Diagnostic)

Unused by FieldPoint.

MIN_CYCLE_T (Diagnostic, Process)

The length of the shortest macrocycle the device supports.

MODE_BLK (Diagnostic, Process)

Sets the operational and permitted modes of the block. The following table describes the operational and permitted modes of the block.

Table 4-16. Field Modes

Field Mode	Description
TARGET	The desired mode of operation of the block. This field is writable. Several bits may be set in this field, and typically, the highest priority bit that is set will be considered to be the target mode. OOS is the highest priority bit.
ACTUAL	A bit reflecting the current state of operation of the block. This is a read-only field. Only one bit will be set at a time by the block. ACTUAL mode is a function of the target mode and the current conditions in which the block is executing. Several conditions (such as cascade initialization or fault state conditions) can cause the ACTUAL mode to differ from the TARGET mode.
PERMITTED	A bitmask indicating which modes are permitted target modes and which are not. This field is writable. This could be used by the plant operator to disallow certain modes the block would normally be permitted to have as a TARGET mode.
NORMAL	Not used by the block. This can be used by an operator to store the normal mode of operation for the block in normal plant operations. This field is writable. Used by the NI-FBUS Configurator to set the TARGET mode after download when the Automatic Mode Handling feature is in effect.

Table 4-17. Operational Modes

Operational Mode	Description
Out of Service (OOS)	The block is out of service, block execution is suspended, and all output parameters take a status of Bad::OutOfService.
Initialization Manual (IMan)	The block is in the process of initializing a cascade. This is used for upstream (control) blocks when they are initializing for smooth transfer into Automatic mode. This is not a valid TARGET mode, but it is a valid ACTUAL mode.
Local Override (LO)	Faultstate or an interlock is active and causing the output value of the block to be overridden. This is not a valid TARGET mode, but is a valid ACTUAL mode.
Manual (Man)	The output value of the block is set by the user.
Automatic (Auto)	The output value of the block is set by the block algorithm, and the block is using a local value for its setpoint.
Cascade (Cas)	The setpoint for the block is taken from the CAS_IN parameter, which is typically connected to the output of another block. This mode cannot be entered before cascade initialization takes place. When Cascade is desired as a TARGET mode, the Auto bit is also set in the target.
Remote Cascade (RCas)	Like Cascade mode, in Remote Cascade mode the setpoint of the block comes from an outside data source. Unlike Cascade mode, in Remote Cascade mode the setpoint is sourced from the *RCAS_IN* parameter, which is written by a host application and not another function block.
Remote Output (ROut)	Remote Output mode is analogous to Remote Cascade mode, except that the remote host application directly sets the output of the block and not the setpoint. In the case of an analog output block, this bypasses setpoint rate and absolute limiting.

NV_CYCLE_T (Diagnostic)

The regular time interval, in milliseconds, at which nonvolatile parameters are committed to nonvolatile storage. A value of zero means that the parameters are never written to nonvolatile memory. Note that nonvolatile parameters are stored to nonvolatile memory when they are changed by a user over the network. The NV_CYCLE_T parameter sets the rate at which changes caused by the device itself are stored to nonvolatile memory.

OUT (Process, Scaling, Tuning)

The current output value of the block.

OUT_D (Process)

The current output value of a discrete block.

OUT_HI_LIM (Limiting)

A limit for the maximum output value from a block in modes other than manual.

OUT_LO_LIM (Limiting)

A limit for the minimum output value from a block in modes other than manual.

OUT_SCALE (Scaling)

The scaling parameter used for the output parameter.

Table 4-18. OUT_SCALE Parameter

Subfield	Meaning
`EU_100`	Engineering units value at 100 percent of scale.
`EU_0`	Engineering units value at zero percent of scale.
`UNIT_INDEX`	Actual engineering units code (such as mA).
`DECIMAL`	Number of digits a host shows to the right of the decimal for display purposes (not used by the NI-FBUS Configurator).

OUT_STATE (Process)

Index to the text description of the discrete output state.

PV (Process, Scaling, Tuning)

The process value, or primary value for this block. For AI and control blocks such as PID, this represents a measurement of the state of the process (such as temperature or level). For AO blocks, the process value is the current setpoint of the block.

PV_D (Process)

The process value, or primary value for this block. For DI and discrete control blocks, this represents a measurement of the discrete state of the process. For DO blocks, the process value is the current discrete setpoint of the block.

PV_FTIME (Scaling, Tuning)

The filter time, in seconds, used in input blocks. For analog blocks, it is the time constant for a low pass exponential filter used to damp out rapid oscillations in the input value before using it as the process variable. For discrete blocks, it is the time the PV must remain constant after a change for the change to be reported.

PV_SCALE (Scaling)

The scaling parameter used by the process variable of the block. Converts from percent of scale to a process variable in engineering units. Contains the same subfields as `OUT_SCALE`.

PV_STATE (Process)

Index to the text describing the state of a discrete PV.

RA_FTIME (Tuning)

The filter time constant, in seconds, for the value to be used in the ratio.

RATE (Tuning)

The time constant for the derivative component of the PID block. A zero disables the derivative term. The units are seconds.

RCAS_IN (Mode Shedding, Process)

The cascade input for a control or output block set by a remote host. This is propagated to the setpoint of the block when it is in `RCas` mode. If the block is in `RCas` mode and this parameter is not updated in `SHED_RCAS` time (a parameter in the resource block), the block enters mode shedding. Mode shedding allows the block to degrade from `RCas` mode into some higher priority mode.

RCAS_IN_D (Mode Shedding, Process)

The discrete cascade input for a control or output block set by a remote host. This is propagated to the setpoint of the block when it is in `RCas` mode. If the block is in `RCas` mode and this parameter is not updated in `SHED_RCAS` time (a parameter in the resource block), the block enters mode shedding. Mode shedding allows the block to degrade from `RCas` mode into some higher priority mode.

RCAS_OUT (Process)

The back calculation output used by the supervisory host when establishing a Remote cascade loop.

RCAS_OUT_D (Process)

The discrete back calculation output used by the supervisory host when establishing a Remote cascade loop.

READBACK (Scaling, Tuning)

The valve or actuator position read back from the transducer, in transducer units.

READBACK_D (Scaling, Tuning)

The transducer state for the actual discrete valve or actuator position.

RESET (Tuning)

The time constant for the integral component of the PID block. It is measured in seconds per repeat (so larger values have less effect, and `INF` effectively disables the integral term).

RESTART (Diagnostic, Option)

Allows the user to restart the device remotely.

Table 4-19. Restart Values

Value	Behavior
`Restart Resource`	Restarts the device.
`Restart to Defaults`	Restarts the device, restoring all parameter values to default values.
`Restart Processor`	Restarts the device as if the power was cycled.

⚠ **Caution** Using `Restart to Defaults` causes all your configured parameters in the FP-3000 to revert to their factory default settings.

ROUT_IN (Mode Shedding, Process)

The cascade input set by a remote host. This is propagated to the output of the block when it is in `ROut` mode. If the block is in `ROut` mode and this parameter is not updated in `SHED_ROUT` time (a parameter in the resource block), the block enters mode shedding. Mode shedding allows the block to degrade from `ROut` mode into some higher priority mode.

ROUT_OUT (Process)

This is the back calculation output used by the host when trying to establish a remote output loop. While the loop is being established, it is the current value of the output channel and can be used by the host to initialize for smooth transfer of control.

RS_STATE (Diagnostic, Process)

The current state of the device.

Table 4-20. Device States

State	Meaning
`Start/Restart`	The device has just started a restart cycle.
`Initialization`	The device is performing startup diagnostics.
`Failure`	A hardware failure has been detected.
`On-Line Linking`	The device is online and waiting for new parameter linkages to be established.
`On-Line`	The device is online and in service.
`Standby`	The device is online, but currently out of service.

SEL_1 through SEL_3 (Process, Scaling, Tuning)

Input values for the selector.

SEL_TYPE (Scaling)

Defines the selector action—High, Medium, or Low.

SET_FSTATE (Faultstate, Option)

Allows the user to set the device faultstate to active. This, in turn, forces all output blocks into their own faultstate behavior.

SHED_OPT (Mode Shedding, Option)

Controls the way blocks enter mode shedding. Each option listed below has a companion `No Return` option. The `No Return` shedding options change the target mode of the device to the shed mode and prevent the device from re-entering `RCas` or `ROut` mode after the shed condition has ended.

Table 4-21. Shed Conditions

Shed Mode	Behavior
`Normal Shed`	The block sheds into the next higher-priority mode set in the permitted mode field of `MODE_BLK`.
`Shed to Auto`	The block sheds into automatic mode.
`Shed to Manual`	The block sheds into manual mode.
`Shed to Retained`	The block sheds to the next higher priority mode set in the target mode field of `MODE_BLK`.

SHED_RCAS (Mode Shedding)

The shed time for the `RCAS_IN` parameter. If the block is in `RCas` mode and the `RCAS_IN` parameter has not been updated in `SHED_RCAS` time, the block performs mode shedding as determined by the `SHED_OPT` parameter.

SHED_ROUT (Mode Shedding)

The shed time for the `ROUT_IN` parameter. If the block is in `RCas` mode and the `ROUT_IN` parameter has not been updated in `SHED_RCAS` time, the block performs mode shedding as determined by the `SHED_OPT` parameter.

SIMULATE (Option)

Used to bypass the physical I/O channel and allow the block to operate normally, using a simulated I/O channel.

Note For this feature to be enabled on an FP-3000, you must set a switch on the back of the device. To see how to configure the switch, refer to the FP-3000 manual.

SIMULATE_D (Option)

Used to bypass the physical I/O channel and allow the block to operate normally, using a simulated discrete I/O channel.

Note For this feature to be enabled on an FP-3000, you must set a switch on the back of the device. To see how to configure the switch, refer to the FP-3000 manual.

SP (Process)

The analog setpoint.

SP_D (Process)

The discrete setpoint.

SP_HI_LIM (Limiting, Option)

The upper limit on an operator-entered setpoint for the block. If the operator enters a setpoint that exceeds this value, the setpoint is considered to be `SP_HI_LIM` with a status that indicates that it is limited.

SP_LO_LIM (Limiting, Option)

The lower limit on an operator-entered setpoint of the block. If the operator enters a setpoint below this value, the setpoint is considered to be `SP_LO_LIM` with a status that indicates that it is limited.

SP_RATE_DN (Limiting, Option)

In `Auto` mode, the rate, in PV units per second, the setpoint can be moved downwards. If the setpoint moves faster than `SP_RATE_DN`, the block acts as if the setpoint is moving downwards at the maximum rate with a status bit that indicates that it is limited. If set to zero, the set point is used immediately.

SP_RATE_UP (Limiting, Option)

In `Auto` mode, the rate, in PV units per second, the setpoint can be moved upwards. If the setpoint moves faster than `SP_RATE_UP`, the block acts as if the setpoint is moving upwards at the maximum rate with a status bit that indicates that it is limited. If set to zero, the set point is used immediately.

ST_REV (Diagnostic)

`ST_REV` is incremented by one each time a static parameter is modified.

STATUS_OPTS (Faultstate, Limiting, Option)

A collection of options that affect the status behavior of the block.

Table 4-22. Status Options

Option	Meaning
IFS if Bad IN	Set the status of the block output to initiate faultstate if the *IN* parameter goes bad.
IFS if Bad CAS_IN	Set the status of the block output to initiate faultstate if the CAS_IN parameter goes bad.
Use Uncertain as Good	If set, blocks will treat the Uncertain status on an input parameter as if it were a Good status. If clear, Uncertain status is treated as Bad.
Propagate Failure Forward	If the status of the IN parameter of the block is Bad::Device_Failure or Bad::Sensor_Failure, the failure will be propagated to the OUT parameter. No alarm will be generated.
Propagate Failure Backward	If the status at BKCAL_IN or from the physical I/O channel is bad, the failure will be propagated to the BKCAL_OUT parameter. No alarm will be generated.
Target to Manual if Bad IN	Set the target mode of the block to Man if the IN parameter has a bad status.
Uncertain if Limited	For input or calculation blocks, the output status will be set to Uncertain if the transducer or calculated value is limited (that is, at its high or low limit).
Bad if Limited	Set the output status to Bad if the transducer value is limited (that is, at its high or low limit).
Uncertain if Manual Mode	Set the output status to Uncertain if the block is in Man mode.
Do Not Select if Not Auto Mode	Set the output status to Do Not Select if the block is not in an actual mode of Auto, Cas or RCas and not initializing. This is useful for blocks upstream of the selector block.
Do Not Select if Not Cas Mode	Set the output status to Do Not Select if the block is not in an actual mode of Cas or RCas and is not initializing. This is useful for blocks connected to a selector block.

STRATEGY

Used to identify groupings of blocks. Not used by the block itself.

TAG_DESC (Diagnostic)

User description for the purpose of the block.

TEST_RW (Process)

Unused by the block algorithm. Used to test interoperability of reads and writes of different parameter types.

TRK_IN_D (Scaling)

Used to enable tracking of the output value to `TRK_VAL`. When this is enabled, the output value of the block takes on the value specified in `TRK_VAL`.

TRK_SCALE (Scaling)

The scaling parameter used for the value specified by `TRK_VAL`.

TRK_VAL (Scaling)

The value the block will track when tracking is enabled by `TRK_IN_D`.

UPDATE_EVT (Diagnostic)

The current state of the update event, along with a time and date stamp. This event is issued whenever a static parameter is changed and `ST_REV` is incremented. The index information for the parameter that changed and the new value of `ST_REV` is included in the alert.

WRITE_ALM (Alarming)

State of the alert generated if `WRITE_LOCK` is cleared, along with a time and date stamp.

WRITE_LOCK (Option)

The software write lock for the device. When this is set to true, writes to all configuration parameters of all blocks are disallowed. The `WRITE_ALM` block alarm is active when writes are allowed and clear when they are disallowed.

WRITE_PRI (Alarming, Option)

The priority of the write alarm.

XD_SCALE (Scaling)

The scaling parameter used to interpret values from the physical I/O channel. This is used to translate from a physical transducer value to a percent of scale.

Table 4-23. Scaling Parameter Values

Subfield	Meaning
`EU_100`	Engineering units value at 100 percent of scale.
`EU_0`	Engineering units value at zero percent of scale.
`UNIT_INDEX`	Actual engineering units code (such as mA).
`DECIMAL`	Number of digits a host shows to the right of the decimal for display purposes. (Not used by the NI-FBUS Configurator.)

XD_STATE (Process)

Index to the text description of the transducer state.

A

Error Messages and Warnings

This appendix lists error messages you might receive while using the NI-FBUS Configurator and describes the error messages.

Error Messages

If you receive an error message while using the NI-FBUS Configurator, refer to this section for a description of the error and possible solutions.

NIF_ERR_ALARM_ACKNOWLEDGED

The alarm has already been acknowledged.

NIF_ERR_BAD_ARGUMENT

The value you gave is not of the correct data type.

NIF_ERR_BAD_DEVICE_DATA

The device returned inconsistent information.

NIF_ERR_BUF_TOO_SMALL

The buffer does not contain enough entries to hold all the information for the blocks. If you receive this error, buffer entries that you allocated did not contain valid block information when the call returned.

NIF_ERR_COMM_ERROR

An error occurred when NI-FBUS tried to communicate with the device.

NIF_ERR_CONFIG_ERROR

Some configuration information, such as Windows registry information or network configuration information, is incorrect.

NIF_ERR_DATA_NEVER_WRITABLE

The specified object is read-only.

NIF_ERR_DEVICE_NOT_PRESENT

The requested device is not operational.

NIF_ERR_EXCEED_LIMIT

The device reported that the value exceeds the allowable limit during a write operation.

NIF_ERR_INVALID_DESCRIPTOR

The descriptor you gave is invalid.

NIF_ERR_MULTIPLE

There are duplicate block tags.

NIF_ERR_NOT_FOUND

NI-FBUS could not find the specified object.

NIF_ERR_OBJECT_ACCESS_DENIED

The Fieldbus interface does not have the required privileges. The access group you belong to is not allowed to acknowledge the event, or the password you entered is incorrect.

NIF_ERR_OBJECT_ACCESS_UNSUPPORTED

The device does not support write access to this object.

NIF_ERR_OBSOLETE_DESC

The input descriptor is no longer valid. It was closed before a function completed.

NIF_ERR_ORDINAL_NUM_OUT_OF_RANGE

The ordinal number is out of the device's range.

NIF_ERR_PARAMETER_CHECK

The device reported a violation of parameter-specific checks.

NIF_ERR_RESOURCES

NI-FBUS is unable to allocate some system resource; this is usually a memory problem.

NIF_ERR_SERVER_CONNECTION_LOST

The session established with NI-FBUS for this descriptor has been closed or lost.

NIF_ERR_SERVER_NOT_RESPONDING

Either the NI-FBUS server has not been started, or the server, in its current state, cannot respond to the request.

NIF_ERR_SM_NOT_OPERATIONAL

The device is present, but cannot respond because it is at a default address.

NIF_ERR_SYMBOL_FILE_NOT_FOUND

NI-FBUS could not find the symbol file.

NIF_ERR_TIMEOUT

The device containing the object is present but did not respond within the timeout period.

NIF_ERR_WRITE_IS_PROHIBITED

The device reported that the `WRITE_LOCK` parameter value is set. The `WRITE_LOCK` parameter prohibits writing to the `name` parameter.

NIF_ERR_WRONG_MODE_FOR_REQUEST

The device reported that the current function block mode does not allow you to write to the parameter.

NIFCONF_ERR_ADDRESS_OUT_OF_RANGE

This address does not fit in the allowable range of addresses.

NIFCONF_ERR_ADDRESS_USED

This address is already taken.

NIFCONF_ERR_BAD_WIRE

This wire is not completely attached to an output and input(s).

NIFCONF_ERR_BLOCK_SCHEDULE_CONFLICT

A block in this device is already scheduled for this time. Do not schedule more than one block in the device at the same time.

NIFCONF_ERR_BUF_TOO_SMALL

The buffer is too small.

NIFCONF_ERR_CONFIGURING

You are configuring this project. If you get this error, stop configuring the project to continue.

NIFCONF_ERR_CONFUSED_WIRE

This wire is not working. The NI-FBUS Configurator has removed it.

NIFCONF_ERR_CORRUPT_FILE

The NI-FBUS Configurator found a corrupt file object.

NIFCONF_ERR_COULD_NOT_DETERMINE_SCHEDULE

The NI-FBUS Configurator cannot fit the schedule of the loop into the current schedule. Usually, insufficient time or poor choice of loop times cause this error. Use integer multiples such as 1, 2, 4, and so on, instead of multiples like 5, 7, 13, and so on, to solve this problem.

NIFCONF_ERR_COULD_NOT_START_THREAD

The NI-FBUS Configurator is unable to start a new thread.

NIFCONF_ERR_DD_SIZE_MISMATCH

The actual data size does not match the device description data size.

NIFCONF_ERR_DEVICE_NOT_ALIVE

The device is not responding.

NIFCONF_ERR_DOWNLOAD_INCOMPLETE

The link is only partially configured.

NIFCONF_ERR_DUPLICATE_ADDRESS

The address for this device was found elsewhere in the configuration file. Duplicate addresses are not allowed on a Fieldbus link.

NIFCONF_ERR_DUPLICATE_TAG

The tag for this object was found elsewhere in the configuration file. Duplicate tags are not allowed on a Fieldbus link.

NIFCONF_ERR_INCOMPATIBLE_VERSION

This file was made with a newer version of the NI-FBUS Configurator. There might be unpredictable problems with the project.

NIFCONF_ERR_INVALID_MACROCYCLE

This macrocycle conflicts with a previously defined macrocycle.

NIFCONF_ERR_INVALID_NUMBER

This is an invalid number. A number can use the digits 0–9 and the letters A–F. Hexadecimal numbers can be preceded with `0x`.

NIFCONF_ERR_INVALID_TAG

This tag is not valid.

NIFCONF_ERR_INVALID_VALUE

You have entered an incorrect value. Please check the value(s) and the valid ranges for the values.

NIFCONF_ERR_LOOP_IS_EMPTY

This loop does not contain any blocks.

NIFCONF_ERR_MACROCYCLE_CONFLICT

This macrocycle conflicts with a previously defined macrocycle time for a device in this loop. Devices can only schedule one macrocycle time. The best solution is to increase the macrocycle time of the loop that contains the other block(s) for this device.

NIFCONF_ERR_MACROCYCLE_NOT_GOOD_MULTIPLE

This macrocycle is a bad choice in comparison to the other macrocycles.

NIFCONF_ERR_MACROCYCLE_TOO_SHORT

There was not enough time in this macrocycle to schedule all the function blocks and parameters. Increase the macrocycle time of the loop to get more free time. A macrocyle generally must be at least 10 ms.

NIFCONF_ERR_MAXIMUM_LOOPS_EXCEEDED

The maximum number of loops you can configure on this link is exceeded.

NIFCONF_ERR_MISSING_NIFBUS_PATH

The install path for NI-FBUS Configurator is not in the Windows registry.

NIFCONF_ERR_MULTI_LAS

Multiple primary Link Active Schedulers are defined. Only one primary LAS is allowed on a link.

NIFCONF_ERR_MULTI_TM

Multiple time masters are defined. Only one time master per link is allowed.

NIFCONF_ERR_NO_CONVERSION

There is no conversion for this data type.

NIFCONF_ERR_NO_DATA

No data.

NIFCONF_ERR_NO_LAS_SELECTED

A schedule has been created but no LAS is selected.

NIFCONF_ERR_NO_LM_SELECTED

A schedule has been created but no link master is selected.

NIFCONF_ERR_NO_OBJECT_SELECTED

Nothing is selected.

NIFCONF_ERR_NO_PRINTER_DEVICE_CONTEXT

The NI-FBUS Configurator cannot draw the document because of an incorrect device context for the selected printer.

NIFCONF_ERR_NO_TM_SELECTED

No time master is selected.

NIFCONF_ERR_NO_UNSCHEDULED_TIME_LEFT

The auto scheduler ran out of unscheduled time. Increase the macrocycle time(s) to allow more free time for scheduling.

NIFCONF_ERR_NOT_ON_LINK

This object does not belong on this link.

NIFCONF_ERR_OD_SIZE_MISMATCH

The size of the actual data object does not match the object dictionary size.

NIFCONF_ERR_OUT_OF_MEMORY

Memory is low; the NI-FBUS Configurator cannot allocate objects. Close windows to solve this problem.

NIFCONF_ERR_PARAMETER_SCHEDULE_CONFLICT

A parameter communication has already been scheduled for this time. Do not schedule more than one parameter communication at a time on the Fieldbus.

NIFCONF_ERR_REGISTRY_ERROR

Some registry information for your application is missing or corrupt. You might need to reinstall the NI-FBUS software and the NI-FBUS Configurator. Refer to your getting started manual for installation instructions.

NIFCONF_ERR_STANDARD_DD_MISSING

The NI-FBUS Configurator could not find the standard device description file.

NIFCONF_ERR_STRING_NOT_FOUND

There is a missing resource string.

NIFCONF_ERR_TEMPLATE_NOT_ASSIGNED

The template is not assigned to a valid project object.

NIFCONF_ERR_TOO_MANY_FB_SCHEDULE_ENTRIES

The schedule for this object contains too many entries for the function block schedule of the device. Normally, bad choices for macrocycle times cause this error. Use macrocycle times that are integer multiples of each other to correct this problem.

NIFCONF_ERR_TOO_MANY_LAS_SCHEDULE_ENTRIES

The number of LAS schedule entries exceeds the number of available entries in the LAS device. Choose integer multiple macrocycle values and reduce the number of data links that go between devices to correct this problem.

NIFCONF_ERR_TYPE_INFO_UNAVAILABLE

The NI-FBUS Configurator cannot read the type information for the object.

NIFCONF_ERR_UNSCHEDULED_TRAFFIC_CONFLICT

The publish time is conflicting with the reserved 128 octet time at the end of a macrocycle. This time is used for unscheduled traffic.

NIFCONF_ERR_WRONG_TYPE

This object has the wrong type.

Warnings

If you receive a warning while using the NI-FBUS Configurator, refer to this section for a description of the warning and possible solutions.

NIFCONF_WARN_MEMORY_LOW

The system is low on memory.

NIFCONF_WARN_OBJECT_NOT_WIRED

This object is not connected by wires to any other object.

NIFCONF_WARN_NO_TREND_SELECTED

A trend is wired, but no trend is checked.

NIFCONF_WARN_TREND_NOT_WIRED

A trend is selected, but not wired to a device.

NIFCONF_WARN_PROJECT_ERRORS

There are errors that might cause problems in the download of the configuration. Look at the **Errors** tab in the status window.

B

Troubleshooting and Common Questions

Mode Problems

A function block alternates between Iman and Auto modes. The status of the input parameter is Bad::No comm.

The function block and communication schedules do not have enough time between them. Follow these steps:

1. Edit the schedule to add more time between the function block and communication schedules. To edit the schedule, double-click on the **Schedule** icon in the configuration tree.

   For more information about editing the schedule, refer to the *View and Edit a Schedule* section in Chapter 2, *Using the NI-FBUS Configurator*.

2. After you edit the schedule, select **Configure»Download Configuration** to download the changes.

   For more information about downloading the configuration, refer to the *Download a Project Configuration* section in Chapter 2, *Using the NI-FBUS Configurator*.

The PID does not transition to Auto mode.

If the status of an OUT or BKCAL_OUT parameter on an AI or AO block is Bad::Not Connected or Bad::No Comm, or the downstream block is not in cascade mode, follow these steps:

1. Use the Monitor mode to make sure that the AI and AO blocks are publishing correct values.
2. Make sure that the LAS executes the communication schedule. Use monitoring mode to view the modes, inputs, and outputs of the block. If the LAS is not executing the schedule, select **Configure»Download Configuration** to download the configuration.

For more information about downloading the configuration, refer to the *Download a Project Configuration* section in Chapter 2, *Using the NI-FBUS Configurator*.

3. Make sure the target mode of both the PID and AO blocks is set to `Cas`.

The AI block does not transition to `Auto` mode from `OOS` mode.

The configuration is incorrect or incomplete. Complete the following steps:

1. Make sure that the resource block is in `Auto` mode.
2. Make sure the `XD_SCALE` fields are set to the value specified by the vendor.
3. Set the target mode of the AI block to `Auto`.

The function blocks are in `OOS` mode.

The resource block is in `OOS` mode. Set the target mode of the resource block to `Auto`, then set function block to their normal mode.

AI block will not go to `Auto` Mode.

The block must be scheduled to execute (done by placing it in the Function Block Application Editor window and downloading) before it can go into `Auto` mode. If the block still will not go into `Auto`, try these solutions:

- Check that the resource block is in `Auto`.
- Look at the `BlockError` parameter on the **Diagnostics** tab. This will often give reasons for staying in `OOS` mode.
- For FieldPoint, you can look at the `LAST_BLOCK_EVENT` parameter.
- Make sure that the parameter `L_TYPE` has been initialized and other parameters are set properly.

When using an FP-3000 with a PID block cascade, the high limit flag goes ON as it should when the output reaches the high limit, but it stays ON even when that output goes back down.

Reduce the `BKCAL_HYS` parameter. When outputs hover around alarm limits, this triggers the alarm to turn on and off. After an alarm has been triggered, this parameter specifies how far below the alarm limit the output must fall before the alarm condition will clear. Setting this to zero means that as soon as the output falls below the alarm limit, the condition will clear.

The Fieldbus device stays in OOS mode even though the resource block's MODE_BCK.TARGET = Auto and MODE_BCK.ACTUAL = Auto

The Fieldbus device needs to be configured. Start by dragging the function block in question, such as an AI, AO, or PID block, into the Function Block Application Editor window. The block does not need to be connected with anything else. Then, download the configuration—this will allow it to execute on the device. If there are problems, try these steps:

1. Make sure that the resource block is in `Auto` mode—not `OOS`. This is found by either checking which button is depressed in the resource block's block configuration window or by looking at the `MODE_BCK.ACTUAL`.
2. See if the `Options` parameter is in `Run`. If so, check the `Block_ERR` in the resource block—it should be zero.
3. Go to the I/O function block in question, and make sure that the `OUT/IN.status` is in `Good`… mode. If not, go to the `Block_ERR` parameter and see what the problem is.

The resource block must be in `Auto` mode for any other blocks in the device to execute. The resource block is a good place to start troubleshooting if the device is not behaving as desired. It has diagnostic parameters that help you determine the cause of the problem.

A resource block specifies the general characteristics of the resource. This includes the device type and revision, manufacturer ID, serial number, and resource state. Each device has only one resource block. The resource block also contains the state of all of the other blocks in the device.

If the device cannot change its `ACTUAL` mode to the `TARGET` mode, first verify that you have downloaded the configuration to the device. If the configuration has been downloaded, you can diagnose the problem by looking at the `BLOCK_ERR` parameter of the function block. This parameter often identifies the problem. For example, if the device's resource block is off-line, the `BLOCK_ERR` parameter will read `Resource Block Offline`, indicating that you must change the resource block to `Auto` mode before you can change the function block's mode. Also, if the device is an FP-3000, you can look at the additional parameters on the **Diagnostics** tab in the block configuration window. Note that the `LAST_BLOCK_EVENT` is not cleared after an error condition clears. Check the time stamp to determine whether the event you are looking at is current.

Why are discrete states integers instead of booleans?

Because it allows for multiple state discrete functionality such as off, turning on, on, turning off. Not all devices take full advantage of this. For example, on the FP-3000, 0 is off, and 1–16 are on. Because these parameter values are integers, they show up in your HMI as analog.

Bad Status

You receive a status of `Bad:Device Failure`.

An error exists with the hardware. Set the `RESTART` parameter in the block configuration window to `Processor` or `Default`. If the problem persists, replace the device.

PID block shows bad quality.

- The function block is not actually executing on the device. Make sure you placed it in the Function Block Application Editor window and downloaded the project.
- The PID is not subscribing correctly. Go to the schedule and pull the blocks apart a bit to allow more time. Make sure that there is sufficient unscheduled time left in the schedule. Download the project after changing the scheduler or Function Block Application Editor.
- If this still does not solve the problem and your configuration file contains the entire application, download with the **Clear Devices** box checked.

I am using a PID loop to control a 4-20 mA signal. The range for my analog input block is 4-20 mA. The quality of the reading of my PID block is Uncertain when it reaches the setpoint (20 mA). Why?

The value the analog input is reading when the PID reaches the setpoint is 20 mA. The device reports the quality as Uncertain because the device is able to read inputs only up to 20 mA. If you actually have an input of 21 mA, the device will read this as 20 mA, since this is the maximum range setting. If the input is 20 mA, the device will read this also as 20 mA. For this reason, it is impossible to know whether a reading of 20 mA is actually a good reading or not. The device reports Uncertain status so that the user is aware that signal value might not really be 20 mA. To make sure that you get a quality of Good, you simply increase the range to something slightly larger than your maximum expected value. You can also force the NI-FBUS Configurator to treat Uncertain as Good. This is done on a function block-by-function block basis. Double-click on the block for which you would like to change this setting. Under the Options tab, locate the parameter called `STATUS_OPTS`. If you click on its field, a drop-down menu will appear with a check box called Uncertain as Good. If you select this option and click on **Write Changes**, the function block will now treat Uncertain quality received on its input as Good.

Missing Device

Selecting **Update Attributes** from a device's menu on the menu bar, or when right-clicking on the device, causes the device description to be reread. This is sometimes useful after replacing devices.

Information from the device description does not show up under Windows NT.

If you are not logged in as administrator under Windows NT, the block types in the configuration tree are listed as being unknown. The block configuration window correctly lists parameter names but not the names of enumeration values. You need to log in as an administrator to view the device description information.

A device does not appear in the configuration tree.

The communication parameters are incorrect. Complete the following steps:

1. Check all connections.
2. Make sure that the bus is appropriately powered, if your devices require it or that their own power supplies are providing power.
3. Select **Start»Programs»National Instruments FBUS»Interface Config** and click on the **Advanced** button.
4. Set the following parameter values.

Parameter	Value
Number of unpolled nodes	See discussion on setting the number of polled addresses
MinInterPduDelay	20
Slot time	10
Max Response delay	8

5. Shut down the NI-FBUS Configurator and NIFB process.
6. Select **Start»Programs»National Instruments FBUS»nifb** to restart the `nifb` process.
7. Select **Start»Programs»National Instruments FBUS»NI-FBUS Configurator** to restart the NI-FBUS Configurator.

If your device still does not show up on the bus, check your device's documentation for any requirements on network parameter settings.

The device does not show up in the NI-FBUS Configurator

1. Try setting the board to a visitor address. Select **Start»Programs» National Instruments FBUS»Interface Config**. Click on the appropriate port. Click on the **Edit** button. Change the **Address** radio button to **Visitor**. If there is another link master device on the bus, set the **Device type** of the interface board to **Basic device**. Restart NIFB. Now see if the device shows up on the network.
2. If the undetected device is a link master, try setting the NI-FBUS interface as a basic device rather than a link master. To do this, go into the NI-FBUS Interface Configuration utility. Click on the appropriate port. Click on the **Edit** button. Change the **Device type** to **Basic device**. Restart NIFB. Now see if the device shows up on the network.
3. Start NIFB. In the NI-FBUS Dialog utility, right-click on the open descriptors. Choose **Expand All**. Then, right-click on the session and open devices, etc. See what kind of devices show up. If an unknown device shows up, the device is probably causing interference. The software can tell that something is there, but since it cannot communicate with it, it shows up as an unknown device. If this is the case, then it is probably a problem on the device side. The device is not keeping within its time boundaries. The device might be trying to transmit before its allotted time or continuing to talk past the allotted time. Try increasing the minimum inter-PDU delay. This is set in the NI-FBUS Interface Configuration utility. If you click on the Port, then **Edit**, then `Advanced`, you will see this category under **Dlme Basic Info**. The default is 10. By increasing this time, you can increase the minimum amount of time a device must wait before sending another packet. Restart NIFB. Now see if the device shows up on the network.
4. You can try changing bus timing parameters `Slot Time`, `Max Response Delay`, and `Min Inter-PDU Delay`. If you want to try this, contact the device manufacturer for recommended settings. Do not guess at these parameters.

I have assigned an address to my device, but the NI-FBUS Configurator is not seeing the device. Why?

In the NI-FBUS Interface Configuration utility, there is an advanced parameter called `First Unpolled Node`. After the node is specified by this parameter, the NI-FBUS Configurator will not poll the subsequent *N* nodes, where *N* is the `NumUnpolledNodes`. Make sure that your device's address is being polled. The default is 0x25.

The second possibility is that the timing parameters (T1, T2, T3) are improperly set. The default for these are 0x40000, 0x40000, 0x30000, respectively. Other settings to try are 0x60000, 0x60000, and 0x35000. If these parameters are set inappropriately, an identify message can go out before the device is at its address, resulting in the device missing its chance to identify itself. This problem causes the Set Address sequence to fail. This problem has in particular been seen with the Fisher Rosemount 3244MS rev3.

How can I get a slow device onto the bus?

There are timing parameters (MIB parameters) that can be set for the link, including the minimum interPDU delay, the maximum response time, and the slot time. This allows you to accommodate devices with slower response times.

I am getting an error `DD_SIZE_MISMATCH_ERROR`

The device description file installed in your machine does not match the device.

1. Open the resource block for the mismatched device.
2. Check the `MANUFAC_ID`, `DEV_TYPE`, `DEV_REV`, and `DD_REV`.
3. Open the NI-FBUS Interface Configuration utility and click on the **DD Info** button.
4. Find the `.ffo` file in the base directory for the device description.
5. Compare the `.ffo` revision number with the manufacturer ID, device type, device revision, and device description revision from the resource block. If the revisions are different, import the device description file for the correct revision of the device. If they are the same revision, contact the device vendor to get correct device description files.

Read/Write Errors

The **Write Contained Block Parameters** option when downloading causes all contained parameters to be written—not just those that were changed. This can cause error messages for parameters that you did not even modify because they are initially set to illegal values as defined in the FOUNDATION Fieldbus specification. These values are only quasi-illegal. Devices are often designed to be able to operate with the default illegal values to allow you to quickly bring up a test system. To encourage you to set them to values appropriate for your process, trying to write the default parameter value causes errors. Examples of such parameters are `ALERT_KEY`, `L_TYPE`, and `SHED_OPT`.

You are unable to write to a block parameter.

The parameter is read-only, the blocks are in an incorrect mode, or the data is out of range. If the parameter is not read-only, complete the following steps to solve the problem:

1. Change the target mode to `Man` or `OOS` and then try to write to the parameter.
2. The data might be out of range. Check the range and make sure the value is valid.

A device does not accept a new tag.

The device is in an invalid state. Power the device off and on.

You receive frequent communication errors when you try to read or write to a device.

- Other devices are actively and frequently polling the device, causing reads to time out. Reduce polling. Make sure you disable periodic updates. If you have scheduled other devices to poll this device frequently, try changing the poll rate.
- The device is in an invalid state. Use the `RESET` parameter to cycle power and reset the device.

A device does not transmit alarms or trends.

Either you have not configured them yet, or there is not enough time on the schedule. For instructions on how to configure alarms and trends, refer to the *Configuring Alarms* and *Configuring Trends* sections in Chapter 2, *Using the NI-FBUS Configurator*, respectively.

If a device does not publish the data you configured it to transmit, complete the following steps:

1. Edit the schedule to add more time to the communication schedule. To edit the schedule, double-click on the **Schedule** icon in the configuration tree.

   For more information about editing the schedule, refer to the *View and Edit a Schedule* section in Chapter 2, *Using the NI-FBUS Configurator*.
2. If your project reads and polls information continuously, turn off polling to see if the device publishes the data. To turn off polling, uncheck the **Periodic Updates** checkbox at the top of the block configuration window.

Set Address Problems

A device does not accept a permanent address.

The device does not have a tag. For instructions on setting the tag, refer to the *Set Device or Block Tags* section in Chapter 2, *Using the NI-FBUS Configurator*. If the device already has a tag, the T3 network parameter might not be the right length. Remove the device from the active bus and configure its address separately.

Why are my MIB parameters all showing up as zero?

If you have no address assigned, the NI-FBUS Configurator is not able to read the parameters.

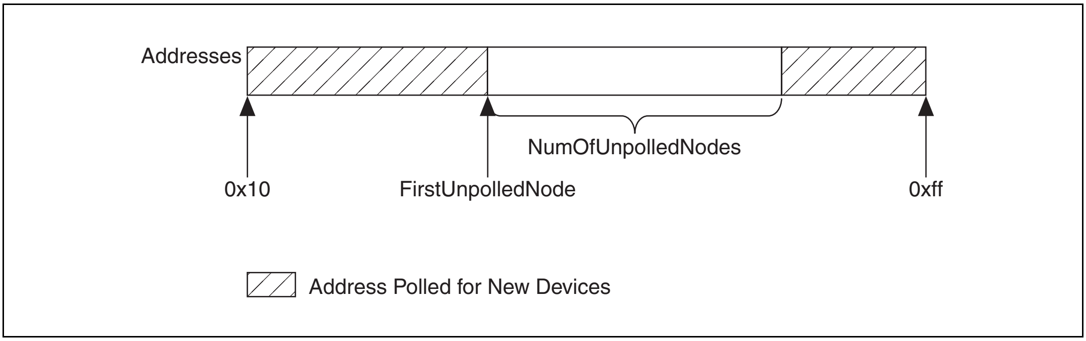
I get a Read vfdlist fail error.

There are probably two devices at the same address. Change one to a different address.

Setting Number of Polled Addresses

The Fieldbus specification describes how a Link Active Scheduler (LAS) device probes a list of addresses to allow devices to come online during normal operation. The LAS sends a Probe Node (PN on the bus monitor) packet to each address in its list of addresses every so often during operation, where “every so often” is a time that depends on the number of devices on the link, and the setting of the Link Maintenance Token Hold Time parameter.

The Fieldbus specification describes how to tell the LAS to “skip” probing certain addresses in the range, to speed up how long it takes to detect new devices on the bus (or devices that are having their addresses changed). The two parameters involved in maintaining the list are called `FirstUnpolledNode` and `NumOfUnpolledNodes`, and they can be found in the NI-FBUS Interface Configuration utility advanced settings for a port, in the **DLME Master Info** section. The following diagram shows how the LAS determines the list.



In other words, `FirstUnpolledNode` tells the LAS the beginning of a region of addresses not to probe, and `NumOfUnpolledNodes` tells the LAS the length of that region. So if `FirstUnpolledNode` was 0x25, and `NumOfUnpolledNodes` was 0xba, then no addresses from 0x25 to 0xdf would be probed. That means that if a device with an address of 0x25 were placed on this bus, the LAS would not probe it, and it would never be able to send or receive packets on the bus.

Note The reason to have a `NumOfUnpolledNodes` whose value is nonzero is as follows. The LAS probes every address in the list, then starts over again at the beginning. Because a device cannot come on the bus until its address is probed, if the LAS is probing all 255–16+1=240 possible addresses, and each probe node request goes out every *T* milliseconds, it might take 240*T* milliseconds for a device to get on the bus. If, however, the LAS probed only the first 16 addresses and the last 16 addresses, it might take 32*T* milliseconds for the device to get on the bus; this results in the new device being recognized almost eight times faster.

These parameters also affect the Set Address protocol, because recognizing a device at a new address is really the same as recognizing a completely new device, as the new address must be probed for the device to come online. In this way, the `NumOfUnpolledNodes` parameter can affect the value of the Set Address protocol parameter `T3`. For example, increasing the `NumOfUnpolledNodes` parameter might fix a `SetAddress T3` problem because it takes the device less time to be recognized at the new address.

For testing purposes, setting the `NumOfUnpolledNodes` to zero ensures that all addresses will be checked. Once the device is found, you will probably want to advise the `NumOfUnpolledNodes` and/or `FirstUnpolledNode` to avoid the performance slowdown discussed above.

Performance Problems

The control is slow.

The macrocycle time might be too long. Edit the function block schedule and decrease the macrocycle time. For more information about editing the schedule, refer to the *View and Edit a Schedule* section in Chapter 2, *Using the NI-FBUS Configurator*.

Missing Link

Your interface(s) does not appear in the Add Links dialog box.

You may need to upgrade your software key to support the additional links. If you purchased additional licenses to access more than the default number of links, contact National Instruments to obtain a new software key.

A device believed to be a link master does not show up in the network parameters window.

For a link master device to receive the schedule, it must be configured to receive it. This setting is made by double-clicking on **Network Parameters** in the configuration tree and ensuring that the checkbox for the appropriate link master devices are checked. If a device does not appear in this list, follow these steps:

1. Right-click on the interface in the configuration tree and select **Advanced View**. The MIB parameters appear underneath the interface board.
2. Double-click on the MIB to bring up another window. Look at the parameter called `BOOT_OPERAT_FUNCTIONAL_CLASS`. The value `2` indicates that the device identifies itself as a link master. If this parameter does not equal `2`, the device is not identifying itself as a link master. Contact your device manufacturer.

Fisher Rosemount 3244MS rev3 (Link Master device) does not show up on the bus.

The Fisher Rosemount 3244MS rev3 temperature transmitters are intrinsically safe link master devices. With the default timing settings, they do not appear on the bus. These devices require the tweaking of some bus parameters for them to show up in the NI-FBUS Configurator. Revision 2 of the 3244MV works fine without these adjustments. These changes are made by right-clicking on the link (the top level in the hierarchy) and selecting Advanced View. Now the MIB parameters appear underneath the representation of the interface board. Double-clicking on the MIB will bring up a block configuration window where these values can be changed.

- Make the interface's `PRIMARY_LINK_MASTER_FLAG` parameter `FALSE`.
- Make the interface's `CONFIGURED_LINK_SETTINGS` match those of the Rosemount device. The values for the `CONFIGURED_LINK_SETTINGS` on the Rosemount transmitter are (in order): 10, 6, 5, 37, 0, 12, 186, 2, 1, 0, 4

At the time of publication, Fisher Rosemount was advising customers to make the following changes to the advanced settings.

Caution Do *not* modify these parameters without good reason. If you must modify parameters for certain devices, the device manufacturer will recommend settings. Modifying these parameters can have an adverse affect on data throughput rates. If settings are incorrectly modified, some devices may disappear off the bus.

Table B-1. Rosemount Advanced Parameter Settings

Parameter	Value
`T1`	0x17700
`T2`	0x1D4C00
`T3`	0x249F0
`Slot Time`	8 or higher
`Maximum Response Delay`	3 or higher
`Minimum inter-PDU Delay`	12 or higher
`CONFIGURED_LINK_SETTINGS`	Make the same for all link masters on the link

Missing Information

A function block is missing symbolic information (such as names and help strings).

The function block has no device description, and the block does not correspond to the standard function blocks defined by FOUNDATION Fieldbus. You may need to import the device descriptions. To do so, refer to the *Missing Symbolic Information* section.

Another possibility is that the NI-FBUS Configurator cannot read pertinent block information from the device. Reset the device.

Missing Symbolic Information

If a function block is missing symbolic information (such as names and help strings), you may need to import the Device Descriptions (DDs). For more information about device description files, refer to the *Device Description Files* section in the *FOUNDATION Fieldbus Overview* document.

Add/Delete Tabs

The NI-FBUS Configurator crashes when you try to add or delete tabs in the block configuration window for a block which has menus and/or methods.

The best solution would be to customize the existing tabs. Follow these steps to add your own tabs to a block with menus and methods:

1. Select **Start»Programs»National Instruments FBUS»Interface Config** to run the NI-FBUS Interface Configuration utility.
2. Click on the **DD Info** button. The **DD Info** dialog box appears.
3. Write down your existing device description base path.
4. Clear out the device description base path and close the program.
5. Restart NI-FBUS and launch the NI-FBUS Configurator.
6. Bring up the block in question (the NI-FBUS Configurator will show some block types as “UNK”—this is expected).
7. Add the new tab (or tabs).
8. Close the block configuration window (say **Yes** to **Save custom changes**).
9. Close the NI-FBUS Configurator (do not save the project).
10. Launch NI-FBUS Interface Configuration utility again.

11. Restore the device description base path that you wrote down in Step 3.
12. Restart NIFB and launch the NI-FBUS Configurator. The new tabs should show up in the block configuration window.

Missing Function Blocks

Only some Fieldbus devices and configuration software support the instantiation (creation) and deletion of function blocks. Most devices are shipped with hard-coded function blocks. The NI-FBUS Configurator detects the difference automatically.

The function blocks in the NI-FBUS Configurator show up as unknown (UNK).

Information is stored in Device Description (DD) files that describe the function blocks available on the device. This data is stored in two files: the `.ffo` (definitions) and the `.sym` (names). You import device descriptions in the NI-FBUS Interface Configuration utility. There is an **Import DD** button that will copy the `.ffo` and `.sym` files into the correct directory. Files install under `C:\NIFBUS\Data` by default. There is a subdirectory for each device manufacturer. Under the manufacturer, there are additional directories for each device. If these files are not imported properly, any function blocks specific to the device will show up as UNKnown. Just copying the `.ffo` and `.sym` files into the `nifbus\data` directory will NOT work. Refer to the *Start the NI-FBUS Configurator and NIFB Process* section in Chapter 2, *Using the NI-FBUS Configurator*. Note that you do not need a device description for the National Instruments interface boards, but you do if you have an FP-3000 device.

After updating the firmware, the NI-FBUS Configurator no longer recognizes these as the same function blocks.

If there is a firmware update which results in a change in item IDs, the NI-FBUS Configurator cannot recognize the new blocks as being compatible with the old blocks. Below is a brute-force way to correct the problem.

1. Make a backup copy of your project.
2. Create a template, as described in the *Using Templates* section in Chapter 2, *Using the NI-FBUS Configurator*. Select the blocks that need to be changed to the new item IDs, and create a template from them.

3. Drop the template you just created into the project. When prompted, check the **CHANNEL values** checkbox and choose **Yes, accept values**. This will result in the template parameters (which you had configured in your original projects) being applied to the blocks which you will later open.
4. Replace the template blocks with the new blocks from the updated device.
5. Delete the original blocks used to make the template and rewire, if necessary.

If you save the project as a template, you will be able to keep the old parameter values.

Viewing Problems

You can choose whether you see parameter values in decimal or hexadecimal. Right-click in the block configuration window and check or uncheck the **Hexadecimal Data** option.

All the types in the block configuration window all look alike.

The drop-down box for editing enumerated types on blocks is sized according to the width of the **Value** column. You may need to resize the **Value** column manually to view the entire string.

Modify window display.

If you do not like all the clutter in your status window, you can right-click and choose **Show Errors Only**

Under the **View»Preferences**, you can change several settings for the block configuration window, including whether or not to **Show Block Information**. Selecting this option will add another tab in the block configuration window. This tab gives block-specific information, such as its worst-case execution time. This tab can also be brought up by selecting the **Add or Delete Custom Tab** button in the block configuration window.

Windows NT-Specific Problems

What do I do if the NI-FBUS Configurator does not work like it should under Windows NT?

Make sure you are logged in with an administrator account.

Using Fieldbus with Lookout

How do I use Fieldbus with Lookout?

1. In Lookout, create an OPCClient object. Edit connections for this OPCclient object and set the Activate member to FALSE.
2. Add all the items you are interested in.
3. Activate the OPCclient object by going through Edit Connections dialog box and setting the Activate member to TRUE. This will configure the MVC items.
4. To open a previously saved lkp/process file, follow a similar deactivation/activation procedure.

What do I do if Fieldbus objects do not appear in Lookout?

If you want to use the native Fieldbus objects in Lookout, you have to delete the lookout.dat file in the Lookout directory. This file is an index file that tells Lookout what objects it has available. Fieldbus objects are not available by default. Lookout will regenerate the lookout.dat file the next time it is started. When it regenerates the file, it will see that Fieldbus software has been installed and will make the Fieldbus objects available.

How do I use Fieldbus Alarms in Lookout?

In Lookout, there is a separate alarms window for Fieldbus alarms. Under the **Options** menu, choose **Fieldbus** to show this window. The window can also be shown using traditional Lookout datamember ShowAlarms. See the entry for National Instruments Fieldbus in the Lookout object reference manual (also available from the Help menu within Lookout).

Note that if you want alarms to appear in the standard Lookout alarm window (rather than the Fieldbus alarms window), you must create Lookout alarm objects.

I am trying to control a PID loop within the FP-3000. On occasion, the Lookout operator may want to override the output of the PID with a manual output setpoint. How can I implement this?

1. Double-click on your PID block in the NI-FBUS Configurator.
2. Under the **Options** tab, you will see a parameter called CONTROL_OPTS. There are several options that you can enable or disable. When you click to the right of the parameter's name, a drop box appears. Check the box next to TRACK ENABLE. This gives you the

option of using a direct value rather than the PID-calculated value. Which value you use depends on another parameter, TRK_IN_D.

3. Now go to the **Scaling** tab. You need to set the quality of the TRK_IN_D and TRK_VAL to Good_NonCascade. TRK_IN_D turns tracking on and off. For normal PID operation, you want this value to be zero. To send the TRK_VAL directly to the PID output, you want this to be 1. Connect the switch in Lookout to the TRK_IN_D.VALUE of the PID object. The Pot should be connected to TRK_VAL.VALUE.

   One other thing to be aware of is that the PID algorithm in the firmware revision 1 of the FP-3000 device has a problem. Its PV_SCALE must be set to 0-100%. Other scaling options are ignored. Until this is corrected in the firmware, a work-around is to simply use the percent of scale value and set the PV_SCALE of your AO block to be 0–100% as well.

   You may also need to set TRK_SCALE to 0–100%. You should probably use whatever values come directly from the Pot object in Lookout as long as you set TRK_SCALE appropriately. If you have troubles, try doing the scaling of the Pot (making it 0–100%) in Lookout.

Open/Find Project

If you want to load a saved process file, choose **Cancel** in the **Add Links** dialog box, which appears at startup. Then, choose **File»Open**. Otherwise, the NI-FBUS Configurator will open a new file automatically, and you will have too many windows to keep track of.

One thing that is often confusing is knowing which configuration tree you are in. If you have multiple configuration files open and then open the Function Block Application Editor window, you can often lose track of which configuration file they are manipulating. Notice that the name of the active file appears in the title bar. Choosing **File»Close** will close that file. The additional windows (Schedule, Function Block Application Editor window, etc.) to the right correspond to the active configuration file.

Using NIFB

If you need to shut down NIFB and restart, you do not have to shut down the NI-FBUS Configurator. After restarting NIFB, you can reconnect by right-clicking on the link (the top level in the hierarchy) and selecting **Online**.

Restart Devices

You can do a software restart of a device's on-board processor by double-clicking on the device's resource block. Under the **Options** tab, you should find a `RESTART` parameter. Changing this to `Processor` and writing changes will restart the processor.

Using Fieldbus with Server Explorer

1. Launch the Server Explorer.
2. Create an inactive OPC client group:
   a. Right-click on OPCNifbus and select **Add/Edit Groups**.
   b. Create a group with the appropriate parameters. Make sure you uncheck the **Active** box.
3. Add all items.
4. Select **File»OPC»Save** to save the file.
5. Activate the group by right-clicking on the group and selecting **Activate Group**.
6. When you open the file you saved and want to go live, right-click on **OPCNifbus** and select **Connect to Server**. After Server Explorer has connected to the server, activate the group as described in Step 5.

Uninstalling the Software

To uninstall your NI-FBUS software, select **Start»Settings»Control Panel»Add/Remove Programs.** Select **NI-FBUS Configurator** to uninstall.

The uninstall utility does not remove the NI-FBUS directory itself or any files in the `DATA/NVM` directory. To completely uninstall the software, manually remove the files in the `DATA/NVM` directory and the NI-FBUS directory structure.

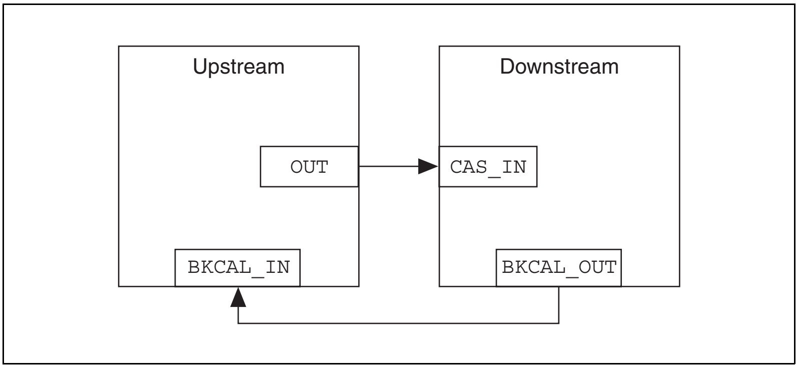
Cascade and Remote Cascade Setups

Cascade Initalization

FOUNDATION Fieldbus provides a protocol called Cascade Initialization that allows a control function block to smoothly transition from `Man` to `Auto` mode. Cascade Initialization allows the PID algorithm to know the current setpoint of the AO block to balance the actual setpoint with the control's setpoint over time. Cascade Initialization is also used to prevent windup in the PID.

Parameter Connections for Cascade Initialization

Cascade initialization takes place between two blocks: an upstream controlling block, and a downstream controlled block. In a PID loop, the upstream block is the PID block, and the downstream block is the AO block. In the case of cascaded PID blocks, the upstream PID feeds a setpoint into a second PID that is acting as the downstream block. In both cases, the parameter connections are the same. The output (`OUT`) parameter of the upstream block is connected to the cascade input (`CAS_IN`) parameter of the downstream block. This connection controls the setpoint of the downstream block. To allow the upstream block to determine the current setpoint of the downstream block, you must also connect the backward calculation output (`BKCAL_OUT`) parameter of the downstream block with the backward calculation input (`BKCAL_IN`) of the upstream block. The connections are shown below.



Mode and Status Behavior During Cascade Initialization

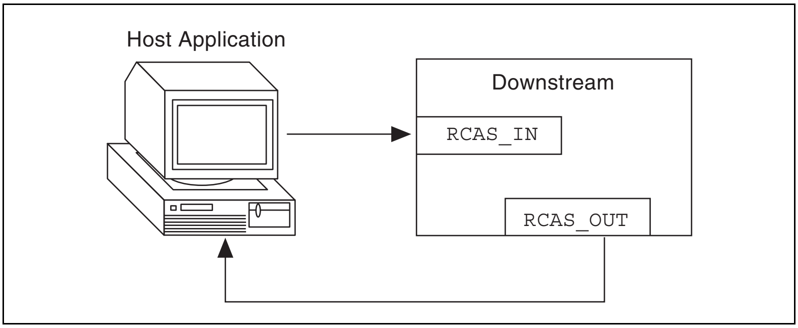
Cascade initialization is arbitrated through the status of the backward calculation path from the downstream block and the forward calculation path on the upstream block. If the upstream block publishes a status of `Good, Non-Cascade`, it does not support cascade initialization, and the lower block immediately transitions into a Cascade mode. This happens in the case where an Analog Input (AI) block is acting as the upstream block for an Analog Output (AO) block. Since the AI block does not have a back calculation input, it does not support cascade initialization.

If the upstream block does support cascade initialization, it publishes a status of `Good, Cascade` on its forward calculation output. This signals to the downstream block to begin the cascade initialization process as soon as it is able. If the downstream block is unable to begin cascaded control, it publishes a status of `Good Cascade, Not Invited` on its backward calculation output. This signals to the upstream block that the control path from the downstream block to the process has been broken. As soon as the ability to begin control is established, the downstream block publishes a status of `Good Cascade, Initialization Request` on its backward calculation output. This signals to the upstream block that it should initialize itself for cascade control. While the initialization request status is active, the downstream block is also publishing its current setpoint to the upstream block. This enables the upstream block to prepare for a smooth transfer to automatic control. While the upstream block is initializing itself for automatic control, it enters an actual mode of Initialization Manual (`IMan`). When it is ready to begin control, it publishes a status of `Good Cascade, Initialization Acknowledge` to signal that it is beginning cascade control. The lower block then enters Cascade mode.

To prevent windup, the control loop needs to be aware when it is unable to control the process. If the downstream block can no longer control the process, it reports a status of Bad to the upstream block. This breaks the cascade until automatic control can be resumed, in which case cascade initialization takes place again.

Remote Cascades

If a host application (rather than another block) provides the setpoint of a block, FOUNDATION Fieldbus provides the Remote Cascade mode. The remote cascade mode is equivalent to Cascade mode, except that the cascade input parameter is RCAS_IN instead of CAS_IN, and the back calculation output is RCAS_OUT instead of BKCAL_OUT. Unlike CAS_IN and BKCAL_OUT, which are input/output parameters, RCAS_IN and RCAS_OUT are contained parameters and can only be written by a host application. To allow the controlled block to enter Remote Cascade mode, the host application must act as the upstream block in the cascade initialization and implement the status handling described above.



There is a second remote mode in which a cascade must be initialized: Remote Output (ROut). Unlike RCas mode, where the block setpoint is set remotely, in ROut mode the block output is set by the host application. The back calculation output for ROut mode is ROUT_OUT, and the cascade input is ROUT_IN. Otherwise, cascade initialization proceeds normally.

Bypassing Cascade Initialization

If cascade initialization is unnecessary, you can bypass it by leaving the back calculation input of the upstream block unwired. By manually writing a status of `Good, Non-cascade` to the back calculation input, the block bypasses cascade initialization and immediately transitions to automatic control. The upstream block signals to the downstream block that cascade initialization has been bypassed by setting its output to a status of `Good, Non-cascade`.

D

NI-FBUS Dialog Utility

This chapter describes the NI-FBUS Dialog utility and gives examples of how to use it.

Note You generally will not want to use the dialog utility, since the NI-FBUS Configurator has the same functionality and is easier to use.

NI-FBUS Dialog Utility Overview

The NI-FBUS Dialog utility lets you interact with your devices over the Fieldbus by opening descriptors, making single NI-FBUS calls, and viewing the results. You might want to use the NI-FBUS Dialog utility to verify installation and device operation, or to learn the NI-FBUS Communications Manager API. You can run the NI-FBUS Dialog utility by selecting **Start»Programs»National Instruments FBUS»NI-FBUS Dialog**.

When you open the NI-FBUS Dialog utility, a window appears containing a single item called **Open Descriptors**. This is the root of a tree that shows an icon for each of the NI-FBUS descriptors you open using the NI-FBUS Dialog utility. The area below the icon remains empty until you make an NI-FBUS call to open a descriptor. When you open a descriptor, the NI-FBUS Dialog utility adds an icon representing that descriptor.

You can use the NI-FBUS Dialog utility to perform operations on the descriptors you have opened. Select the operation you want to perform on a descriptor by right-clicking on the descriptor icon and choosing an item on the menu that appears, or by selecting the icon with a single click and choosing an item on the **Actions** menu. The choices that appear on the menu depend on the type of descriptor you have selected.

NI-FBUS Dialog Examples

These examples describe the typical steps you go through when using the NI-FBUS Dialog utility. Before you begin the examples, open the NI-FBUS Dialog utility by selecting **Start»Programs»National Instruments FBUS»NIFBus Dialog**.

Example 1. Get a Device List

Follow these steps to practice using the NI-FBUS Dialog utility to get a device list.

1. Open the NI-FBUS Dialog utility.
2. Right-click on the **Open Descriptors** icon and select **Open Session**.
3. In the **Open Session** dialog box that appears, click on the **Open Session** button. The NI-FBUS Dialog utility makes an `nifOpenSession` call to the NIFB process. This call opens a session descriptor, which represents your connection to the NIFB process.

   If the call succeeds, the NIFB process is running and responding to requests, and a new session descriptor is created under the **Open Descriptors** icon. If the call fails, make sure that your NI-FBUS Communications Manager process is running, and that it has not displayed any error message boxes during startup. You can check this by and looking at the `nifb.exe` console window. To open the window, select **Start»Programs»National Instruments FBUS»NIFB**.
4. Right-click on the session descriptor icon to see its pop-up menu.

   The list that appears represents the NI-FBUS Communications Manager API calls you can make with a session descriptor.
5. Choose the **GetInterfaceList** function from the list of choices. This choice displays the logical name of all known interfaces.
6. Highlight the interface name of your choice and click on the **OpenLink** button. In the dialog box that appears, click on **Open Link**, then click on **Cancel** to close any open dialog boxes.
7. Right-click on the **Link** icon and choose **GetDeviceList**. The NI-FBUS Dialog utility displays a list of active devices on your Fieldbus link. Your Fieldbus interface board is also included in this list.

Example 2. Download a Schedule to an Interface

1. Complete all the steps of *Example 1. Get a Device List*.
2. Select an interface board by clicking on an entry in the device list that has an asterisk (*) on its left.
3. Click on the **Open Device** button. A new dialog box appears with the identifying information for the interface board already filled in.
4. Click on the **Open Device** button on the new dialog box. If the call completes successfully, a new icon for the device descriptor appears in the tree structure on the screen.

5. Right-click on the new device icon and select **DownloadLASSched**. Browse to open the schedule file for your device. Click on the **Load&Activate** button. A new dialog box appears with identifying information for the device already filled in.
6. In the new dialog box, click on **Browse** to locate your `.ini` file that contains the LAS schedule you want to download, or enter the full path to the file, if you know it.
7. Click on the **Download** button. The NI-FBUS Communications Manager downloads the schedule to the interface board and activates it immediately.

Example 3. Read a Parameter Using TAG.PARAM Access

1. Open the NI-FBUS Dialog utility.
2. Click on the **Actions** menu and select **Open Session**.
3. Click on the **Open Session** button. If the call succeeds, the NIFB process in running and responding to requests, and a new session descriptor is created under the **Open Descriptors** icon.
4. Right-click on the session descriptor icon and select **ReadObject**.
5. In the dialog box that appears, enter the name of the parameter to read in the *`BLOCKTAG.PARAM`* format, where *`BLOCKTAG`* is the tag of the block containing the parameter, and *`PARAM`* is the name of the parameter. For example, to read the `out` parameter of an Analog Input block called FT-201, enter `FT-201.OUT`.
6. Click on the **Read** button to perform the read operation. If the call completes successfully, the NI-FBUS Dialog utility automatically determines the type of the data and displays it in the **Data** box. If the call fails, the error message appears in the **Result** box.
   - Read by name—Enter the name of the parameter in the *`BLOCKTAG.PARAM`* format.
   - Read by index—Enter the index of the parameter in the *`BLOCKTAG.INDEX`* format, where *`BLOCKTAG`* is the tag of the block containing the parameter, and *`INDEX`* is the index of the parameter.
   - Read by device description item—Enter the device description item of the parameter in the *`BLOCKTAG.ITEM`* format, where *`BLOCKTAG`* is the tag of the block containing the parameter, and *`ITEM`* is the device description item of the parameter.

Example 4. Wait for a Trend

Note You will not be able to receive any trends unless you have configured a device to generate them, and configured an interface to receive them. Use your configuration software package to do this.

1. Open the NI-FBUS Dialog utility.
2. Click on the **Actions** menu and select **Open Session**.
3. Click on the **Open Session** button. If the call succeeds, the NIFB process in running and responding to requests, and a new session descriptor is created under the **Open Descriptors** icon.
4. Right-click on the session descriptor icon and select **WaitTrend**.
5. A dialog box appears. This dialog box waits until the NI-FBUS Communications Manager receives a trend from any device on the bus. The trend data is displayed in the **Results** box when the trend is received. The **Trend** dialog continues to wait for and display trends as they are received until you close it with the **Cancel** button.

   You can wait on trends from all types of descriptors, not just session descriptors. For example, if you wait on a trend from a device descriptor, the dialog box only displays trends coming from the device that the specified descriptor represents. The same is true of link, virtual field device, and block descriptors.

To exit the NI-FBUS Dialog utility, select **Exit** from the **File** menu.

E

Technical Support Resources

Web Support

National Instruments Web support is your first stop for help in solving installation, configuration, and application problems and questions. Online problem-solving and diagnostic resources include frequently asked questions, knowledge bases, product-specific troubleshooting wizards, manuals, drivers, software updates, and more. Web support is available through the Technical Support section of `ni.com`

NI Developer Zone

The NI Developer Zone at `ni.com/zone` is the essential resource for building measurement and automation systems. At the NI Developer Zone, you can easily access the latest example programs, system configurators, tutorials, technical news, as well as a community of developers ready to share their own techniques.

Customer Education

National Instruments provides a number of alternatives to satisfy your training needs, from self-paced tutorials, videos, and interactive CDs to instructor-led hands-on courses at locations around the world. Visit the Customer Education section of `ni.com` for online course schedules, syllabi, training centers, and class registration.

System Integration

If you have time constraints, limited in-house technical resources, or other dilemmas, you may prefer to employ consulting or system integration services. You can rely on the expertise available through our worldwide network of Alliance Program members. To find out more about our Alliance system integration solutions, visit the System Integration section of `ni.com`

Worldwide Support

National Instruments has offices located around the world to help address your support needs. You can access our branch office Web sites from the Worldwide Offices section of `ni.com`. Branch office web sites provide up-to-date contact information, support phone numbers, e-mail addresses, and current events.

If you have searched the technical support resources on our Web site and still cannot find the answers you need, contact your local office or National Instruments corporate. Phone numbers for our worldwide offices are listed at the front of this manual.

Glossary

Prefix	Meaning	Value
m-	milli-	10^{-3}

Numbers

4-20 mA system
Traditional control system in which a computer or control unit provides control for a network of devices controlled by 4-20 mA signals.

A

A
Amperes.

Actuator
A device that translates electrical signals into mechanical actions.

Acyclic communication
Unscheduled communication on the bus.

Address
Character code that identifies a specific location (or series of locations) in memory.

AI
Analog Input.

Alarm
A notification the NI-FBUS Communications Manager software sends when it detects that a block leaves or returns to a particular state.

Alarm condition
A notification that a Fieldbus device sends to another Fieldbus device or interface when it leaves or returns to a particular state.

Alert
An alarm or event.

Analog
A description of a continuously variable signal or a circuit or device designed to handle such signals.

AO
Analog Output.

API
See Application Programmer Interface.

Application Programmer Interface
A message format that an application uses to communicate with another entity that provides services to it.

Argument
A value you pass in a function call. Sometimes referred to as a parameter, but this documentation uses a different meaning for parameter, which is included in this glossary.

Array
Ordered, indexed list of data elements of the same type.

Attribute
Properties of parameters.

B

Bank
The combination of one FieldPoint network module and one or more terminal bases and I/O modules.

Basic device
A device that can communicate on the Fieldbus, but cannot become the LAS.

Block
A logical software unit that makes up one named copy of a block and the associated parameters its block type specifies. The values of the parameters persist from one invocation of the block to the next. It can be a resource block, transducer block, or function block residing within a virtual field device.

Block tag
A character string name that uniquely identifies a block on a Fieldbus network.

Boolean
Logical relational system having two values, each the opposite of the other, such as true and false or zero and one.

Buffer
Temporary storage for acquired or generated data.

Bus
The group of conductors that interconnect individual circuitry in a computer. Typically, a bus is the expansion vehicle to which I/O or other devices are connected. Examples of PC busses are the ISA and PCI buses.

Bus scheduler
See Link Active Scheduler.

C

C
Celsius.

Channel
A pin or wire lead to which you apply or from which you read the analog or digital signal.

Character string name
See Tag.

Configuration object
See Linkage.

Contained parameter

A parameter that does not receive or send data and is contained within a function block.

Control loop

A set of connections between blocks used to perform a control algorithm.

Control strategy

See Function Block Application.

Cyclic

Closed-loop control.

D

Data Link Layer

The second-lowest layer in the ISO seven-layer model (layer two). The Data Link Layer splits data into frames to send on the physical layer, receives acknowledgment frames, and re-transmits frames if they are not received correctly. It also performs error checking to maintain a sound virtual channel to the next layer.

Data link time synchronization

A time distribution message broadcast on the Fieldbus so that all devices have exactly the same data link time.

dB

Decibel.

DD

See Device Description.

Descriptor

A number returned to the application by the NI-FBUS Communications Manager, used to specify a target for future NI-FBUS calls.

Device

A sensor, actuator, or control equipment attached to the Fieldbus.

Device address

A memory address that you use to access a device in a computer system.

Device Description

A machine-readable description of all the blocks and block parameters of a device.

Device Description Service

A set of functions that applications use to access Device Descriptions.

Device ID

An identifier for a device that the manufacturer assigns. No two devices can have the same device ID.

Device tag

A name you assign to a Fieldbus device.

DI

Discrete Input.

Distributed control

Process control distributed among several devices connected by network.

DLL	*See* Dynamic Link Library.
DO	Discrete Output.
Dynamic Link Library	A library of functions and subroutines that links to an application at run time.

E

Event	An occurrence on a device that causes a Fieldbus entity to send the Fieldbus event message.

F

FAS	Fieldbus Access Sublayer.
FB	Function Block.
FBAP	*See* function block application.
FF	Foundation Fieldbus.
Field device	A Fieldbus device connected directly to a Fieldbus.
Fieldbus	An all-digital, two-way communication system that connects control systems to instrumentation. A process control local area network defined by ISA standard S50.02.
Fieldbus Foundation	An organization that developed a Fieldbus network specifically based upon the work and principles of the ISA/IEC standards committees.
Foundation Fieldbus specification	The communications network specification that the Fieldbus Foundation created.
FP-3000	National Instruments network interface module for the FieldPoint I/O system.
FTP	File Transfer Protocol.
Function block	A named block consisting of one or more input, output, and contained parameters. The block performs some control function as its algorithm. Function blocks are the core components you control a system with. The Fieldbus Foundation defines standard sets of function blocks. There are ten function blocks for the most basic control and I/O functions. Manufacturers can define their own function blocks.
Function Block Application	The block diagram that represents your control strategy.

Function Block Application Editor window

The middle window of the NI-FBUS Configurator where you create your block diagram.

Function block execution schedule

A list of times in the macrocycle when the function block will begin to execute its algorithm.

H

H1

The 31.25 kbit/second type of Fieldbus.

hex

Hexadecimal. A base-16 numbering system which uses 0–9 and A–F.

HMI

Human-Machine Interface. A graphical user interface for the process with supervisory control and data acquisition capability.

Host device

A computer or controller on a Fieldbus network.

HSE

High Speed Ethernet.

I

Index

An integer that the Fieldbus specification assigns to a Fieldbus object or a device that you can use to refer to the object. A value in the object dictionary used to refer to a single object.

Input parameter

A block parameter that receives data from another block.

I/O

Input/output.

IRQ

Interrupt request.

ISO

International Organization for Standardization. A technical standards organization that creates international technical standards for computers and communications. The ISO is composed of national standards organizations in 89 countries. The American National Standards Institute (ANSI) represents the United States in the ISO.

Isolation

A type of signal conditioning in which you isolate the transducer signals from the computer for safety purposes. This protects you and your computer from large voltage spikes and makes sure the measurements from the devices are not affected by differences in ground potentials.

L

LAS	*See* Link Active Scheduler.
Link	A FOUNDATION Fieldbus network is made up of devices connected by a serial bus. This serial bus is called a link (also known as a segment).
Link Active Schedule	A schedule of times in the macrocycle when devices must publish their output values on the Fieldbus.
Link Active Scheduler	The Fieldbus device that is currently controlling access to the Fieldbus. A device that is responsible for keeping a link operational. The LAS executes the link schedule, circulates tokens, distributes time, and probes for new devices.
Link master device	A device that is capable of becoming the LAS
Linkage	A connection between function blocks.
Linkage object	An object resident in a device that defines connections between function block input and output across the network. Linkage objects also specify trending connections.
LM	Link Master.
Lookout	National Instruments Lookout is a full-featured object-based automation software system that delivers unparalleled power and ease of use in demanding industrial measurement and automation applications.
Loop	*See* Control loop.

M

Macrocycle	The least common multiple of all the loop times on a given link, or one iteration of a the process control loop.
Manufacturer's identification	An identifier used to correlate the device type and revision with its device description and device description revision.
Menu	An area accessible from the command bar that displays a subset of the possible command choices. In the NI-FBUS Configurator, refers to menus defined by the manufacturer for a given block.
Method	Methods describe operating procedures to guide a user through a sequence of actions.
Mode	Type of communication.

N

Nifb.exe
The NIFB process that must be running in the background for you to use your AT-FBUS or PCMCIA-FBUS interface to communicate between the board and the Fieldbus.

NI-FBUS API
The NI-FBUS Communications Manager.

NI-FBUS Communications Manager
Software shipped with National Instruments Fieldbus interfaces that lets you read and write values. It does not include configuration capabilities.

NI-FBUS Configurator
National Instruments Fieldbus configuration software. With it, you can set device addresses, clear devices, change modes, and read and write to the devices.

NI-FBUS Fieldbus Configuration System
See NI-FBUS Configurator.

NI-FBUS process
Process that must be running in the background for you to use your AT-FBUS or PCMCIA-FBUS interface to communicate between the board and the Fieldbus.

Non-volatile memory
Memory that does not require electricity to hold data.

O

Object
An element of an object dictionary.

Object description
Describes data that is communicated over the Fieldbus.

Object Dictionary
A structure in a device that describes data that can be communicated on the Fieldbus. The object dictionary is a lookup table that gives information such as data type and units about a value that can be read from or written to a device.

Octet
A single 8-bit value.

OD
See Object Dictionary.

Offline
Not connected to or installed in the computer.

OOS
Out of Service mode.

OPC
OLE for Process Control.

Output parameter
A block parameter that sends data to another block.

P

Parameter
One of a set of network-visible values that makes up a function block.

PC
Personal Computer.

PCMCIA
Personal Computer Memory Card International Association.

PD
Proportional Derivative.

PDU
Protocol Data Unit.

PID
Proportional/Integral/Derivative. A common control function block algorithm that uses proportions, integrals, and derivatives in calculation.

PLC
See Programmable Logic Controller.

PN
Probe Node.

Poll
To repeatedly inspect a variable or function block to acquire data.

Port
A communications connection on a computer or remote controller.

Process variable
A common Fieldbus function block parameter representing some value in the process being controlled.

Programmable Logic Controller
A device with multiple inputs and outputs that contains a program you can alter.

PT
Pass Token.

Publisher
A device that has at least one function block with its output value connected to the input of another device.

Publishing schedule
Determines when data members are published over the Fieldbus.

PV
Process Variable.

R

RA
Ratio.

Resource block
A special block containing parameters that describe the operation of the device and general characteristics of a device, such as manufacturer and device name. Only one resource block per device is allowed.

S

s

Seconds.

Sample type

Specifies how trends are sampled on a device, whether by averaging data or by instantaneous sampling.

Scheduled/cyclic communications

Communication that occurs at the same time during each control cycle.

Segment

See Link.

Sensor

A device that responds to a physical stimulus (heat, light, sound, pressure, motion, flow, and so on), and produces a corresponding electrical signal.

Server

Device that receives a message request.

Service

Services allow user applications to send messages to each other across the Fieldbus using a standard set of message formats.

Session

A communication path between an application and the NI-FBUS Communications Manager.

Stack

A set of hardware registers or a reserved amount of memory used for calculations or to keep track of internal operations.

Stale

Data that has not been updated for `stale_limit` number of macrocycles, where the stale limit is a parameter of the connection.

Subscriber

A device that has at least one function block with its input value connected to the output of another device.

Symbol file

A Fieldbus Foundation or device manufacturer-supplied file that contains the ASCII names for all the objects in a device.

T

Tag

A name you can define for a block, virtual field device, or device.

Timeout

A period of time after which an error condition is raised if some event has not occurred.

Traditional system

See 4-20 mA system.

Transducer block

A block that is an interface to the physical, sensing hardware in the device. It also performs the digitizing, filtering, and scaling conversions needed to present input data to function blocks, and converts output data from function blocks. Transducer blocks decouple the function blocks from the hardware details of a given device, allowing generic indication of function block input and output. Manufacturers can define their own transducer blocks.

Trend

A Fieldbus object that allows a device to sample a process variable periodically, then transmit a history of the values on the network.

Trend function

An NI-FBUS call related to trends.

U

Unscheduled

Messages sent on the Fieldbus between transmissions of scheduled messages.

Upstream

Fewer network hops away from a backbone or hub. For example, a small ISP that connects to the Internet through a larger ISP that has their own connection to the backbone is downstream from the larger ISP, and the larger ISP is upstream from the smaller ISP.

V

VFD

See Virtual Field Device.

View objects

Predefined groupings of parameter sets that HMI applications use.

Virtual Field Device

The virtual field device is a model for remotely viewing data described in the object dictionary. The services provided by the Fieldbus Messaging Specification allow you to read and write information about the object dictionary, read and write the data variables described in the object dictionary, and perform other activities such as uploading/downloading data and invoking programs inside a device. A model for remotely viewing data described in the object dictionary.

Index