FOUNDATION™ Fieldbus

NI-FBUS Configurator User Manual

January 2014
370514J-01

NATIONAL INSTRUMENTS™

Worldwide Technical Support and Product Information

ni.com

Worldwide Offices

Visit ni.com/niglobal to access the branch office Web sites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

National Instruments Corporate Headquarters

11500 North Mopac Expressway Austin, Texas 78759-3504 USA Tel: 512 683 0100

For further support information, refer to the *Technical Support and Professional Services* appendix. To comment on National Instruments documentation, refer to the National Instruments website at ni.com/info and enter the Info Code feedback.

© 2002–2014 National Instruments. All rights reserved.

Legal Information

Warranty

NI devices are warranted against defects in materials and workmanship for a period of one year from the invoice date, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace equipment that proves to be defective during the warranty period. This warranty includes parts and labor.

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 the invoice date, 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.

National Instruments respects the intellectual property of others, and we ask our users to do the same. NI software is protected by copyright and other intellectual property laws. Where NI software may be used to reproduce software or other materials belonging to others, you may use NI software only to reproduce materials that you may reproduce in accordance with the terms of any applicable license or other legal restriction.

End-User License Agreements and Third-Party Legal Notices

You can find end-user license agreements (EULAs) and third-party legal notices in the following locations:

- Notices are located in the `<National Instruments>\_Legal Information` and `<National Instruments>` directories.
- EULAs are located in the `<National Instruments>\Shared\MDF\Legal\license` directory.
- Review `<National Instruments>\_Legal Information.txt` for more information on including legal information in installers built with NI products.

Trademarks

Refer to the *NI Trademarks and Logo Guidelines* at `ni.com/trademarks` for more information on National Instruments trademarks.

ARM, Keil, and µVision are trademarks or registered of ARM Ltd or its subsidiaries.

LEGO, the LEGO logo, WEDO, and MINDSTORMS are trademarks of the LEGO Group. ©2014 The LEGO Group.

TETRIX by Pitsco is a trademark of Pitsco, Inc.©2014

FIELDBUS FOUNDATION™ and FOUNDATION™ are trademarks of the Fieldbus Foundation.

EtherCAT® is a registered trademark of and licensed by Beckhoff Automation GmbH.

CANopen® is a registered Community Trademark of CAN in Automation e.V.

DeviceNet™ and EtherNet/IP™ are trademarks of ODVA.

Go!, SensorDAQ, and Vernier are registered trademarks of Vernier Software & Technology. Vernier Software & Technology and `vernier.com` are trademarks or trade dress.

Xilinx is the registered trademark of Xilinx, Inc.

Taptite and Trilobular are registered trademarks of Research Engineering & Manufacturing Inc.

FireWire® is the registered trademark of Apple Inc.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Handle Graphics®, MATLAB®, Real-Time Workshop®, Simulink®, Stateflow®, and xPC TargetBox® are registered trademarks, and TargetBox™ and Target Language Compiler™ are trademarks of The MathWorks, Inc.

Tektronix®, Tek, and Tektronix, Enabling Technology are registered trademarks of Tektronix, Inc.

The Bluetooth® word mark is a registered trademark owned by the Bluetooth SIG, Inc.

The ExpressCard™ word mark and logos are owned by PCMCIA and any use of such marks by National Instruments is under license.

The mark LabWindows is used under a license from Microsoft Corporation. Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

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

Members of the National Instruments Alliance Partner Program are business entities independent from National Instruments and have no agency, partnership, or joint-venture relationship with National Instruments.

Patents

For patents covering National Instruments products/technology, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your media, or the *National Instruments Patent Notice* at `ni.com/patents`.

Export Compliance Information

Refer to the *Export Compliance Information* at `ni.com/legal/export-compliance` for the National Instruments global trade compliance policy and how to obtain relevant HTS codes, ECCNs, and other import/export data.

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.

Contents

Chapter 1
NI-FBUS Configurator Overview

Introduction to the NI-FBUS Configurator 1-1
NI-FBUS Configurator Windows 1-1
- Device Type Tree 1-2
- Configuration Tree 1-3
- Help Window 1-3
- Status Window 1-4
  - Status Tab 1-4
  - Download Tab 1-4
  - Errors Tab 1-4
  - Assignment Tab 1-4
  - Device Live List Tab 1-5

Middle Frame 1-5
- Function Block Application Editor Window 1-7
- Schedule Window 1-8
- Device Window 1-9

Network Parameters Window 1-10
Block Configuration Window 1-10
- Changing Parameter Settings 1-11
- Block Configuration Window Icons 1-12
- Block Configuration Window Tabs 1-12
- Block Configuration Window Units 1-12

Chapter 2
Using the NI-FBUS Configurator

Start the NI-FBUS Configurator and NIFB Process 2-1
Create a New Project 2-1
Open an Existing Project 2-3
Take Link Offline 2-3
Add Offline Devices 2-4
Take Link Online 2-6
Upload Project Configuration 2-8
Set Device Addresses 2-8
View Device Properties 2-9
- Customize the Device Image/Icon 2-10

Function Block Instantiation and Deletion 2-11
Set Device or Block Tags 2-11
Function Block Modes 2-12
- The MODE_BLK Parameter 2-12

Configure Block Parameters 2-13
Categorizing Block Parameters 2-13
Adding Tabs 2-14
Customizing Parameters on a Tab 2-14
Removing Tabs 2-15
Modify Block Configuration Window View 2-15
Reading the Default Cff Values of Block Parameters 2-15
Editing Block Parameters 2-16
Updating Block Parameters Values 2-17
Manually Update Block Parameters 2-17
Automatically Update Block Parameters 2-17
Create and Edit Function Block Applications 2-17
Add Blocks to the Function Block Application 2-18
Connecting Blocks 2-19
Wiring Blocks Manually 2-19
Checking for Function Block Application Errors 2-20
Customizing Your Function Block Application 2-20
Using Templates 2-21
Inserting Templates 2-21
Creating Templates 2-24
Export Function Block Application Image 2-24
Defining Loops 2-24
Defining Multiple Loops Running at Different Rates 2-25
Determining Function Block Execution Time 2-27
Changing the Block Execution Order 2-27
Setting the Stale Limit 2-28
Using Menus and Methods 2-29
Using EDDL Extensions 2-29
Configuring Alarms 2-32
Configuring Trends 2-33
View and Edit a Schedule 2-34
Change the Link Active Schedule 2-34
Multiple Loop Representation 2-35
Set Network Parameters 2-35
Change Language 2-36
Save a Project Configuration 2-36
Download a Project Configuration 2-36
Download Options 2-37
Write Contained Block Parameters 2-37
Clear Devices 2-38
Automatic Mode Handling 2-38
Verify and Diff 2-38
Bringing a Block Online 2-38
Monitoring Parameters 2-39

Verify a Configuration 2-39
Quick Verification 2-39
Compare Two Configurations 2-40
Check for Errors 2-40
Add An Additional Function Block Application 2-40
Replace Devices 2-41
Reset a Device to Factory Defaults 2-41
Change Device Type 2-41
Export & Import Parameters 2-42
Set Preferences 2-43
Changing the Software Key 2-44
Print 2-44
Log Information 2-45

Chapter 3
NI-FBUS Configurator Tutorials

Tutorial 1: Connect to Network and Hardware 3-1
Tutorial 2: Create Basic Connections to AI and AO Blocks 3-2
Tutorial 3: PID Control of a Simulated Process 3-4
Tutorial 4: Create Offline Project 3-7
Tutorial 5: Modify Saved Project 3-8

Chapter 4
Parameters

Alarming Parameters 4-1
Diagnostic Parameters 4-2
Faultstate Parameters 4-2
Scaling Parameters 4-3
I/O Parameters 4-7
Limiting Parameters 4-7
Mode Shedding Parameters 4-8
Option Parameters 4-8
Parameters That Must Be Initialized 4-9
Process Parameters 4-9
Tuning Parameters 4-10
Parameter Descriptions 4-10
ACK_OPTION (Alarming) 4-10
ALARM_HYS (Alarming) 4-10
ALARM_SUM (Alarming) 4-10
ALERT_KEY (Alarming) 4-11
BAL_TIME (Tuning) 4-11
BIAS (Tuning) 4-11
BKCAL_HYS (Limiting) 4-11
BKCAL_IN (Limiting, Process) 4-11

BKCAL_OUT (Process) 4-11
BKCAL_OUT_D (Process) 4-11
BLOCK_ALM (Alarming, Diagnostic) 4-11
BLOCK_ERR (Diagnostic) 4-12
BYPASS (Scaling, Tuning) 4-13
CAS_IN (Process) 4-13
CAS_IN_D (Process) 4-13
CHANNEL (I/O, Process) 4-13
CLR_FSTATE (Faultstate, Option) 4-14
CONFIRM_TIME (Alarming) 4-14
CONTROL_OPTS (Option, Scaling) 4-14
CYCLE_SEL (Tuning) 4-15
CYCLE_TYPE (Tuning) 4-15
DD_RESOURCE (Diagnostic) 4-15
DD_REV (Diagnostic) 4-15
DEV_REV (Diagnostic) 4-15
DEV_TYPE (Diagnostic) 4-15
DISC_ALM (Alarming) 4-15
DISC_LIM (Alarming) 4-15
DISC_PRI (Alarming) 4-15
DV_HI_ALM (Alarming) 4-16
DV_HI_LIM (Alarming) 4-16
DV_HI_PRI (Alarming) 4-16
DV_LO_ALM (Alarming) 4-16
DV_LO_LIM (Alarming) 4-16
DV_LO_PRI (Alarming) 4-16
FAULT_STATE (Faultstate, Option) 4-16
FEATURE_SEL/FEATURES (Diagnostic, Option) 4-16
FF_GAIN (Scaling, Tuning) 4-17
FF_SCALE (Scaling) 4-17
FF_VAL (Process, Scaling, Tuning) 4-17
FIELD_VAL (Process, Scaling, Tuning) 4-17
FIELD_VAL_D (Process, Scaling, Tuning) 4-17
FREE_SPACE (Diagnostic, Process) 4-17
FREE_TIME (Diagnostic, Process) 4-17
FSTATE_TIME (Faultstate, Option) 4-18
FSTATE_VAL (Faultstate, Option) 4-18
FSTATE_VAL_D (Faultstate, Option) 4-18
GAIN (Tuning) 4-18
GRANT_DENY (Option) 4-18
HARD_TYPES (I/O, Process) 4-18
HI_ALM (Alarming) 4-19
HI_HI_ALM (Alarming) 4-19
HI_HI_LIM (Alarming) 4-19
HI_HI_PRI (Alarming) 4-19

HI_LIM (Alarming) 4-19
HI_PRI (Alarming) 4-19
IN (Process, Scaling, Tuning) 4-19
IN_1 (Process, Scaling, Tuning) 4-19
IO_OPTS (I/O, Options, Scaling) 4-19
ITK_VER 4-20
L_TYPE (Scaling) 4-20
LIM_NOTIFY (Alarming) 4-21
LO_ALM (Alarming) 4-21
LO_LIM (Alarming) 4-21
LO_LO_ALM (Alarming) 4-21
LO_LO_LIM (Alarming) 4-21
LO_LO_PRI (Alarming) 4-21
LO_PRI (Alarming) 4-21
LOW_CUT (I/O, Option, Scaling, Tuning) 4-22
MANUFAC_ID (Diagnostic) 4-22
MAX_NOTIFY (Alarming) 4-22
MEMORY_SIZE (Diagnostic) 4-22
MIN_CYCLE_T (Diagnostic, Process) 4-22
MODE_BLK (Diagnostic, Process) 4-22
NV_CYCLE_T (Diagnostic) 4-24
OUT (Process, Scaling, Tuning) 4-24
OUT_D (Process) 4-24
OUT_HI_LIM (Limiting) 4-24
OUT_LO_LIM (Limiting) 4-24
OUT_SCALE (Scaling) 4-24
OUT_STATE (Process) 4-25
PV (Process, Scaling, Tuning) 4-25
PV_D (Process) 4-25
PV_FTIME (Scaling, Tuning) 4-25
PV_SCALE (Scaling) 4-25
PV_STATE (Process) 4-25
RA_FTIME (Tuning) 4-25
RATE (Tuning) 4-26
RCAS_IN (Mode Shedding, Process) 4-26
RCAS_IN_D (Mode Shedding, Process) 4-26
RCAS_OUT (Process) 4-26
RCAS_OUT_D (Process) 4-26
READBACK (Scaling, Tuning) 4-26
READBACK_D (Scaling, Tuning) 4-26
RESET (Tuning) 4-26
RESTART (Diagnostic, Option) 4-26
ROUT_IN (Mode Shedding, Process) 4-27
ROUT_OUT (Process) 4-27
RS_STATE (Diagnostic, Process) 4-27

SEL_1 through SEL_3 (Process, Scaling, Tuning) 4-27
SEL_TYPE (Scaling) 4-28
SET_FSTATE (Faultstate, Option) 4-28
SHED_OPT (Mode Shedding, Option) 4-28
SHED_RCAS (Mode Shedding) 4-28
SHED_ROUT (Mode Shedding) 4-28
SIMULATE (Option) 4-28
SIMULATE_D (Option) 4-29
SP (Process) 4-29
SP_D (Process) 4-29
SP_HI_LIM (Limiting, Option) 4-29
SP_LO_LIM (Limiting, Option) 4-29
SP_RATE_DN (Limiting, Option) 4-29
SP_RATE_UP (Limiting, Option) 4-29
ST_REV (Diagnostic) 4-29
STATUS_OPTS (Faultstate, Limiting, Option) 4-30
STRATEGY 4-31
TAG_DESC (Diagnostic) 4-31
TEST_RW (Process) 4-31
TRK_IN_D (Scaling) 4-31
TRK_SCALE (Scaling) 4-31
TRK_VAL (Scaling) 4-31
UPDATE_EVT (Diagnostic) 4-31
WRITE_ALM (Alarming) 4-31
WRITE_LOCK (Option) 4-31
WRITE_PRI (Alarming, Option) 4-31
XD_SCALE (Scaling) 4-32
XD_STATE (Process) 4-32

Appendix A
Error Messages and Warnings

Error Messages A-1
Warnings A-12

Appendix B
Troubleshooting and Common Questions

Mode Problems B-1
Bad Status B-3
Missing Device B-4
Read/Write Errors B-6
Set Address Problems B-7
    Setting Number of Polled Addresses B-7
Performance Problems B-8
Missing Link B-8

Missing Information B-10
Missing Symbolic Information B-10
Add/Delete Tabs B-10
Missing Function Blocks B-11
Viewing Problems B-12
Using Fieldbus with Lookout B-12
Open/Find Project B-13
Using NIFB B-13
Restart Devices B-13
Using Fieldbus with Server Explorer B-13
Uninstalling the Software B-14

Appendix C
Cascade and Remote Cascade Setups

Cascade Initialization C-1
Parameter Connections for Cascade Initialization C-1
Mode and Status Behavior During Cascade Initialization C-2
Remote Cascades C-3
Bypassing Cascade Initialization C-3

Appendix D
Technical Support and Professional Services

Glossary

Index

1

NI-FBUS Configurator Overview

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

This manual assumes that you are already familiar with the Windows operating system and have read the *FOUNDATION Fieldbus Overview* manual or are otherwise familiar with using FOUNDATION™ Fieldbus.

If you do not install the hardware, you can use only the offline functions of the NI-FBUS Configurator.

Refer to the *NI-FBUS Installation Guide* for more information about installing the NI-FBUS software and hardware. This manual is available in PDF-format in the *NI-FBUS Software* media and in printed-format shipped with the media.

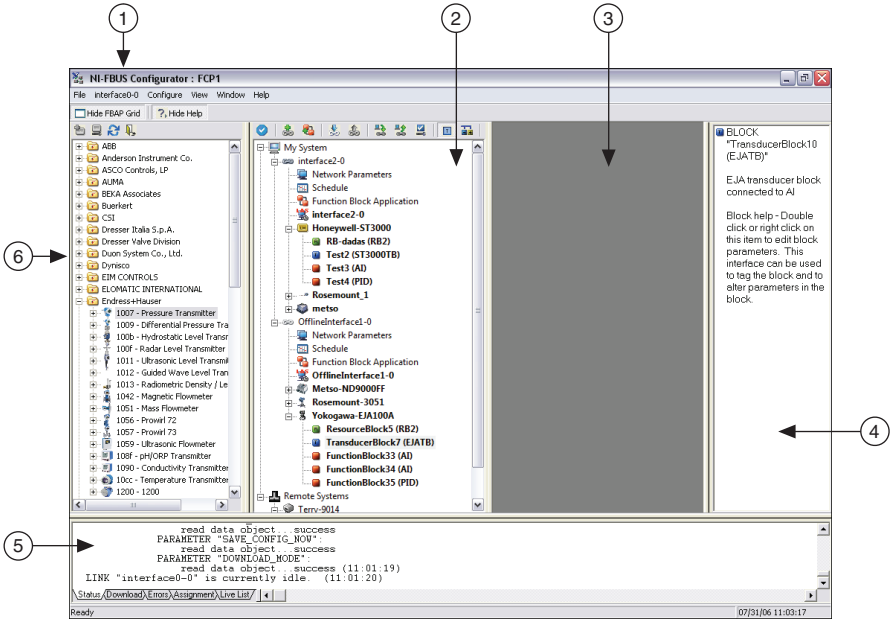
Introduction to the NI-FBUS Configurator

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* manual.

NI-FBUS Configurator Windows

The NI-FBUS Configurator has four resizable windows within the main window: the configuration tree, help window, status window, and device type window. You can open additional specialty windows in the middle frame to configure your function block applications, change parameters, and update the schedule. You also can open separate windows for block parameter adjustment on top of the main window. The elements of the NI-FBUS Configurator main window are shown in Figure 1-1.

Figure 1-1. NI-FBUS Configurator Main Window Elements



1	Object Menu (based on object selected in configuration tree)	4	Help Window
2	Configuration Tree	5	Status Window
3	Middle Frame	6	Device Type Tree

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 the active window in the middle frame. To see other windows in the middle frame, minimize the active window in the middle frame.

Device Type Tree

The device type tree window, which appears to the most left side of NI-FBUS Configurator, displays the existing device types already imported into NI-FBUS. The device types are grouped by vendor name. Click the + symbol to view the device types belonging to each manufacturer. Click the **Hide Device Type Window** button to hide the device type window; select **Window» Device Type Window** to show the device type window.

Note You can dock the device type window on the right side of the main window. To do so, drag the device type window and drop to the right side of the main window, or select **View»Preferences**, select the **General** page, and check the **Dock Device Type Window in the Right Side** box to move the device type window to the right side.

Configuration Tree

The configuration tree appears to the left of the middle frame. This tree displays local and remote configurable objects that connect 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 also can view the items on this menu by right-clicking the object.

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

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

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

During the initial bus scan, 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 Objects may become invalid briefly during the initial bus scan or when changing the device address because the device suspends communication while setting the 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 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 **Download** tab is updated only when a configuration is downloaded to the bus.

Errors Tab

The **Errors** tab displays errors in the local and remote targets. These errors include all recent I/O errors, project configuration errors, and warnings in the current project. To fix an error, double-click 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 the **Check Project for Errors** toolbar button.

Assignment Tab

The **Assignment** tab shows all the status information for the process of assignment when taking a link online. The NI-FBUS Configurator retains this information until the next assignment. The **Assignment** tab automatically appears on top when you begin to assign your devices. The **Assignment** tab is updated only when an offline link is taken online.

Device Live List Tab

The **Device Live List** tab is a quick view window to display all the live links and devices on the local and remote targets. This tab lists the tags, device IDs, and addresses of devices. The **Device Live List** automatically refreshes according to the current links/devices status within the fieldbus segments.

Note The **Device Live List** tab does not detect new links or new targets automatically.

Tip By default, the **Device Live List** tab shows the online links/devices only. The offline links/devices will be shown only when the **Show offline link** is checked.

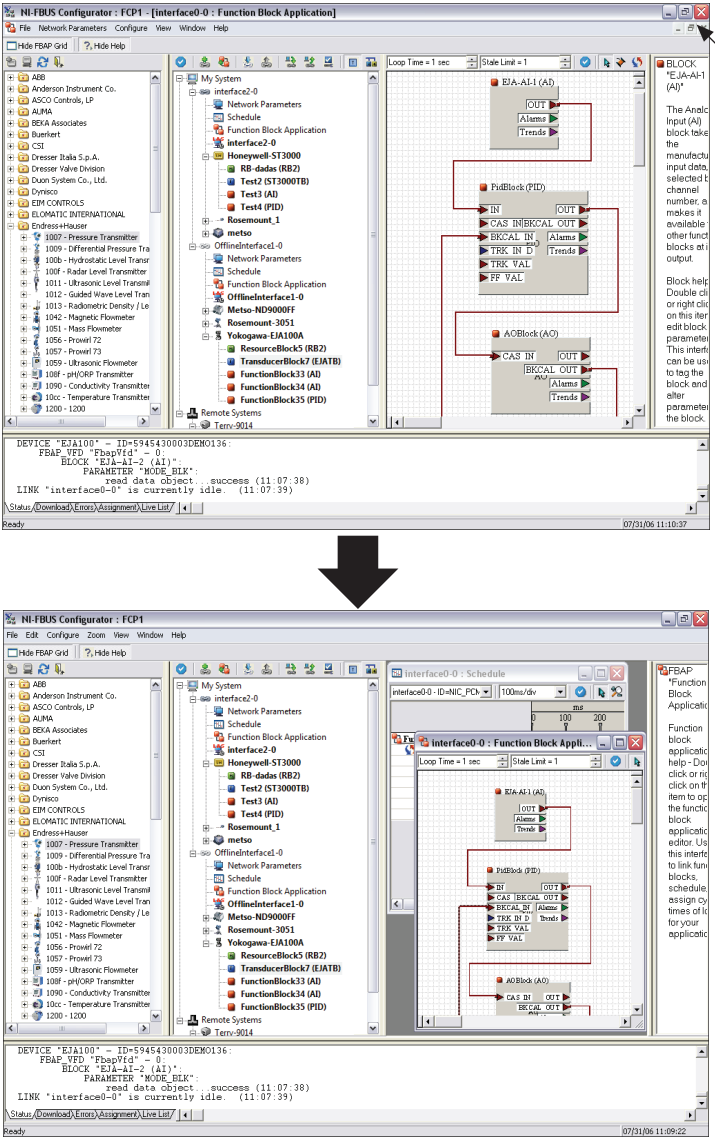
Use the drop-down list at the upper-left corner to filter the links. **Device Live List** shows the link only when the corresponding box in the drop-down list is checked.

Middle Frame

The contents of the middle frame are determined by what you choose to open there. Double-clicking 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, or any device.

Double-clicking 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 Figure 1-2.

Figure 1-2. Window View Options



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 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, the 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, refer to the *Wiring Blocks Manually* section of Chapter 2, *Using the NI-FBUS Configurator*.

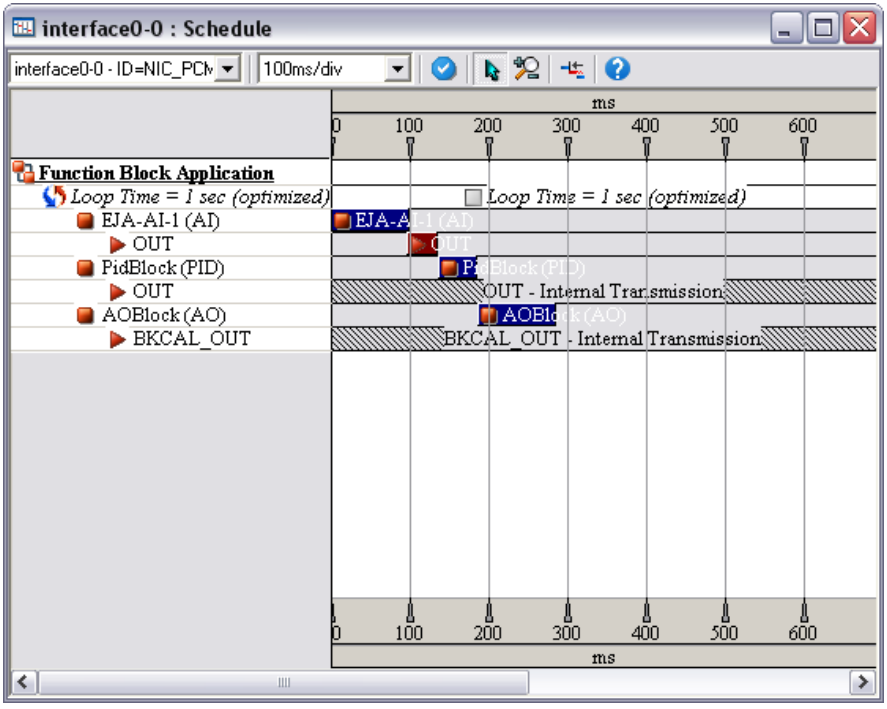
To switch between the pointer, wiring, and loop tools in the Function Block Application Editor window, press the <Tab> key. To switch between the pointer and the wiring tool, press the spacebar.

You can drag and drop block or device icons from one window into another. Use this method to drag 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 Figure 1-3.

Figure 1-3. Link Active and Function Block Schedules



To view or edit the link active schedule, double-click 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 cannot see data transmission because the device does not transmit any parameters to other devices. For some fast devices, the blue block execution bars are so short that the blocks are entirely hidden behind the block icons.

Note An optimized schedule can show that the transmission time overlaps the end of function block execution 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 the execution time next to the loop icon and enter the new time.

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 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 the **Device** icon to open the device window for a field device. The icon is followed by the device tag and its unique serial identifier.

- Double-click the **HSE/H1 Linking Device** icon to open the device window for a linking device. The icon is followed by the device tag and its unique serial identifier.

Tip The **Host** device only occurs in the H1 segment, and the **HSE/H1 Linking Device** only occurs in the HSE segment.

- This icon represents the **H1** segment.

- This icon represents the **HSE** segment.

- This icon represents the **H1 Offline** link.

Network Parameters Window

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

Caution This feature is disabled in the HSE segment. Do *not* modify these parameters without good reason. If you must modify parameters for certain devices, refer to the recommend settings of the device manufacturer. 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

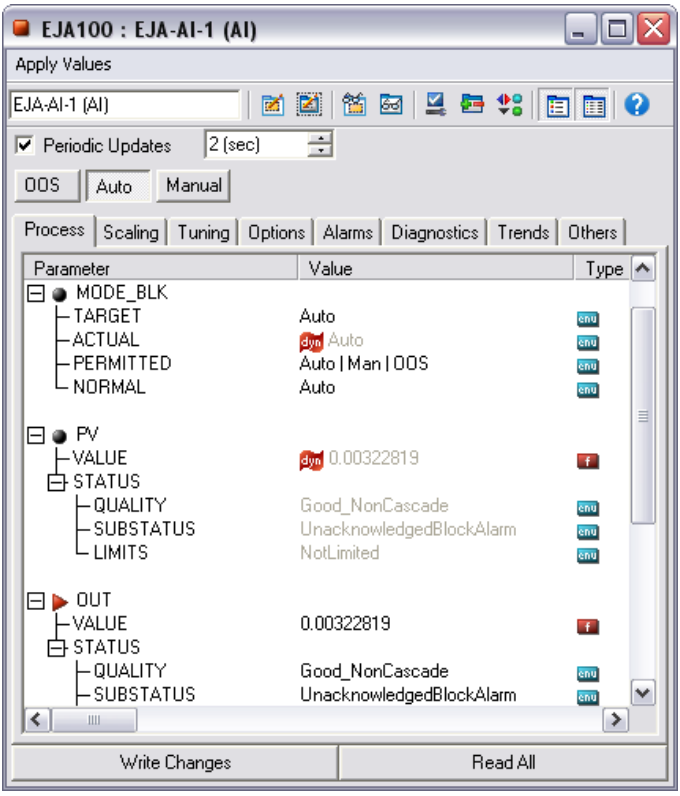
Each function block in your application has a block configuration window that you can use to change block parameters and other settings. When you read or write data in the block configuration window, the **Status** tab in the status window shows the status of the transactions.

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

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

The block configuration window appears floating over the main window. A block configuration window is shown in Figure 1-4.

Figure 1-4. Block Configuration Window



Changing Parameter Settings

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

Yellow highlighting in the block configuration window indicates that a parameter 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 Light blue—Tuning Black—Others	Represents contained parameters. Contained parameters cannot be linked to other parameters; they are contained in the block.
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 and their contents that you see will vary, depending on the type of block you select.

You also can customize the tabs of the block configuration window. You can add or delete tabs, change the tab order in the window, and add or remove any block parameters from a tab. For more information, refer to the *Categorizing Block Parameters* section of 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 the software and any FOUNDATION Fieldbus interfaces.
2. Install device descriptions.
3. Install and wire your FOUNDATION Fieldbus device(s), as instructed in the documentation that came with your device(s).
4. Launch the NI-FBUS Configurator and when prompted, click **Yes** to start `NIFB.EXE`.

   or

   Launch the NI-FBUS Communications Manager to start the NIFB process and then launch the NI-FBUS Configurator.

Note To enable the offline feature support, you must configure an offline interface first. Refer to the *Configuring an Offline Interface* topic in the *NI-FBUS Utilities Help*, to learn how to do this through the **NI-FBUS Interface Configuration Utility**.

Create a New Project

1. When you start the NI-FBUS Configurator, the **Start Up** 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 project. The multiple window structure of the NI-FBUS Configurator may become confusing with multiple open projects.

   To hide the **Start-up** dialog box the next time you launch the NI-FBUS Configurator, check **Don't display this dialog again**. To reveal the **Start-up** dialog box, select **Show New-Open Dialog at Start-up** in the **General** page of the preferences dialog under **View»Preferences**.

2. In the **Start Up** dialog box, all interfaces in the local and remote systems appear. Select the link(s) you want to include in your project in the **Added Link(s)** checklist.

3. Select **Offline Project** to filter the online links.
4. Click **OK**.

The NI-FBUS Configurator scans the FOUNDATION Fieldbus network and checks addresses and object tags. If a device does not have an address or a tag, the NI-FBUS Configurator assigns one to the device. However, the NI-FBUS Configurator does not assign addresses or tags if you deselected the **Automatically Set the pdTag and Node-address** option. If the NI-FBUS Configurator detects duplicate tags, the NI-FBUS Configurator appends a numeric identifier to the original device or block tag.

You can click the **Refresh** button to refresh links in the local and remote systems. You need to install the NI-FBUS cRIO support feature to use this button.

Note You must launch the NI-FBUS Communications Manager to access local links.

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 bus scan, 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 Objects become invalid during the initial bus scan or when changing the device address. The reason is that the device suspends communication while setting the device address.

Open an Existing Project

To open an existing project, complete the following steps.

1. When you start the NI-FBUS Configurator, the **Start Up** dialog box appears automatically. If you want to open an existing project at any other time, select **File»Open**, browse to the file you want to edit, and click **Open**.
2. In the **Start-up** dialog box, the recently accessed files appear in the bottom listbox.
3. Select the **Open an Existing Project** option.
4. (Optional) To open other files, double-click **More Files** in the listbox.
5. Choose the file item in the listbox, click the **OK** button, or double-click the file item to open it directly.

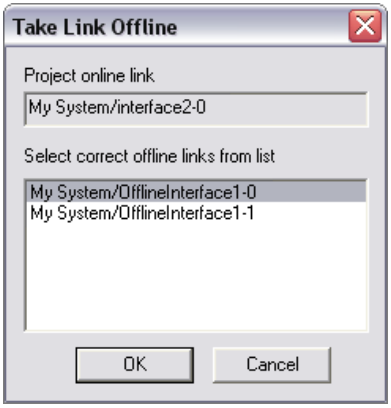
Take Link Offline

Taking a link offline moves a local online link to an offline state. Working with an offline link is faster than working with an online link. You can take a local link offline to speed up configuration, such as setting pdTag and node address, and then take the link back online for use.

Note You cannot take a remote link offline.

To take a local link offline, complete the following steps:

1. Configure an offline interface by following the instructions in the *Configuring an Offline Interface* topic of the *NI-FBUS Utilities Help*.
2. Right-click the link and select **Take Offline**, or select the link and click the **Take Offline** button on the toolbar. The **Take Link Offline** dialog appears:



3. Select the appropriate link in the **Take Link Offline** dialog box, then click **OK**.

Caution If you take a local link offline during loading or initialization, some parameters may not complete the loading or initialization process. The values of these parameters become invalid.

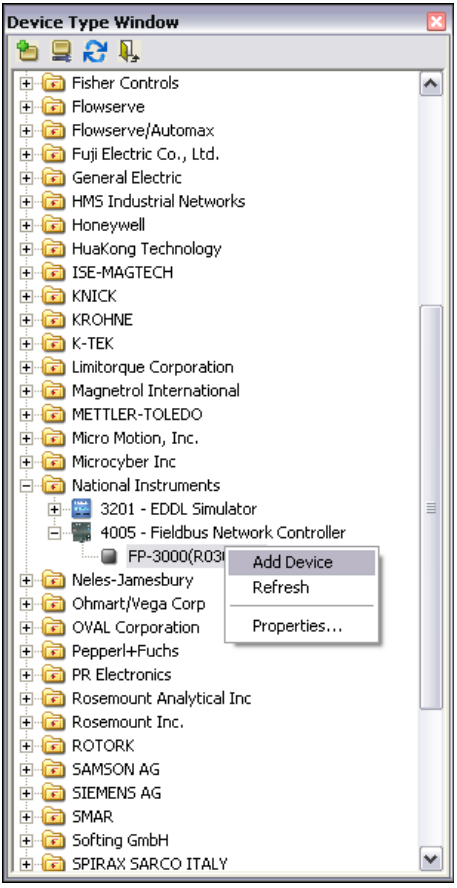
Add Offline Devices

To add offline devices to an offline link, complete the following steps:

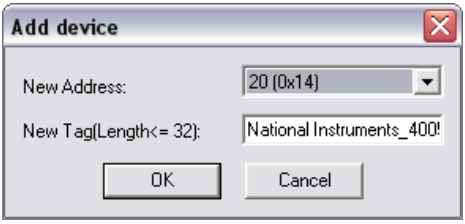
1. Select the **Window»Device Type Window** menu item if the device type window is not already shown.
2. Drag an offline device from the device type tree in the left-hand window and drop the device onto the active offline link of the configuration tree in the project window.

The device type window shows the devices in hierarchy and group the devices from the same vendor under a folder. You can right-click the device types and select **Property** to view detailed information.

Figure 2-1. Adding an Offline Device



3. When you add an offline device to the configuration tree, the **Add device** window prompts you to set the FBUS network address and tag of the offline device. The NI-FBUS Configurator sets the default values. The NI-FBUS Configurator checks the default address and tag to ensure the address and tag do not conflict with other devices. You also can set a different address and tag. Click **OK** to add this offline device.



Caution The NI-FBUS Configurator creates offline devices on the basis of `.ffo` and `.cff` files. If you use incorrect `.ffo` or `.cff` files to create offline devices, the devices may not work properly.

Note You must configure offline interfaces before adding interfaces to the project and adding offline devices to the interfaces. Refer to the n the *Configuring an Offline Interface* topic of the *NI-FBUS Utilities Help*, to learn how to configure interfaces through **NI-FBUS Interface Configuration Utility**.

Tip To assure the device type list indicates the latest `.cff` files, click the **Refresh** button on the toolbar of the device type window. Click the **Hide Device Type Window** button to hide the device type window when not adding offline devices.

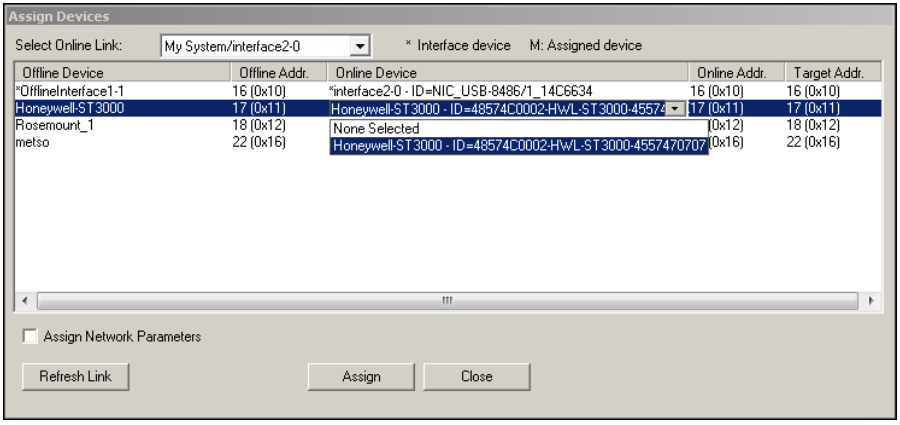
Take Link Online

Taking a link online moves an offline link to the online state. You can take a link online before downloading the offline configuration to the field network.

To take a link online, complete the following steps:

1. Select an offline link and click the **Take Online** button on the toolbar, or right-click the offline link and select **Take Online**. The **Assign Devices** dialog box appears as shown in Figure 2-2.

Figure 2-2. The Assign Devices Box



2. In the **Assign Devices** dialog box, select the online link to which you want to connect in the **Select Online Link** combo-box.
3. After the online link loads completely, select an online device from the **Device List** window. The NI-FBUS Configurator assigns the device to the selected offline device.
4. Click within the **Target Addr** column to select an address. The NI-FBUS Configurator assigns this address to the selected online device.
5. Click the **Refresh Link** button to reload the selected online link.
6. Select the **Assign Network Parameters** option to assign network parameters.
7. Click the **Assignment** button to enter the assignment.

The whole process may take a few minutes or longer. The assignment status window records all warning messages during the process.

Tip You can compare the configurations before and after assignment.

To compare the configurations before and after assignment, complete the following steps:

1. Save the project before assignment.
2. Save the project under a different name after assignment.
3. Right-click the online link after assignment, select the **Compare Configurations** menu item.
4. In the **Compare Configurations** dialog box that appears, check the option **From file** under the **Target Link** bound box.
5. Input the file name of the project file that was saved before assignment, or click the **Browse** button to select the project file.

6. Select the offline link in the **Link** combo-box under the **Target Link** bound box.
7. Click the **Diff** button to view the results of the comparison.

Upload Project Configuration

Use the **Upload Project** button or select **Configure»Upload Configuration** to read a configuration from a configured device and overwrite the current project with the information. To prevent overwriting an existing project, close all projects and create an empty project before uploading a configuration. The NI-FBUS Configurator uploads and adds the following items to the project:

- Function block parameters
- Schedules
- Linkages for publisher or subscriber
- Alarms
- Trends

The NI-FBUS Configurator does not upload the LAS schedule. The NI-FBUS Configurator creates the LAS schedule from the list of linkages and function block schedules by the NI-FBUS Configurator's usual scheduling algorithm. When the upload is complete, you can modify, save, or download the project.

Set Device Addresses

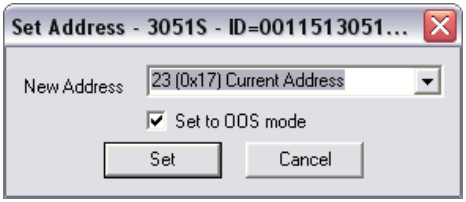
The NI-FBUS Configurator automatically attempts to set addresses for devices when devices link to the bus. You also can manually set device addresses.

Note The HSE device disables this operation for the IP address, and the host device disables this operation for the constant address. Devices may disappear off the bus when you reset the addresses.

Complete the following steps to set the device address.

1. In the configuration tree, right-click 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 Figure 2-3.

Figure 2-3. The Set Address Box



3. Check the **Set to OOS Mode** checkbox to set the block to the out of service (OOS) mode, which prevents the device from operating in its current control system.

Note **Set to OOS Mode** is disabled for offline devices.

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

4. Click the **Set** button.

   The NI-FBUS Configurator sets the device resource block to 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.

Tip To set the device address to default manually, deselect **Automatically Set the pdTag and Node-address** option in the **General** tab of the **Preferences** dialog box. You can launch this dialog box through **View»Preferences**. By default, the NI-FBUS Configurator sets the pdTag and node address of a factory default device automatically.

View Device Properties

The **Device Properties** dialog provides basic information for the device, including device identification, manufacturer information, and online/offline-related information. The **Device Properties** dialog also shows the device image if an image is available. Devices of the same device type share the same image. The NI-FBUS Configurator shows this image as the icon in the **Device Type** tree and the **Configuration** tree for these types of devices as well.

Figure 2-4. Device Properties Dialog Box

To view the properties of a device, select the device from the **Device Type** tree or **Configuration** tree. Open the property dialog by using one of the following ways:

- Right-click the device and click **Properties** from the shortcut menu.
- Double-click the device.
- **(For offline devices only)** Select the device in the **Device Type** tree and click **View Properties** in the toolbar of the **Device Type** window.

Customize the Device Image/Icon

Complete the following steps to customize the device image.

1. Open the **Device Properties** dialog box.
2. Double-click the **Device Image** area.
3. Select the image file from the **Open File** dialog box and click **OK**.

After you customize the device image, the NI-FBUS Configurator updates the corresponding icon(s) in the **Device Type** tree and the **Configuration** tree.

Tip To use the regular icons in the **Device Type** tree and the **Configuration** tree, deselect the **Use Customized Icon in the Device Type tree and Configuration tree** option in the **General** tab of the **Preferences** dialog box. You can launch this dialog

box through **View»Preferences**. By default, the NI-FBUS Configurator uses the customized icons in the trees.

Function Block Instantiation and Deletion

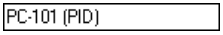
The NI-FBUS Configurator detects whether a device supports function block instantiation (creation) and deletion. If a device supports instantiation and deletion, **Instantiate new block** appears in the device object menu. Access this menu item by right-clicking the device, or using the menu bar after you select the device. When you select **Instantiate new block**, a dialog box appears to show the available block types for this device. The NI-FBUS Configurator reads the block type information 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 in 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. Launch the **Set Tag** dialog box in one of the following ways:
   - In the configuration tree, right-click the device or block whose tag you want to change and select **Set Tag**.
   - Select the object menu item from the main menu bar and select **Set Tag**.
   - Double-click the device or block in the configuration tree to launch the block configuration window. Then, click the Device Tag field in the block configuration window.



2. In the **New Tag** field, enter the tag you want to assign.
3. Make sure the **Set to OOS Mode** checkbox contains a checkmark.

Note **Set to OOS Mode** is disabled for offline devices.

Caution You cannot set tags for the host device because the device keeps in constant communication. You cannot set tags for an HSE/H1 linking device because this device contains the H1 host device. When you set the device tag, the device loses all linkage, communication configuration information, and control of the process. You should not change device tags after you have configured your process.

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

Note Setting a tag may take a few minutes.

5. After you set the tag, double-click the **block** icon to launch the block configuration window.

6. On the **Process** tab, change the block from the OOS mode to a desired mode (usually Auto) by selecting the desired Target mode from the MODE_BLK parameter. For a description of modes, refer to 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 the Automatic mode, the block runs under the normal automatic control, using a local setpoint value in the normal block algorithm to determine the output value.
- In the Cascade mode, the block receives setpoint values from another function block. The block that receives the values uses the values in the normal block algorithm to determine the output value. A linkage object connects 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 the Out Of Service mode, the block does not run at all. You need to set the function block to the Out Of Service mode to change certain parameters of some devices.
- In the Manual mode, the block does not calculate the output by using the normal block algorithm. The operator writes the output of the block directly.

Note For offline devices, the function block modes are ignored and have no effect.

The MODE_BLK Parameter

The MODE_BLK parameter for a function block is the parameter that contains information on the modes of the block. This parameter has the following four fields:

- 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.

The device manufacturer defines the PERMITTED and the NORMAL fields.

Note If the block is not currently scheduled, the block remains in the OOS mode, regardless of any writes to the TARGET field. You can schedule a block by placing the block in a Function Block Application Editor window and downloading the project to the device. Refer to the *Add Blocks to the Function Block Application* section and the *Download a Project Configuration* section for more information.

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

Configure Block Parameters

Categorizing Block Parameters

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

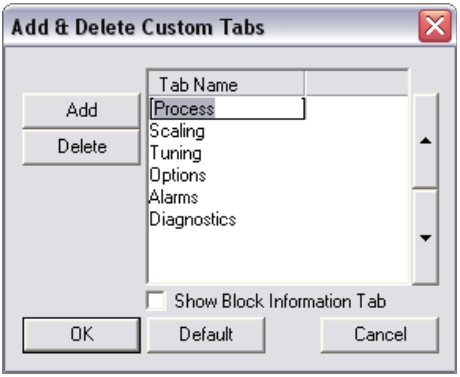
The NI-FBUS Configurator saves all custom changes on the computer, and associates the changes with this block type. Therefore, all blocks of this type in all projects use your custom setup.

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

Adding Tabs

Complete the following steps to add a tab to the block configuration window:

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



2. Click the section where you want to add the tab.
3. Click the **Add** button. Click the blank name field of the new tab and enter the name of the new tab.
4. The **Block Information** tab contains additional read-only information on each block, such as execution time. Add this tab to your view by checking the **Show Block Information Tab** checkbox.
5. Click 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 the *Customizing Parameters on a Tab* section 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 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 the name of the tab that you want to remove.
3. Click the **Delete** button.
4. Click **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 the **Simple Verify** button. In the dialog box that appears, click **Verify**, then click **Done**.

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

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

Reading the Default Cff Values of Block Parameters

The NI-FBUS Configurator reads the default parameter values of the offline device from the `.cff` file automatically after you add the offline device. You also can read the default value by using the **Set Defaults** button or the **Set Selected Default** shortcut menu item.

You can assign the default values to the online devices through an assigning configuration.

Note If the NI-FBUS Configurator cannot reads the default value from the corresponding `.cff` file, the NI-FBUS Configurator displays the value as **(NULL)**.

Caution An incorrect `.ffo` or `.cff` file can result in incorrect values read. Check control and process parameters before assigning or downloading the configuration because these files may contain multiple absent values.

Editing Block Parameters

The block configuration window groups default tabs by functions. Parameters with a plus sign are records or arrays (structures) that contain more than one entry. To expand the record or array, click the plus sign. To collapse the structure, click the minus sign. A sub-elements of a record is a field.

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

Refer to Chapter 4, *Parameters*, for complete descriptions of the parameters and the values.

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

1. Double-click in a function block to open the block configuration window.
2. If a record is collapsed, click 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. You can select the desired setting from a drop-down menu, or type in the new value. If a parameter value is grayed out, this parameter is read-only and you cannot change the value. After you change a parameter value, an asterisk (*) appears by the parameter name and the field turns yellow.
4. Click the **Write/Save Changes** button on the block configuration window toolbar to apply your changes. If the device accepts the new setting, the asterisk and yellow color disappear. If the device does not accept the new setting, ensure the setting is valid for your device. Refer to your device documentation for valid settings of this parameter.

Tip If you do not want to write or save the new value to the device, you can undo your change to the block configuration window by reading the parameter value stored in the device or `.cff` file as described in the *Manually Update Block Parameters* section.

5. If you do not click the **Write/Save Changes** button, you can apply your changes when downloading the configuration by checking the **Write Contained Block Parameters** checkbox.

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

Note You must set the device mode to the `OOS` mode to write certain parameters of some online devices. A dialog box appears to inform you of this behavior if you attempt to write 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 the **Read Selected/Set Selected Default** button in the block configuration window toolbar.

To update all the block parameter values, click the **Read All/Set Defaults** button.

Automatically Update Block Parameters

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

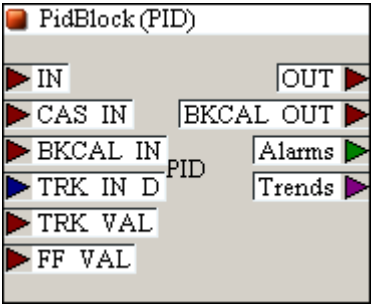
1. At the top of the block configuration window, check the **Periodic Updates** checkbox. Then, the NI-FBUS Configurator periodically checks the bus for any changes in values for the parameters and refreshes the values in the window.

   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 updates 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 NI-FBUS Configurator may not read from the bus at this 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 block diagram displays names of input and output parameters of a function block, 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

To add blocks to the Function Block Application Editor window, complete the following steps:

1. Double-click the **Function Block Application** icon in the configuration tree.

2. Drag blocks from the configuration tree to the Function Block Application Editor window.
3. Repeat the previous step until all the blocks you want to use are in the Function Block Application Editor window.

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* section and the *Configuring Trends* section.

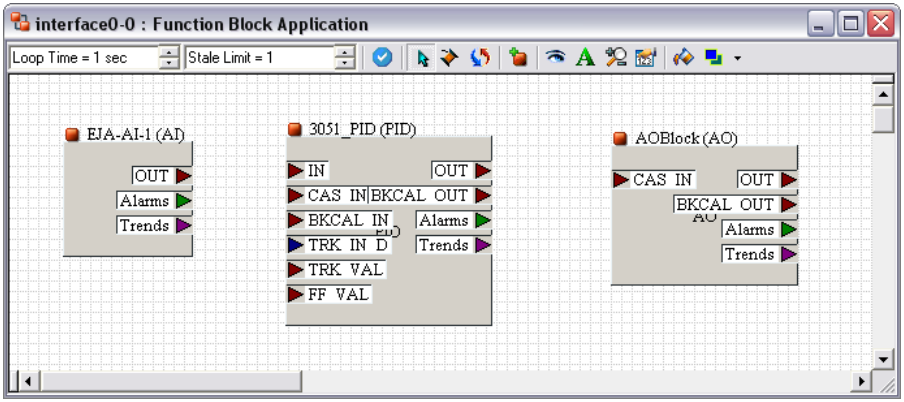
You also can add blocks by completing the following steps:

1. Right-click in the background of the Function Block Application Editor or click 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 add a function block to a Function Block Application Editor window only once. After you add the function block, the NI-FBUS Configurator grays out the function block on the menu and in the configuration tree. To use outputs of a block as inputs to multiple downstream blocks, wire the output to both locations. To use outputs of a block in multiple loops, refer to the *Defining Loops* section.

The following figure shows the Function Block Application Editor window with function blocks.



Connecting Blocks

To send data from one block to another, you must connect 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 to connect these parameters.

The NI-FBUS Configurator generates all necessary arrangements for broadcasting data over the bus when you wire the function blocks together. When you download your project to the bus, the NI-FBUS Configurator downloads the scheduling and linkage information to all the link masters on the link.

Wiring Blocks Manually

To wire blocks manually, complete the following steps:

1. Click the **Wiring** tool on the Function Block Application Editor toolbar.

2. Click 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 in the background of the Function Block Application Editor window.

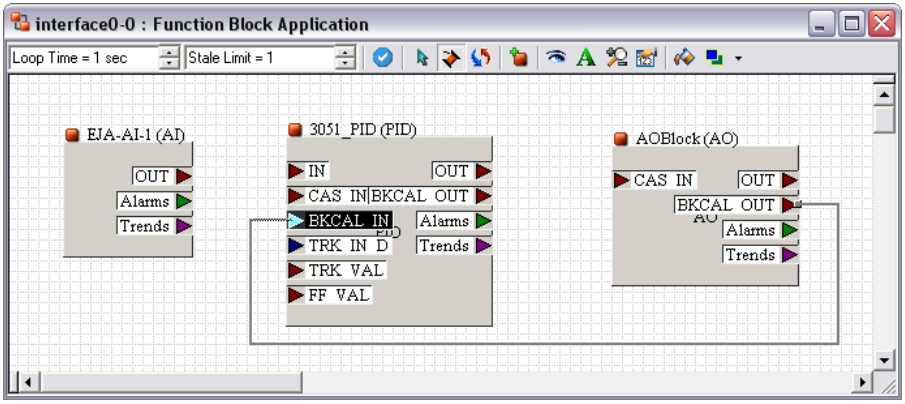
Tip To toggle the direction of the connection you are dragging, press the spacebar while wiring.

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

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

When you click the Wiring tool, the NI-FBUS Configurator performs type checking and highlights valid inputs and outputs when you move the cursor over inputs and outputs. For instance, the NI-FBUS Configurator does 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.



Checking for Function Block Application Errors

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

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

Click 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. You can drop a template into the Function Block Application Editor window as a model for your function block application. Templates provide the wiring for common control strategies, such as PID loops. You also can 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

You can use templates from the NI-FBUS software for wiring only. The function block settings in your application override values in templates from the NI-FBUS software. You can choose to apply the values in user-defined templates. When you use a template from other users, do not accept the template values. The parameter settings may not apply to your devices.

Inserting Templates

To access templates, do one of the following:

- Right-click 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 parameter values to the function block that will eventually replace the template. You can choose one of four different options:

- Click the **No** button to keep the function block's current values. Click this button to ignore any values stored in the template. If you cannot ensure that the parameter setting applies to your devices, ignore these values.
- Uncheck the **Use *Template* CHANNEL values?** checkbox and click the **Yes, use values** button to change the parameter values of the function blocks used to fill the template to match the template parameters, except the `CHANNEL` parameter. Click this button 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.

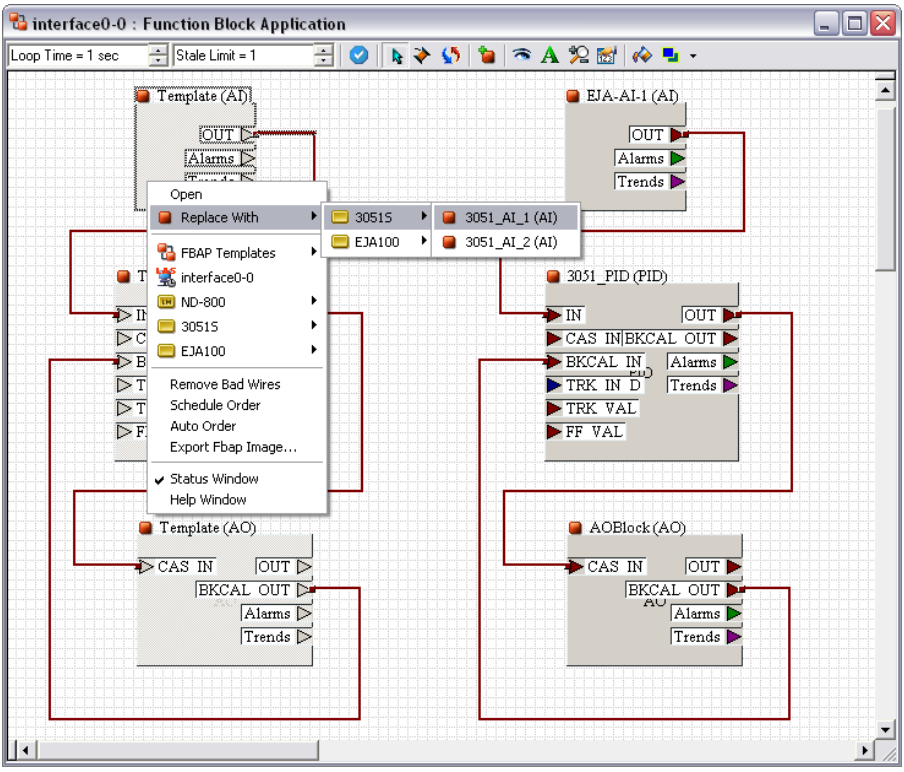
  The channel is device input or output channel for controlling. Refer to the *Creating Templates* section for more information on creating templates.
- Check the **Use *Template* CHANNEL values?** checkbox and choose **Yes, use values** to change the function block parameter values to match all the template parameters, including the `CHANNEL` parameters.
- Click the **Decide Later** button to perform no actions, 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 parameter values by right-clicking the template block outline and choosing **Open**, or by double-clicking the block.

While the Function Block Application Editor window highlights the template, you can drag the template to the desired location.

Tip Moving the mouse just outside of the Function Block Application Editor window while dragging blocks scrolls this window.

The following figure shows an example of two templates. The template on the left illustrates the appearance of a template when you place the template. The template on the right illustrates the appearance of a template when you fill the template.



When you place templates, the template blocks are grayed out. You need to assign a function block to the template block to use the template block. To assign a function block to the template block, complete the following steps:

1. Double-click 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. To create a template, complete the following steps:

1. (Optional) Select a part of the function block application that you want to save if you only need part of the application. You can select part of the function block by dragging the pointer around the desired blocks and wires or using the <Ctrl> key while clicking the desired blocks and wires.

2. With the Function Block Application Editor window open, select **File»Save Template**.
3. 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, this template automatically appears as an available template in the template menu.

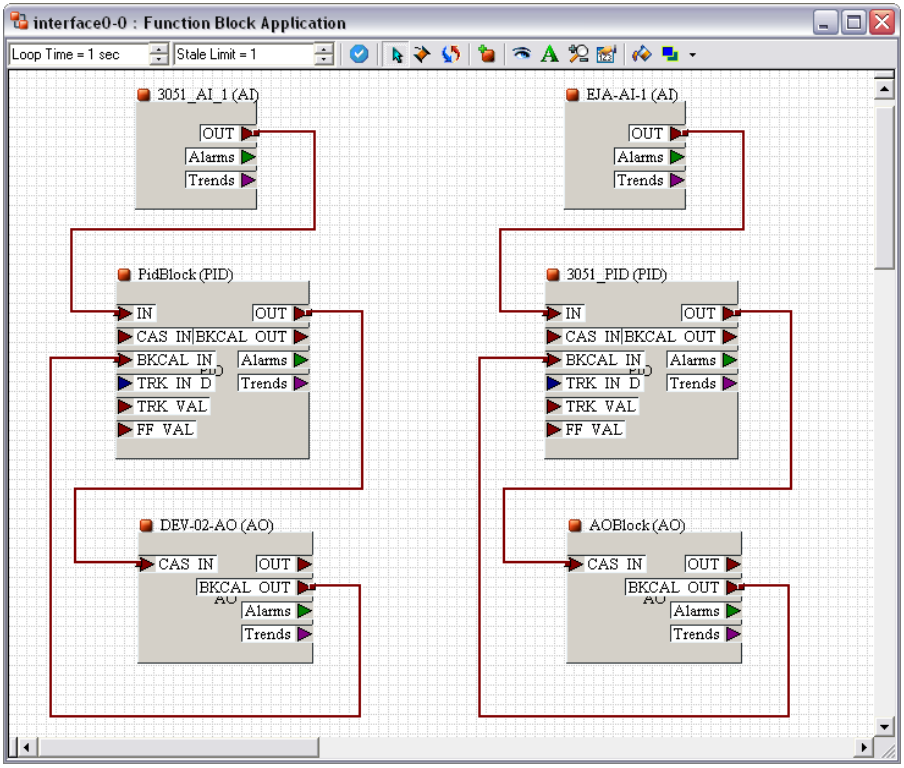
Export Function Block Application Image

You can export the image of configured function block application as a bitmap file.With the Function Block Application Editor window open, select **File»Export Fbap Image**.

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.

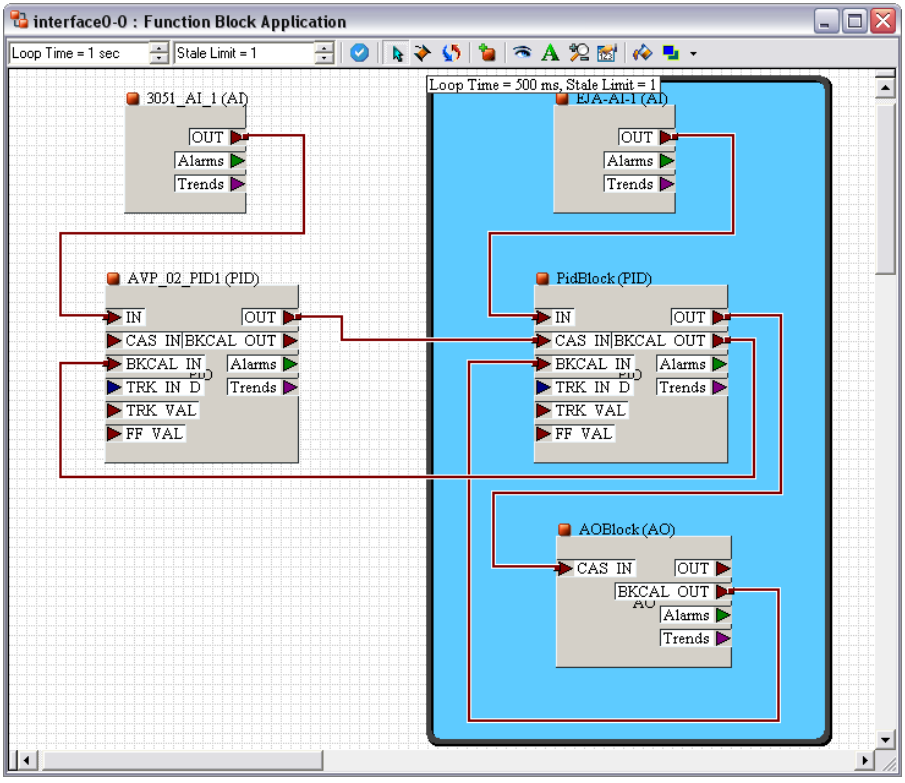
The following figure shows an example of two separate loops running at the same rate. You can add loops running at the same rate to a Function Block Application without using a loop structure.



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.



To create a loop structure, complete the following steps:

1. Click the **Loop** tool on the Function Block Application Editor toolbar.

2. Click 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 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, you must balance between PID loop numbers and the execution rate. The device manufacturer specifies the maximum execution time, or the worst-case execution time, for each block.

To view the maximum execution time of a block, complete the following steps:

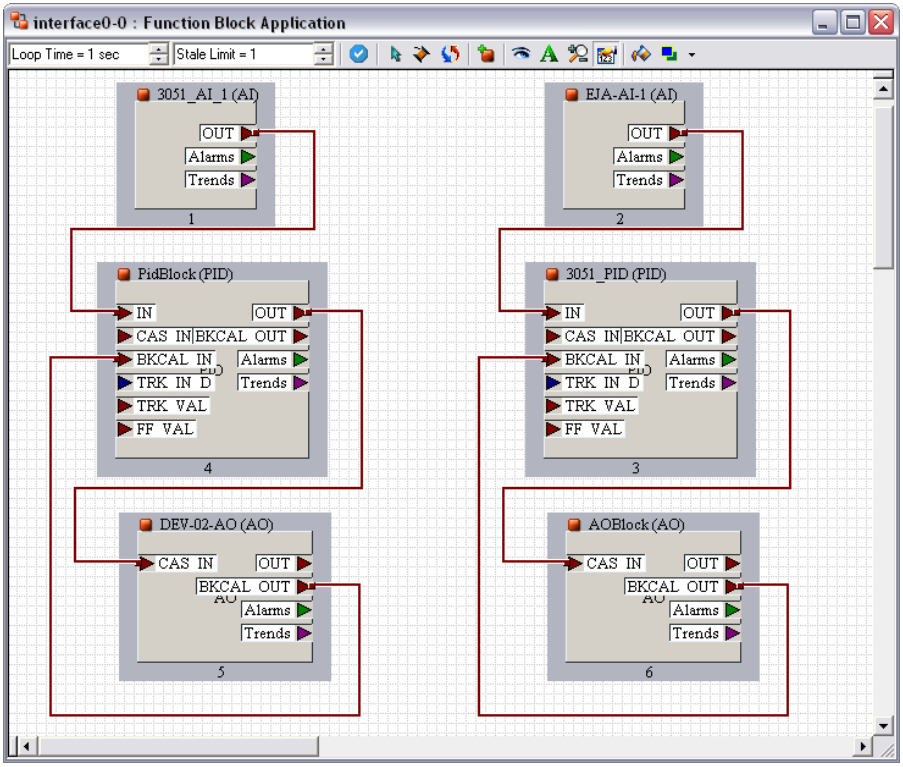
1. Select **View»Preferences** and click the **Block View** tab.
2. Select the **Show Block Information** checkbox, and click **OK**. Now when you double-click on a function block, a new tab called **Block Information** will appear.

Check the parameter `EXECUTION_TIME` on the **Block Information** tab for the function block. If the number is a hexadecimal number, you can change the number 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. You can divide this number by 32 to get the maximum execution time, in milliseconds. For example, if the maximum execution time for a block is 192, dividing this number 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:

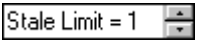
1. In the Function Block Application Editor, click the **Execution Order** button. Numbers appear on the blocks, indicating their current order, as shown in the following figure.



2. Click the blocks in the order that you want these blocks to execute.
3. Click in the background of the Function Block Application Editor or select another tool. The schedule changes accordingly. For more information, refer to the *View and Edit a Schedule* section.

Setting the Stale Limit

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



Using Menus and Methods

The NI-FBUS Configurator supports user methods. These methods 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, such as the “Set up device” method or “Calibrate valve” method. These methods represent specific tasks you want to perform.

If you double-click a function block in the configuration tree and the block has menus and methods associated with the block, 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 appear to the right of this menu. The device description defines the additional menus. These menus contain parameter names or method names. You can invoke the names by selecting these names. The NI-FBUS Configurator also shows the **Methods** menu that contains an entry for each method.

If you select a parameter name from the menu, an edit window for that parameter appears. To invoke a method, select the method from the menu bar. The block configuration window switches to the **Methods** tab. This tab displays all output from the method. When the method requires you to enter input, enter the input in a box in the **Methods** tab. Methods execute until the methods encounter an error or complete. The **Methods** tab also displays the error information. You cannot close the block configuration window while a method is running.

Using EDDL Extensions

Electronic Device Description Language (EDDL) extensions improve the data organization, graphical visualization, and persistent data storage for a field device through the use of dialog boxes, charts, graphs, images, dynamic variables, and so on. The robust graphical interface, which may incorporate cursor, pan, and zoom functionality to inspect the details, provides a comfortable and convenient way to make operational decisions.

Complete the following steps to access the EDDL elements and functions of a device.

1. Add the EDDL device in the online or offline mode.
2. Open the **Block** dialog box of your selected block. You can access the EDDL elements through menu items.

Note The NI-FBUS Configurator may take a few minutes or longer to open an EDDL window, depending on the number of elements in the window.

The following table shows the menu item types supported by the **Block** dialog box and the action performed for each.

Table 2-1. Block Dialog Menu Items

Item Type	Icon	Action
Menu		To expand sub-items or open an EDDL dialog/window (depending on its style).
Method		Execute a defined action, showing the operation process and result in the **Method** tab of the **Block** dialog.
Chart		Activate a chart and display real-time (continuous) data values read from the device in the chart.
Graph		Activate a graph and display the static data stored in the device.
Image		Display a graphical image in a pop-up dialog.
File		Describe an area of persistent storage.
Edit Display		Group data items together during editing in a pop-up dialog.
Array & Record		Show a logical group of variables or values in a pop-up dialog.
Variable		Show selected items (variables, unit and write-as-one relations, item arrays, collections, and programs) in a pop-up dialog.
Other		Show selected items (refresh relations, variable lists, response codes, and domains) in a pop-up dialog.

An EDDL dialog/window box supports the following item types.

Table 2-2. Item Types Supported by an EDDL Window/Dialog Box

Item Type	Note
Page	Shown as a tab page with all sub-items in it.
Group	**(Page or Group's sub item)** Shown as a group box with all sub-items in it.
Chart	**(Page or Group's sub item)** Displays real-time (continuous) data values from the device. It may be displayed as a time-based chart, a gauge, a horizontal bar, or a vertical bar.

Table 2-2. Item Types Supported by an EDDL Window/Dialog Box (Continued)

Item Type	Note
Graph	**(Page or Group's sub item)** Displays the static data stored in the device.
Image	**(Page or Group's sub item)** Displays a graphical image.
Button	**(Page or Group's sub item)** Defines an action to be executed. It could be a nested dialog/window or a method.
Variable	**(Page or Group's sub item)** Displays the variable or the element(s) of a record, containing two parts: the variable's label and its value.
List	**(Page or Group's sub item)** Displays the Edit Display, Array, and other variable-related structures.
Grid	**(Page or Group's sub item)** Displays a horizontal or vertical grid.

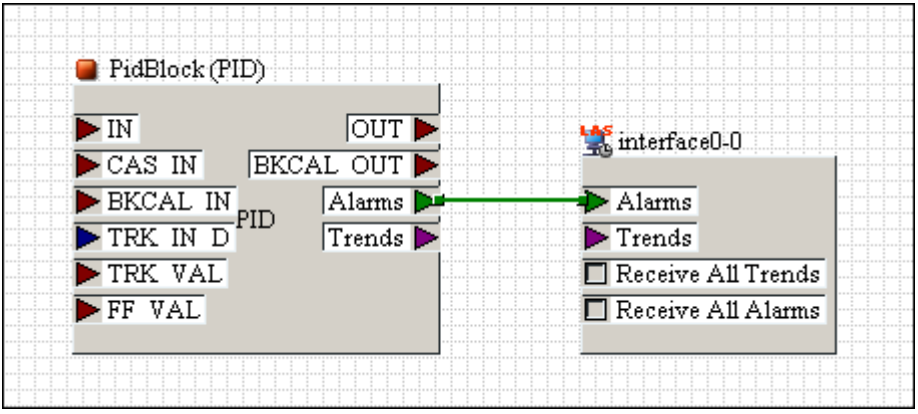
Note An EDDL window *cannot* apply parameter changes directly. Use the **Block** dialog to make and save changes for the parameters.

Configuring Alarms

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

Complete the following steps to configure alarms:

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 window.
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** checkbox. The following figure shows an example of a configured alarm.



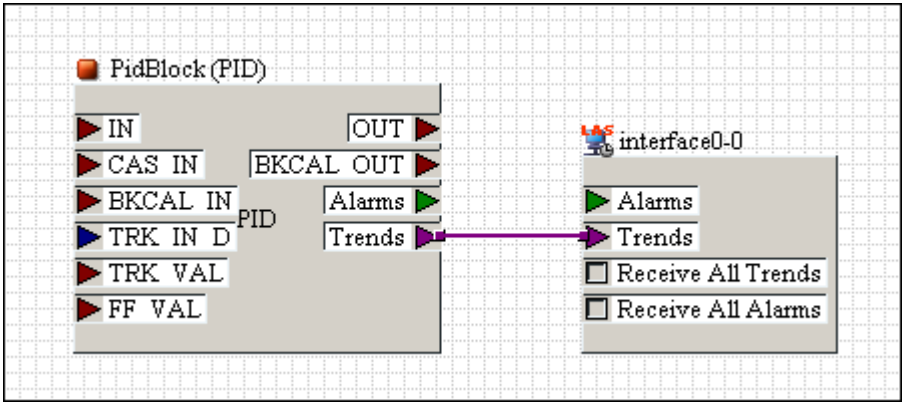
3. Double-click the function block where you want to generate the alarm. The block configuration window appears.
4. In the block configuration window, click the **Alarms** tab to view or change alarm information, such as the high and low limits of an alarm condition, alarm priorities, and so on.

Configuring Trends

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

Complete the following steps to configure trends:

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 window.
2. Either wire the **Trends** output of the block to the device you want to receive the trend, or check the **Receive All Trends** checkbox. The following figure shows an example of a configured trend.



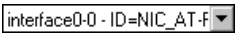
3. Double-click the function block from which you want to send a trend. The block configuration window appears.
4. In the block configuration window, click the **Trends** tab.
5. Check the **Trend** column checkbox next to the parameter you want to trend.
6. Set the sample **Interval** of the trend in loops/sample. The interval is a multiple of any loop macrocycle time of this block. For example, if the interval is 2 loops/sample, and the loop macrocycle time is 1 s, the trend acquires a value every 2 s.
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 trend-viewing software installed. 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.

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.

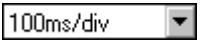


Change the Link Active Schedule

To change the schedule, complete the following steps:

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

2. Click 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 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 the **Errors** button in the schedule window.

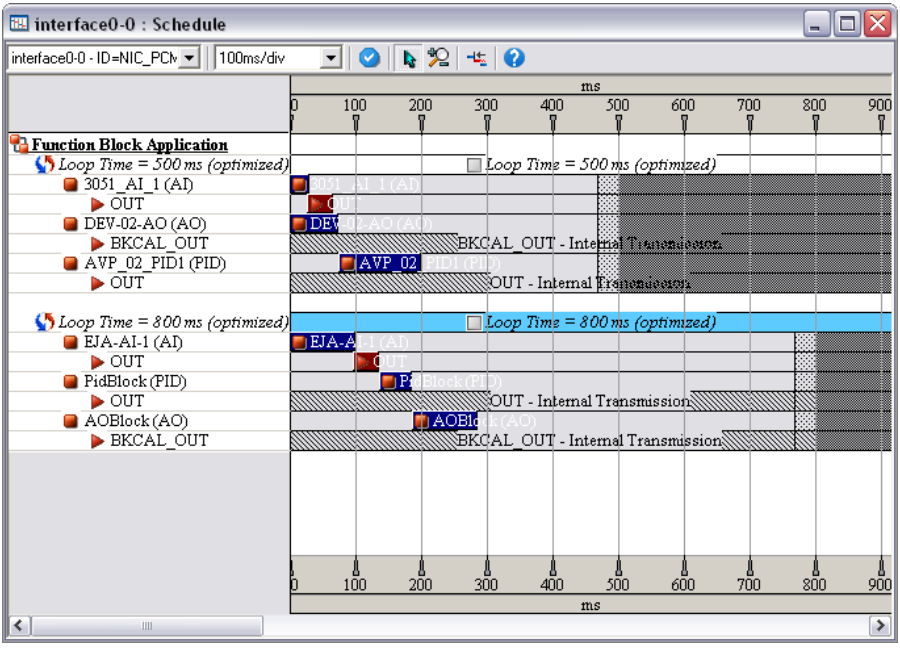
Tip To zoom in and out, click the magnifying glass icon in the schedule window.

Tip With only one device, you cannot see data transmission because this device does not transmit any parameters to other devices. For some fast devices the blue blocks representing execution times are so short that the blocks are 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.



White dividers separate the loops. If you want to lock the current schedule for a loop, check the **Loop Time** checkbox above the colored bars. The NI-FBUS Configurator cannot changing the schedule until you uncheck the checkbox.

Set Network Parameters

To change network parameters such as the Link Active Scheduler (LAS) and primary time master, double-click 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 LAS when you

download configuration. If the primary LAS is unavailable, one of these devices steps in and continues executing the LAS. This behavior provides redundancy for FOUNDATION Fieldbus systems.

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

Caution You cannot set network parameters in the HSE segment. Do *not* modify advanced parameters without good reasons. If you must modify parameters for certain devices, use the recommended settings from the device manufacturer. 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 Network Parameters. Click the Link Settings tab. Press the **Press to See Suggested Values** pushbutton. The next window shows the original settings.

Change Language

Some device manufacturers allow you to change device names and help strings to multiple languages.

Complete the following steps to change 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**. The NI-FBUS Configurator saves the entire project, including the configuration tree, Function Block Application, and schedule.

Download a Project Configuration

Besides parameters, you must download other configuration to devices before executing function blocks on devices. For example, you must download function block linkages to allow dataflow from one block to another. Also, devices need schedules for function block execution and communication. 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

Complete the following steps to download your project configuration.

1. Click 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. Click the **Link Masters** button to set link masters to receive the primary LAS, the primary time master of the link, and the LAS. 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 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. This window shows the current device name, current device settings, and configure result.
6. If you did not check the **Automatic Mode Handling** checkbox, click the **Set All Targets to OOS** button to switch the function blocks to the `OOS` mode. After the configuration downloads, you must set the blocks back to the normal mode of operation to begin operating.

Download Options

The download configuration dialog box contains multiple options for downloading a configuration.

Write Contained Block Parameters

Write Contained Block Parameters writes all contained parameters from the block configuration windows. Contained parameters are internal to the function block, such as the scaling and alarm limit settings. Use this option when you change parameters in multiple function blocks and do not want to write parameters block by block.

Note Per the FOUNDATION Fieldbus specification, some contained parameters have default values that are not legal values. You can change these parameters to values appropriate for your processes. Device manufacturers often allow devices to operate with uninitialized parameter values. However, any attempts to rewrite the uninitialized value to a parameter result in an error. Use **Write Contained Block Parameters** to generate errors if any parameters are uninitialized. Clear these errors by writing a legal value to the parameter or by reading the value into the NI-FBUS Configurator from the device. Generally, these errors are non-fatal and the download can continue successfully.

Clear Devices

Clear Devices clears the existing configuration in the devices before downloading. Using this option 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 to the OOS mode as needed during the download process and returns the blocks to the normal modes of operations (as read from MODE_BLK.NORMAL for each block) after the configuration is downloaded. If you do not use this option, you need to manually change blocks to the OOS mode and set the blocks back to the NORMAL mode 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 can change any of the modes from this table, or to set all blocks to the OOS mode by using the **Set All Targets to OOS** button. The table appears again at the end of configuration, and allows you to set the modes manually, or click the **Set All to Normal** button to set all the blocks into the normal modes. You must click the **Continue** button to write a mode to the bus.

Verify and Diff

This option enables the NI-FBUS Configurator to upload the actual configuration from the devices after a download. The NI-FBUS Configurator compares the uploaded information to the configuration file. Refer to the *Compare Two Configurations* section for instructions. The NI-FBUS Configurator reports differences between the configuration file and the actual configuration of the devices. Use this option when the configuration on the bus is unknown or when you need to verify whether devices accept all assignments.

Bringing a Block Online

If you do not select **Automatic Mode Handling** during the download process, you must manually change the function blocks back to the 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 value should become Auto. If field value is incorrect, examine the parameters on the **Diagnostics** tab to identify the problem.
2. Go to the MODE_BLK parameter of a function block and set the TARGET to Auto.
3. Re-read the MODE_BLK parameter. The ACTUAL field should become Auto. If the filed value is incorrect, examine the parameters on the **Diagnostics** tab to identify the problem, or refer to your device documentation.

Once the block goes to the NORMAL mode, the block starts to execute 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 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 the **Start Monitoring** button to monitor the application.

The NI-FBUS Configurator outlines the mode box for a block 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 the desired object in the configuration tree and select **Simple Verify Configuration**.
2. Click 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.

Caution Clicking **Upload Changes** button modifies the currently open project to match the devices. You cannot revert this modification.

3. If the object settings in the device do not match the project settings, you can click the **Upload Changes** button to upload the actual device settings to the project.
4. Click the **Done** button.

Compare Two Configurations

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

You also can compare two different configurations. Right-click the link and select **Compare Configurations**.

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

With verifying and diffing, 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. If the NI-FBUS Configurator encounters any errors while the NI-FBUS Configurator is creating the report. The report returns these errors first. The related portions of the difference report may be incomplete due to the errors. A standalone window displays the report as a tree with branches. Expand the branches by clicking the “+” sign. Each line of the report may have different color. Yellow represents new items. New items are items that appear on the bus but not in the current project. Blue represents missing items. Missing items are 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 the **Check Project for Errors** button in the configuration tree to check for errors in the project.

Add An Additional Function Block Application

Click 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 in the configuration tree and select **Set Tag**.

Replace Devices

You can transfer device settings, such as parameters, tags, and linkages, to another device. You can transfer device settings regardless whether the original device is functioning or not. Complete the following steps to transfer device settings to a replacement field device.

1. Connect the new device to bus.
2. From the **Configure** menu, select **Replace Devices** or right-click the device to be replaced and select **Replace Devices**. The original device you selected appears 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 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 settings, including linkages, tags, and parameter values, overwrite the settings of the replacement device.

Reset a Device to Factory Defaults

You can clear a field device to reset the device to a state appropriate for shipping. To reset a device, right-click 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. The NI-FBUS Configurator also clears all communication information in the device.

Change Device Type

You can change the type of a device between basic device and master device. To change the type of a device, right-click the device name in the configuration tree and select **Change Device Type** from the menu that appears.

Note This feature only takes effect on the H1 online device.

Export & Import Parameters

You can batch-edit parameters by using the **Export & Import Parameters** feature. This feature allows you to export all object parameters to a file, such as link, device, block, and so on. This feature also allows you to import and change all updated parameter values.

To batch-edit parameters using the **Export & Import Parameters** feature, complete the following steps:

1. Create a new project or open an existing project. Wait until all devices in the project are in the normal state.
2. Right-click an object with the parameters you want to export in the **Configuration Tree**, and select **Export Parameters**.
3. Check the following options according to your application.
   - **Export Trends**—Click this checkbox to export the trends.
   - **Export MIB**—Click this checkbox to export the MIB parameters.
   - **Export Readonly Parameters**—Click this checkbox to export the read-only parameters.
   - **Export Detailed Information**—Click this checkbox to export detailed information including data type, help, and so on.
   - **Customize Parameters**—Click this checkbox to select a custom set of parameters you want to export.
4. Enter the correct file name in the **Save as** dialog box. Click the **Save** button to export parameters as a `csv` file.
5. Open the exported file using a spreadsheet program to edit the value column.
6. Right-click the object with the parameters you want to import in the **Configuration Tree**, and select **Import Parameters**.
7. Check the following options according to your application.
   - **Import Trends**—Click this checkbox to import the trends.
   - **Import MIB**—Click this checkbox to import the MIB parameters.
   - **Write Changes**—Click this checkbox to write the changes after import.
   - **Customize Parameters**—Click this checkbox to select a custom set of parameters you want to import.
8. Enter the correct file name in the **Open** dialog box. Click the **Open** button to import parameters from a `csv` file.

Note Change only the **value field** in the `csv` file. Changing any other field may damage the `csv` file. To avoid a comma in a field being considered as the delimited character, replace the comma with an asterisk.

Set Preferences

To set your preferences in the NI-FBUS Configurator, select **View»Preferences**. The **Preferences** dialog box appears. This dialog box contains the following tabs:

- **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 this checkbox to add space between parameters.
  - **Show Detailed View**—Click this checkbox to show additional type and help strings for the view.
  - **Show Block Information**—Click 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 this checkbox to show descriptive information about parameters in the block configuration window.
  - **Use the Dynamic Attributes of Parameters Dynamically**—Click this checkbox to enable the refresh of dynamic attributes, which is defined in DD provided by the device manufacturer.
  - **Enable the Scaling Factor on the Arithmetical Parameters**—Click this checkbox to enable the scaling factor on the arithmetic parameters, which is defined in DD.
  - **Use the Display-format and Edit-format of Parameters**—Click this checkbox to enable the different formats for display of parameters defined in DD.
- **General**—Use this tab to select which windows appear when you start the NI-FBUS Configurator. However, the **Automatically Set the pdTag and Node-address** option is used to determine whether NI-FBUS Configurator automatically sets the tag or address of the factory default device. Refer to the *Set Device Addresses* section of this chapter for more information.
- **Links**—Use this tab to change the software key, as described in the *Changing the Software Key* section.
- **Log Settings**—Use this tab to set the information log settings, as described in the *Log Information* section;
  - **Status Log File**—Input or select the file name to store status information.
  - **Download Log File**—Input or select the file name to store download information.
  - **Assignment Log File**—Input or select the file name to store assignment information.
  - **When to Log**—Click to log in real time. Click **Quit Program** to log only before NI-FBUS Configurator quits.
  - **Add Timestamp on Each Item**—Check this option to append timestamp at the end of each information item.
  - **Timestamp Format**—The options define the format to display the timestamp.

- **Maximum Size of The File**—The size limit of each log file.
- **Deny other process to modify the files**—Check this option to prevent other processes or programs from modifying the log file.

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 want 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.

You can find the current software key under **Help»About FCS**. You 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 the **Links** tab.
3. On the **Links** tab, click the **Show About Dialog** button. The **About** dialog box appears.
4. Click the **Set SW Key** button, enter the new key, and click the **OK** button. Click 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 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**—Allows you to 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**—Allows you to select printing styles for the graphical schematics, page titles, and numbering.

Log Information

To log the status, download, and assign information in the status window, complete the following steps:

1. Select the main menu **View»Preferences** to launch the **Log Settings** dialog box.
2. Refer to the *Set Preferences* section for information on the settings.
3. Click **Start Log**.

To stop the log, select **View»Preferences** to launch the **Log Settings** dialog box, and click **Stop Log**. Click **Reset Log** to empty the log files.

3

NI-FBUS Configurator Tutorials

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

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

These tutorials provide instructions about using the following types of control applications:

- Basic connections to AI and AO signals
- PID control
- Offline configuration
- Project modification

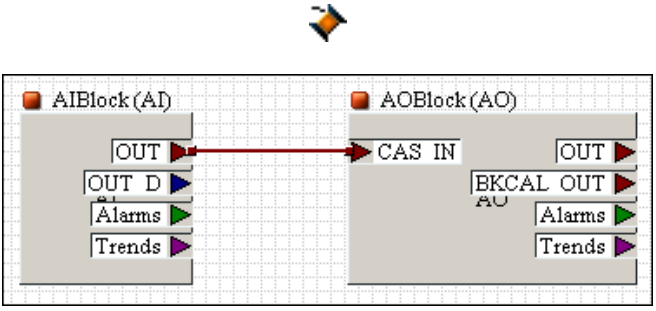
Tutorial 1: Connect to Network and Hardware

1. Launch the NI-FBUS Configurator.
2. In the **Startup** dialog box that appears, choose **Create New Project**, select **Interface0-0**, and click **OK**.
3. If **Interface0-0** is a local link and you did not launch the local NI-FBUS Communications Manager, a dialog box appears asking if you want to launch this application. Select **Yes**.

The NI-FBUS Configurator automatically detects all local and remote valid devices connected to the network. If a device is registered by the FIELDBUS FOUNDATION as a compliant device and the device description files have been imported, the NI-FBUS Configurator detects both the device and the function blocks in the device. This process takes a few minutes. The configuration tree displays all local and remote devices and function blocks connected to the link.

Tutorial 2: Create Basic Connections to AI and AO Blocks

1. Double-click Function Block Application in the configuration tree. A blank window appears to the right of the configuration tree. This window is the Function Block Application Editor window.
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 another function block. Connect the AI and AO modules together. Use the pointer tool to organize the blocks or nudge wires. The **Zoom** menu has several options to make the code within the Function Block Application Editor more manageable, as shown in the following diagram:



4. Download the configuration so that the function blocks can begin executing. However, if the device controls a real-world signal, configure the parameters appropriately before downloading to avoid unexpected output values. The function blocks do not go to the `Auto` mode until you schedule the function blocks to execute. You can schedule function blocks by placing the function blocks in the Function Block Application Editor window and downloading. If you want to download the configuration now, proceed to step 8.
5. Adjust the scaling. For the AI block, you can set three types of scaling: `XD_SCALE`, `OUT_SCALE`, and `L_TYPE`. For more information about scaling, refer to the *Scaling Parameters* section of Chapter 4, *Parameters*.
6. To change the scaling parameters, double-click the AI function block.
7. In the block configuration window, select the **Scaling** tab.
8. Confirm the scaling parameters and check that the other parameters are reasonable.

9. To change a parameter, click the current value. Then, enter the new value for your device, as shown in the following example.

```
XD_SCALE
    EU_10050
    EU_0-50
    Units_IndexmV
    Decimal3
OUT_SCALE
    EU_100400
    EU_0-100
    Units_Index°C
    Decimal3
L_TYPEindirect
```

Note If you select an L_TYPE of direct, the NI-FBUS Configurator stops scaling the transducer value by ignoring the OUT_SCALE parameter. Normally, you should select one of the indirect L_TYPES. Refer to the *L_TYPE (Scaling)* parameter description of Chapter 4, *Parameters*, for more information.

10. Click the **Write Changes** button at the bottom of the window to write all the new values.

    If the block contains previous configuration information, you may receive a warning that the block is in the wrong mode for the request. The following two reasons may cause this warning:

    - The device accepts writing request to this parameter only in the OOS mode. Acknowledge the warning and click the **OOS** button to switch the device to the OOS mode before clicking **Write Changes** again. Change the block back to the original mode after you write the new values.
    - The NI-FBUS Configurator tries to write an invalid value to a parameter. Devices refuse invalid values. You must identify the invalid parameter value and use a valid value instead.

11. Modify the scaling of the AO module. Double-click 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. You can see these values on the bus.

```
PV_SCALE
    EU_100400
    EU_0-100
    Units_Index°C
    Decimal3
XD_SCALE
    EU_10050
    EU_0-50
    Units_IndexmV
    Decimal3
L_TYPEindirect
```

12. Adjust the macrocycle of your configuration. The macrocycle is the least common multiple of all the loop times on a link. To edit the schedule, double-click the **Schedule** icon from the configuration tree. If you use a fast device, you can change the Loop Time to 0.1 second for this configuration. If the macrocycle is too low, a warning appears on the **Errors** tab.
13. Download your configuration by clicking the **Download Project** button.

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

Click the **Download** button 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. Click 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 **Start Monitoring**.

    You also can 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 completed Tutorial 2, which describes how to modify function blocks, write parameters, and test a configuration.

1. If your device supports function block instantiation, create a PID function block.
    a. Right-click the device in the configuration tree and select **Instantiate New Block**.
    b. Select **PID Block** and click **OK**.
    c. Right-click 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 by 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. Click the **No** button because the function block parameters are more suitable than parameters in this generic template. This operation places a PID template on the Function Block Application Editor window. Notice the blocks are grayed out because the blocks are merely shells.

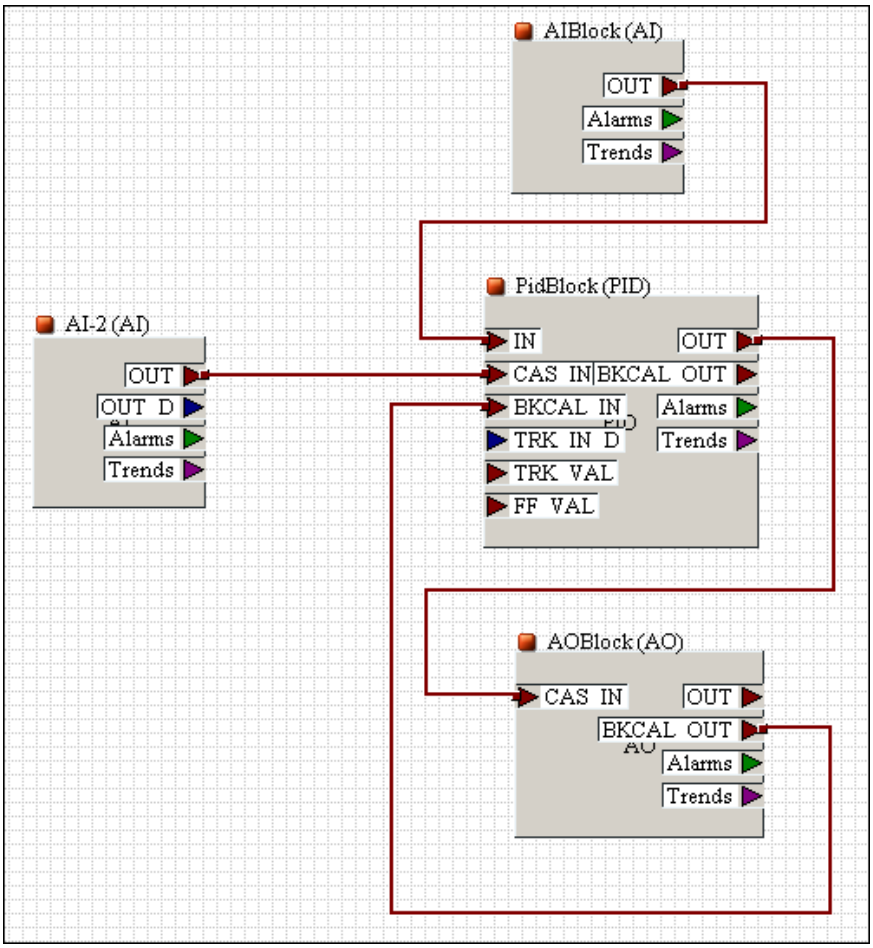
4. Assign function blocks to fill in the template blocks.
   a. Right-click an AI template block, highlight **Replace With**, and select an AI function block. This block provides the process value to the PID block.
   b. Right-click the AO template block, highlight **Replace With**, and select an AO function block. This block is the output that the PID block controls.
   c. Right-click 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, as shown in the following example:

   ```
   GAIN10
   RESET3.5 sec
   BAL_TIME0 sec
   ```

6. Set the PID scaling parameters. The PV_SCALE of the PID block should match the OUT_SCALE of the AI block. The PV_SCALE of the PID block defines the range of the IN parameter of the PID. The OUT_SCALE of the AI block defines the range of the value that the AI passes to the PID. Likewise, the OUT_SCALE of the PID block should match the PV_SCALE of the AO block. The OUT_SCALE of the PID block defines the range of the value that the PID passes to the AO. The PV_SCALE of the AO block defines the range of the CAS_IN parameter of the AO.

   ```
   PV_SCALE
       EU_100400
       EU_0-100
       Units_Index°C
       Decimal3
   OUT_SCALE
       EU_100400
       EU_0-100
       Units_Index°C
       Decimal3
   ```

7. (Optional) Use a second AI block to specify the setpoint of the PID.
   a. Drag another AI block onto the 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. An example of how the final Function Block Application Editor window should look is shown in the following figure.



8. Decrease the macrocycle 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 as follows.

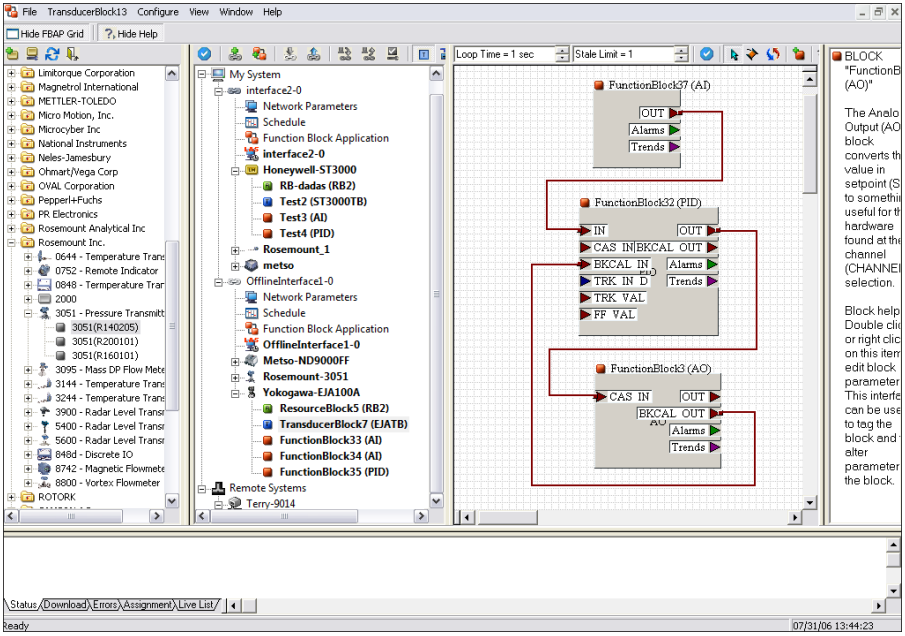
   - First AI—`Auto`
   - Second AI—`Auto`

- PID—`Cas`
- AO—`Cas`

Tutorial 4: Create Offline Project

1. Launch the Interface Configuration Utility.
2. Click the **Add Interface** button in the dialog box showing the currently configured interfaces.
3. In the **Choose Interface Type** dialog box, choose the **offline** type, then click the **Continue** button,
4. Select the number of ports (interfaces) to add and click **OK**.
5. Make configurations for each port, and click **OK**.
6. Close the Interface Configuration Utility, and launch the NI-FBUS Communications Manager.
7. Launch the NI-FBUS Configurator again.
8. In the **Create New Project** field, choose **Offline Project**, select an offline interface link, and click **OK** to create the new offline project.

   The Device Type Tree appears in the left window. This tree lists the devices in the hierarchy tree and groups the devices according to the vendor name.
9. Drag the offline device you want to add into the offline link from the Device Type Tree and drop the device onto the offline link in the project window.
10. The **Add Device** dialog appears informing you to set network address and device tag. You can enter the address and tag, or use the default address and tag. The NI-FBUS Configurator checks the default address and tag values to ensure no conflict with other devices. Click **OK** to add the new offline device.
11. After you add all offline devices to the project, double-click **Function Block Application** under the offline link to create function block connections and function block schedules.



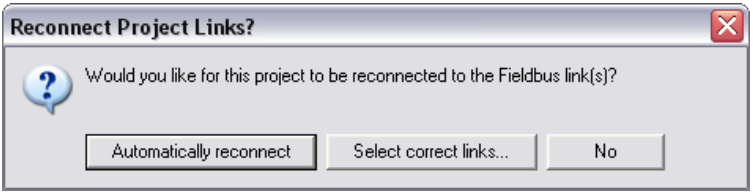
12. After configuring your function block application, select **File»Save As** to save your offline project in a `.fcp` file.
13. Close the NI-FBUS Configurator.

Tutorial 5: Modify Saved Project

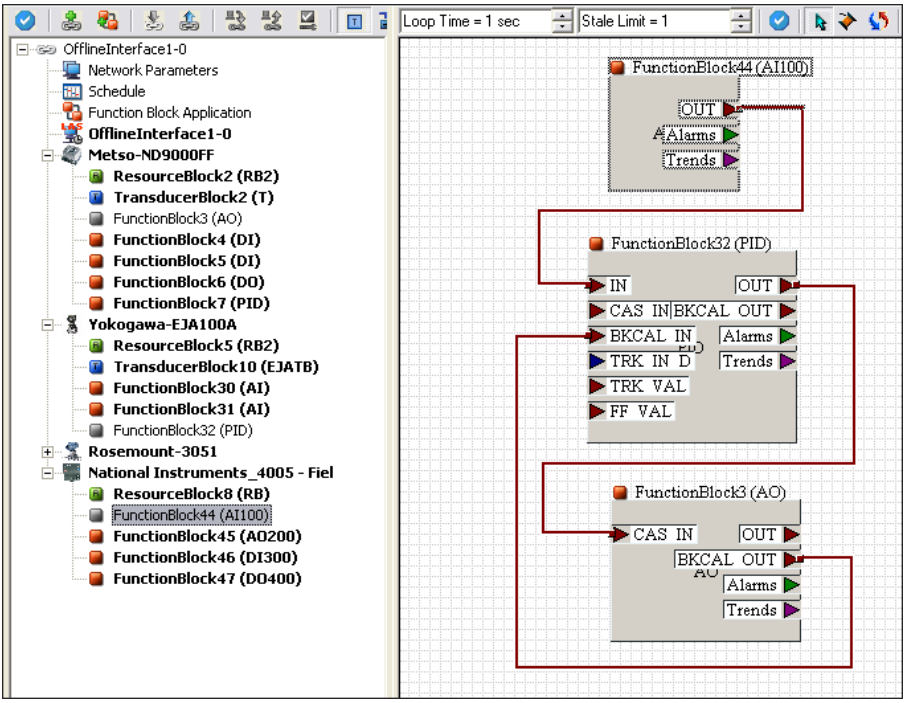
You can modify an exiting saved project file in the NI-FBUS Configurator. Before you make changes to the saved project, you must add an offline interface first.

1. Launch the Interface Configuration Utility. Refer to *Tutorial 4: Create Offline Project* to add offline interfaces.
2. Launch the NI-FBUS Communications Manager.
3. Launch the NI-FBUS Configurator.
4. Choose **Open an Existing (*.fcp)**, then double-click the file in the list, or select **More Files…** and click **OK** to choose an existing saved configuration project from the **Open** dialog box.

5. The **Reconnect Project Links** dialog box appears when you open the configuration project file. Click **No**.



6. Right-click the link you want to modify and select **Take Offline**. If a dialog box appears, select **Yes**. Then choose an offline link from the list in the **Take Link Offline** dialog box and click **OK**.
7. The original link will connect to an offline link, and all the devices and configurations will be created in this offline link as well.



Modify the project as your new design: add/remove device, change function block connection, adjust schedule or modify parameter value.

8. After modification, select **File»Save** to save the new project file.

Tip Refer to the *Take Link Online* section of Chapter 2, *Using the NI-FBUS Configurator*, to learn how to apply the new configuration to the field network.

4

Parameters

This chapter contains 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 allow you to 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. These parameters also provide general information about the device and device 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. These blocks use the faultstate when the control of the output block is lost while the block is in the Cascade mode. If the block is in the 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 the output value. If the option is not set, the block holds the value it receives at the time the input goes bad until the cascade can be re-established.

The faultstate is a predefined state that outputs go to under certain conditions. For example, if the valve 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	—

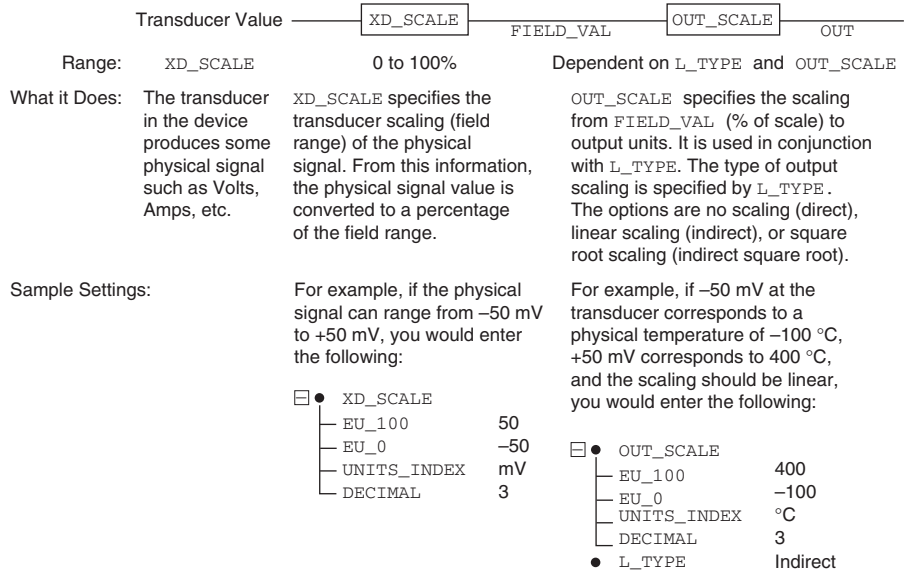
Scaling Parameters

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

- 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 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.

Figure 4-1. AI Scaling Procedure



Calculations:

$$FIELD_VAL = \frac{(transducer_value - XD_SCALE.EU0)}{XD_SCALE.EU100 - XD_SCALE.EU0}$$

$$OUT_{Direct} = transducer_value$$

$$OUT_{Indirect} = \frac{FIELD_VAL}{100} * (OUT_SCALE.EU100 - OUT_SCALE.EU0) + OUT_SCALE.EU0$$

$$OUT_{Indirect\ Square\ Root} = \sqrt{\frac{FIELD_VAL}{100}} * (OUT_SCALE.EU100 - OUT_SCALE.EU0) + OUT_SCALE.EU0$$

Figure 4-2. AO Scaling Procedure

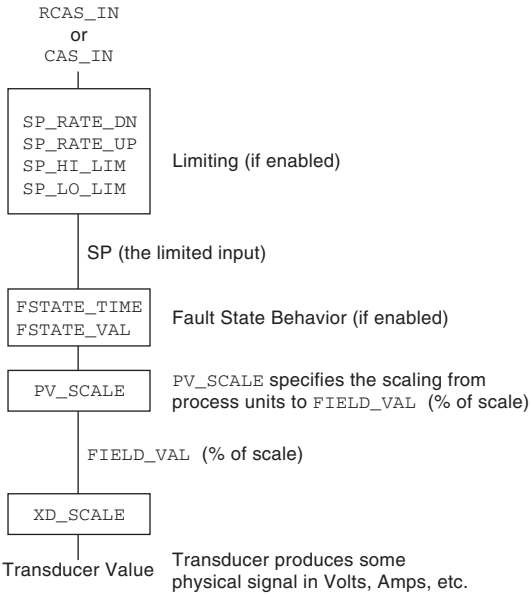


Figure 4-3. PID Scaling Procedure

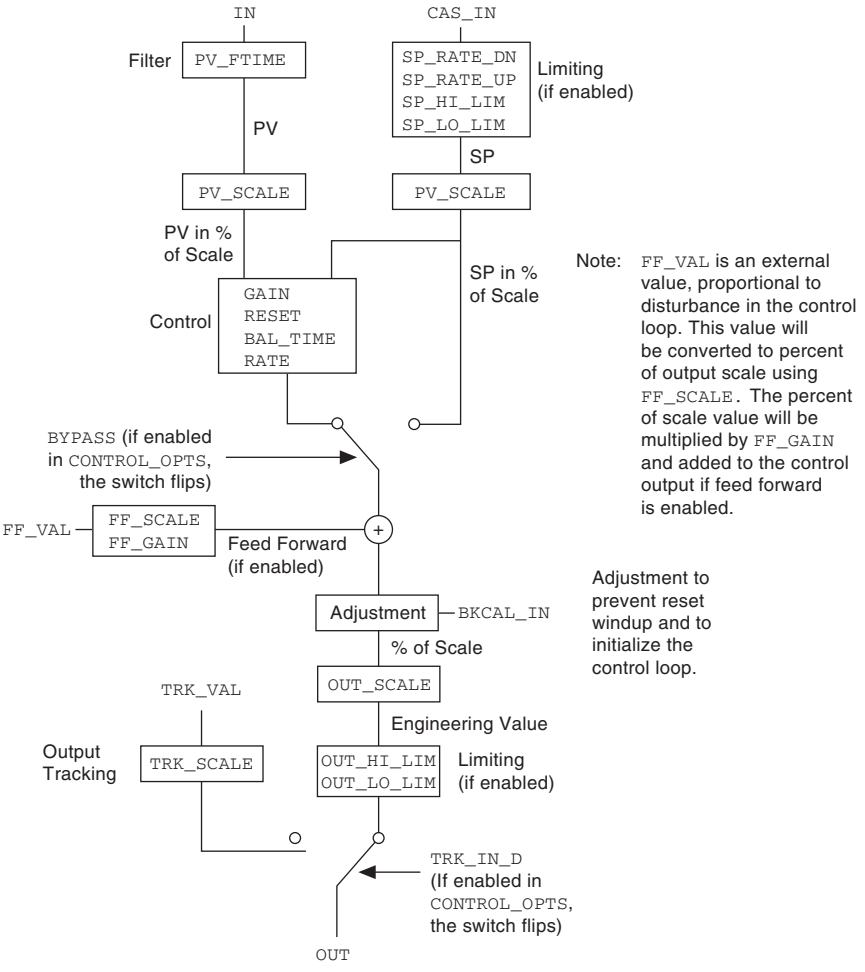


Table 4-4. Scaling Parameters

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 as follows.

CHANNEL
HARD_TYPES
IO_OPTS
LOW_CUT

Limiting Parameters

Limiting is to prevent a calculated value, such as a value from a PID block, 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 a host application periodically writes the cascade input, FOUNDATION Fieldbus provides mode shedding to handle failure of the host application. For the RCas mode and ROut modes, the resource block contains the following two timeout parameters: 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 SHED_OPT parameter defines the block actions when mode shedding occurs, as described in the table below.

Table 4-6. Mode Shedding Parameters

RCAS_IN	SHED_OPT
RCAS_IN_D	SHED_RCAS
ROUT_IN	SHED_ROUT

The shed condition ends when the host writes the remote input parameter. If you select a normal return shed option, the block attempts to enter the original remote mode. If you select a no return shed option, the target mode changes 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.

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
IO_OPTS	STATUS_OPTS

Table 4-7. Option Parameters (Continued)

LOW_CUT	WRITE_LOCK
RESTART	WRITE_PRI

Parameters That Must Be Initialized

You must initialize the following two parameters to a legal value before downloading the configuration.

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
MIN_CYCLE_T	SP_D
MODE_BLK	TEST_RW

Table 4-8. Process Parameters (Continued)

OUT	XD_STATE
OUT_D	—

Tuning Parameters

Tuning parameters allow you to 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 the block to automatically acknowledge alarms with no outside intervention. Use this parameter 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 parameter 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 the block output value must move off a limit, in percent of scale, for the limit status to be turned off.

BKCAL_IN (Limiting, Process)

This parameter links to a downstream block's `BKCAL_OUT` or `SELECT_OUT` parameter. Use this parameter 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 faultstate behavior is active.
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.

Table 4-10. Error Codes (Continued)

Error Code	Code	Description
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 a control is bypassed, the PID uses the setpoint value in percent of scale as the 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. This parameter links to an output of an upstream block. In some blocks, this parameter may be left unlinked. The block uses a local setpoint value (typically SP) instead.

CAS_IN_D (Process)

A remote setpoint value. This parameter links to an output of an upstream block. In some blocks, this parameter may be left unlinked. The block uses a local setpoint value (typically `SP_D`) instead.

CHANNEL (I/O, Process)

Used by I/O function blocks to select a physical I/O channel. The manufacturer defines the mapping.

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 refer to the 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.
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 output value to be increased. When Direct Acting is clear, an increase in the process variable causes the control block 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.

Table 4-11. Control Options (Continued)

Options	Description
`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.

CYCLE_TYPE (Tuning)

Use this parameter to select the block execution method.

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.
`Reports`	The device supports event report messages for alarming. If this feature is not selected in the `FEATURE_SEL` parameter, some devices may continue to detect alarms and events, but do not report alarms and events over the bus. In this case, the host must poll the alarm parameters to detect alarm conditions.
`Faultstate`	The device supports `Faultstate` behavior for output blocks.

Table 4-12. Feature Parameter Options (Continued)

Option	Description
`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.
`Out Readback`	The device provides a way for the action of output transducers to be verified through a readback.
`Direct Write`	The device provides a manufacturer-specific way to directly write to I/O channels.

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 value is zero in preconfigured devices since these devices 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 device 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 device has analog input channels available.
`Analog Output`	This bit is set if the device has analog output channels available.
`Discrete Input`	This bit is set if the device has discrete input channels available.
`Discrete Output`	This bit is set if the device 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`.

Table 4-14. Operation Bitmasks (Continued)

Bitmask	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.
Increase to Close	Remaps the block's scaling so that as the input increases, the output decreases.
Faultstate to Value	When set, the block faultstate behavior sets the output value to the value in FSTATE_VAL. When clear, the block 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, use this parameter 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.

Table 4-16. Field Modes (Continued)

Field Mode	Description
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.

Table 4-17. Operational Modes (Continued)

Operational Mode	Description
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.

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.

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.

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 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_RPC_LOST_CONNECTION

The connection to the remote target is lost. Check the network or the remote target.

NIF_ERR_RUN_MODE_REQUIRED

The NI-FBUS Communications Manager is not in the Run Mode. Launch the NI-FBUS Interface Configuration Utility to switch the current mode to the Run Mode.

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.

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.

NIFCONF_ERR_SAVE_OLDVERSION

This project is an old version, and could not be saved. In order to avoid this error, you should convert the project to the latest version when you open the project.

NIFCONF_ERR_FILENAME_EMPTY

Empty file name. Normally this error occurs when you specify a file without the name.

NIFCONF_ERR_UNSUPPORT_HSE

This project you are opening includes an unsupported HSE feature. You must upgrade your NI-FBUS Configurator to version 3.0 or later to properly open the project.

NIFCONF_ERR_CANT_DELETE_ONLY_ONE_LINK

The last link in the project cannot be deleted. A valid project includes at least one link.

NIFCONF_ERR_WHEN_DISCONNECT_LINKS

The system is busy and the links cannot be disconnected. Try your operation at a later time.

NIFCONF_ERR_EMPTY_TAG

The Empty tag is not permitted due to the FOUNDATION Fieldbus specification.

NIFCONF_ERR_SPACE_IN_TAG

The Space char is not permitted in the middle of the tag due to the FOUNDATION Fieldbus specification.

NIFCONF_ERR_OPERATION_CONFLICT

Operation conflict.

NIFCONF_ERR_UNEXPECTED

An unexpected error has occurred.

NIFCONF_ERR_NO_HOST_DEVICE

There is no host device in one or more H1 links.

NIFCONF_ERR_NO_BLOCK_MATCH

No block matches. If it happens while assigning devices, it means the blocks of an online device can't match the blocks of an offline device.

NIFCONF_ERR_NO_NP_MATCH

No network parameters match. If it happens while assigning devices, it means the network parameters of an online link can't match that of an offline link.

NIFCONF_ERR_MAP_UNCOMPLETED

Device assignment is uncompleted. The device's pdTag, address, block tag, or block parameters are not assigned correctly.

NIFCONF_ERR_LINK_CANT_ACTIVE

The link cannot be set active because the it is either not configured, or it is being used. Only one instance of a link can be active at a time.

E_INTERFACE_INVALID

The interface is currently invalid. It cannot be configured.

E_UNDEFINED

The error is undefined.

E_OFFLINE_ERROR

Offline NI-FBUS error. This is a general error for the offline link or device.

E_BLOCK_NOT_EXIST_IN_CFF

The block is undefined in the related `.cff` file. The file provided by the manufacturer is incomplete and the block to handle is not defined in it.

E_MIB_PARAM_NAME_NOT_FOUND

The `.cff` file provided by the manufacturer is incomplete; the **param** name of MIB is not defined in the related `.cff` file.

E_OFF_DEVICE_NODE_INVALID

The offline device node is invalid. It cannot be configured.

E_CFF_NOT_FOUND

The corresponding `.cff` file is not found under the DD stored directory. Please import the `.cff` file provided by the device manufacturer first.

E_NULL_POINTER

A null pointer has occurred.

E_DEVICE_NOT_FOUND

The device you are looking for is not found.

E_OFFLINE_NOT_SUPPORTED

This feature is not currently supported in offline mode.

E_REVISON_EQUAL

The revision should not be equal.

E_SET_ADDRESS_FAILED

Setting address operation failed.

E_CFF_LOAD_FAILED

Failed to load `.cff` file. The `.cff` file required is neither found nor correct.

E_CFF_NOT_LOAD

The `.cff` file is not loaded. The `.cff` file required is neither found nor correct.

E_CFF_SECT_NOT_FOUND

The section is not found in the `.cff` file. The `.cff` file is incomplete.

E_CFF_VAR_NOT_FOUND

The variable is not found in the `.cff` file. The `.cff` file is incomplete.

E_CFF_DATA_BAD

The data is bad or incomplete in the `.cff` file, which indicates the `.cff` file is either incomplete or corrupt.

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/assignment of the configuration. Look at the **Errors** tab in the status window.

NIFCONF_WARN_NO_OFFLINE_LINK_CFG

There is no offline link configured. Please use the interface configuration utility to configure an offline link.

NIFCONF_WARN_NO_ONLINE_LINK_CFG

There is no online link configured. Please install the interface board and use the interface configuration utility to configure an online link.

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. Complete the following steps:

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

   For more information about editing the schedule, refer to the *View and Edit a Schedule* section of 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 of 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, complete the following 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 of 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 the following 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.

The Fieldbus device stays in `OOS` mode even though the resource block `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 the following solutions:

- 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`.
- See if the `Options` parameter is in `Run`. If so, check the `Block_ERR` in the resource block—it should be zero.
- 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 the 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 resource block is off-line, the BLOCK_ERR parameter should read Resource Block Offline, indicating that you must change the resource block to Auto mode before you can change the function block mode.

Why are discrete states integers instead of booleans?

Because it allows for multiple state discrete functionality such as off, turning on, on, and turning off. Not all devices take full advantage of this. 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 to 20 mA signal. The range for my analog input block is 4 to 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 also can force the NI-FBUS Configurator to treat Uncertain as Good. This is done on a function block-by-function block

basis. Double-click the block for which you would like to change this setting. Under the **Options** tab, locate the parameter called `STATUS_OPTS`. If you click its field, a drop-down menu will appear with a checkbox called **Uncertain as Good**. If you select this option and click **Write Changes**, the function block will now treat Uncertain quality received on its input as Good.

Missing Device

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

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. Launch the Interface Configuration Utility and click the **Advanced** button.
4. Set the following parameter values:

Parameter	Value
Number of unpolled nodes	Refer to the *Setting Number of Polled Addresses* section
MinInterPduDelay	20
Slot time	10
Max Response delay	8

5. Shut down the NI-FBUS Configurator and NIFB process.
6. Launch the NI-FBUS Communications Manager to restart the `nifb` process.
7. Launch the NI-FBUS Configuator to restart the NI-FBUS Configurator.

If your device still does not show up on the bus, check your device 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. Launch the Interface Configuration Utility. Click the appropriate port. Click 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 Interface Configuration Utility. Click the

appropriate port. Click 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 the open descriptors. Choose **Expand All**. Then, right-click 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 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 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, and 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 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* section and the *Configuring Trends* section of 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 the **Schedule** icon in the configuration tree.

   For more information about editing the schedule, refer to the *View and Edit a Schedule* section of 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 of 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.

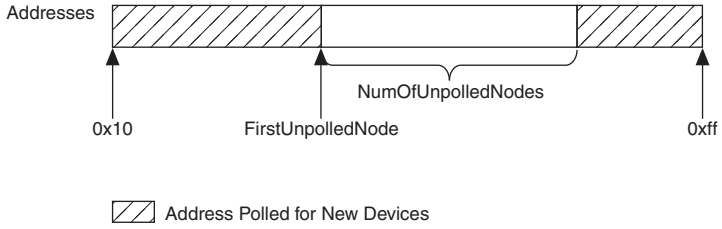
I get a Read vfdlist fail error.

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 of 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 **Network Parameters** in the configuration tree and ensuring that the checkboxes for the appropriate link master devices are checked. If a device does not appear in this list, complete the following steps:

1. Right-click the interface in the configuration tree and select **Advanced View**. The MIB parameters appear underneath the interface board.
2. Double-click 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 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 the MIB will bring up a block configuration window where these values can be changed.

- Make the interface `PRIMARY_LINK_MASTER_FLAG` parameter `FALSE`.
- Make the interface `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

Table B-1. Rosemount Advanced Parameter Settings (Continued)

Parameter	Value
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 of the *FOUNDATION Fieldbus Overview* manual.

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. Complete the following steps to add your own tabs to a block with menus and methods:

1. Launch the Interface Configuration Utility.
2. Click 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(s).
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 the 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 of Chapter 2, *Using the NI-FBUS Configurator*. Notice that you do not need a device description for the National Instruments interface boards.

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 of 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 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 also can be brought up by selecting the **Add or Delete Custom Tab** button in the block configuration window.

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`. Refer to 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.

Open/Find Project

If you want to load a saved process file, refer to the *Open an Existing Project* section of Chapter 2, *Using the NI-FBUS Configurator*, for instructions.

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, and so on) 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 the link (the top level in the hierarchy) and selecting **Take Online** or **Take Offline**.

Restart Devices

You can do a software restart of a device on-board processor by double-clicking the device 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 **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 the group and selecting **Activate Group**.
6. When you open the file you saved and want to go live, right-click **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.

C

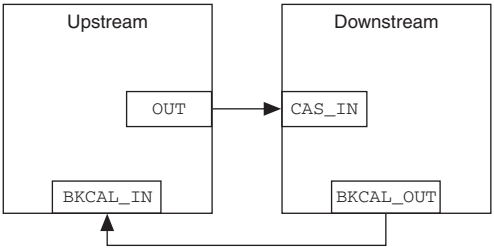
Cascade and Remote Cascade Setups

Cascade Initialization

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 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 in the following figure.



Mode and Status Behavior During Cascade Initialization

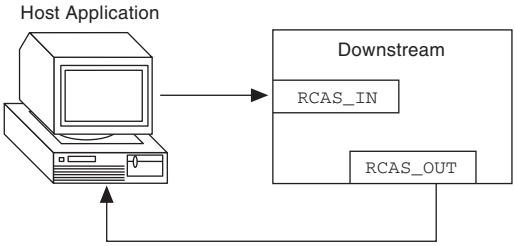
Cascade initialization is arbitrated through the status of the backward calculation path from the downstream block and the forward calculation path on the upstream block. If the upstream block publishes a status of `Good, Non-Cascade`, it does not support cascade initialization, and the lower block immediately 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

Technical Support and Professional Services

Log in to your National Instruments ni.com User Profile to get personalized access to your services. Visit the following sections of ni.com for technical support and professional services:

- **Support**—Technical support at ni.com/support includes the following resources:
  - **Self-Help Technical Resources**—For answers and solutions, visit ni.com/support for software drivers and updates, a searchable KnowledgeBase, product manuals, step-by-step troubleshooting wizards, thousands of example programs, tutorials, application notes, instrument drivers, and so on. Registered users also receive access to the NI Discussion Forums at ni.com/forums. NI Applications Engineers make sure every question submitted online receives an answer.
  - **Standard Service Program Membership**—This program entitles members to direct access to NI Applications Engineers via phone and email for one-to-one technical support, as well as exclusive access to self-paced online training modules at ni.com/self-paced-training. All customers automatically receive a one-year membership in the Standard Service Program (SSP) with the purchase of most software products and bundles including NI Developer Suite. NI also offers flexible extended contract options that guarantee your SSP benefits are available without interruption for as long as you need them. Visit ni.com/ssp for more information.

    For information about other technical support options in your area, visit ni.com/services, or contact your local office at ni.com/contact.
- **Training and Certification**—Visit ni.com/training for training and certification program information. You can also register for instructor-led, hands-on courses at locations around the world.
- **System Integration**—If you have time constraints, limited in-house technical resources, or other project challenges, National Instruments Alliance Partner members can help. To learn more, call your local NI office or visit ni.com/alliance.
- **Declaration of Conformity (DoC)**—A DoC is our claim of compliance with the Council of the European Communities using the manufacturer's declaration of conformity. This system affords the user protection for electromagnetic compatibility (EMC) and product safety. You can obtain the DoC for your product by visiting ni.com/certification.
- **Calibration Certificate**—If your product supports calibration, you can obtain the calibration certificate for your product at ni.com/calibration.

You also can visit the Worldwide Offices section of ni.com/niglobal to access the branch office Web sites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

Glossary

Symbol	Prefix	Value
m	milli	10^{-3}
k	kilo	10^{3}

Numbers

4 to 20 mA system	Traditional control system in which a computer or control unit provides control for a network of devices controlled by 4 to 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 buses 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.
CSV	A text file using the Comma Delimited format.
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 retransmits 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.
FBUS	In this manual, this term refers to the NI-FBUS software, including the NI-FBUS Communications Manager, NI-FBUS Configurator, and NI-FBUS Monitor.
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. The generic term fieldbus refers to any bus that connects to field devices. Field devices include FOUNDATION Fieldbus, CAN, DNET, and Profibus. In this manual, the term Fieldbus refers specifically to the FOUNDATION Fieldbus.
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.
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 10 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/s type of Fieldbus.
hex	Hexadecimal. A base-16 numbering system which uses 0 to 9 and A to 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

I/O	Input/output.
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.
IP	Internet Protocol. An international-standard protocol used in network transmission.

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.
MIB	Management Information Base.
Mode	Type of communication.

N

`NIFB.EXE`	The NIFB process that must be running in the background for you to use your NI-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 NI-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 to 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

A

ACK_OPTION parameter, 4-10
addresses
    problems setting addresses, B-7
    setting device addresses, 2-7
    setting number of polled addresses, B-7
AI and AO blocks, connecting (tutorial), 3-2
AI scaling procedure (figure), 4-4
alarm configuration, 2-30
ALARM_HYS parameter, 4-10
ALARM_SUM parameter, 4-10
alarming parameters, 4-1
ALERT_KEY parameter, 4-10
AO scaling procedure (figure), 4-5
Automatic mode
    MODE_BLK parameter (table), 4-24
    overview, 2-11
    troubleshooting, B-1
Automatic Mode Handling option, 2-37

B

bad status troubleshooting, B-3
BAL_TIME parameter, 4-11
batch-edit parameters, 2-40
BIAS parameter, 4-11
BKCAL_HYS parameter, 4-11
BKCAL_IN parameter, 4-11
BKCAL_OUT parameter, 4-11
BKCAL_OUT_D parameter, 4-11
block configuration window
    adding tabs, 2-12
    adding/deleting tabs, troubleshooting, B-10
    categorizing block parameters, 2-12
    changing parameter settings, 1-11
    configuring block parameters, 2-12
    customizing parameters on a tab, 2-13
    editing block parameters, 2-14
    icons (table), 1-12
    modifying your view, 2-14
    opening, 1-10
    removing tabs, 2-13
    tabs, 1-12
    units of time, 1-12
    updating block parameters
        automatically, 2-16
        manually, 2-16
block parameters. *See* block configuration window
block tags, setting, 2-10
BLOCK_ALM parameter, 4-11
BLOCK_ERR parameter, error codes (table), 4-12
BYPASS parameter, 4-13

C

CAS_IN parameter, 4-13
CAS_IN_D parameter, 4-13
Cascade Initialization
    bypassing, C-3
    mode and status behavior during, C-1
    overview, C-1
    parameter connections, C-1
    remote cascades, C-2
Cascade mode
    See also Remote Cascade mode
    MODE_BLK parameter (table), 4-24
    overview, 2-11
change device type, 2-40
CHANNEL parameter, 4-13
CLR_FSTATE parameter, 4-14
common questions. *See* troubleshooting and common questions
configuration tree, 1-3
configurations
    printing, 2-43
    verifying
        checking for errors, 2-39
        comparing two configurations, 2-39
        quick verification, 2-38
CONFIRM_TIME parameter, 4-14

connecting function blocks
checking for errors, 2-19
creating basic connections to AI and AO blocks (tutorial), 3-2
wiring blocks manually, 2-18
connecting to network and hardware (tutorial), 3-1
CONTROL_OPTS parameter (table), 4-14
conventions used in the manual, *vi*
CYCLE_SEL parameter, 4-15
CYCLE_TYPE parameter, 4-15

D

DD_RESOURCE parameter, 4-15
DD_REV parameter, 4-15
deleting function blocks, 2-10
DEV_REV parameter, 4-15
DEV_TYPE parameter, 4-15
device type, changing, 2-40
device window, 1-9
devices
changing type, 2-40
connecting to network and hardware (tutorial), 3-1
missing device troubleshooting, B-4
replacing, 2-40
resetting to factory defaults, 2-40
restarting, B-13
setting device addresses, 2-7
setting tags, 2-10
viewing device properties, 2-8
diagnostic parameters, 4-2
diagnostic tools (NI resources), D-1
DISC_ALM parameter, 4-15
DISC_LIM parameter, 4-15
DISC_PRI parameter, 4-15
documentation
conventions used in the manual, *vi*
NI resources, D-1
Download tab, status window, 1-4
downloading project configuration, 2-35
bringing blocks online, 2-37
download options
Automatic Mode Handling, 2-37
Clear Devices, 2-37
Verify & Diff, 2-37
Write Contained Block Parameters, 2-36
drivers (NI resources), D-1
DV_HI_ALM parameter, 4-16
DV_HI_LIM parameter, 4-16
DV_HI_PRI parameter, 4-16
DV_LO_ALM parameter, 4-16
DV_LO_LIM parameter, 4-16
DV_LO_PRI parameter, 4-16

E

error checking, project configuration, 2-39
errors
BLOCK_ERR error codes (table), 4-12
error messages, A-1
read/write errors, B-6
Errors tabs, status window, 1-4
examples (NI resources), D-1
export and import parameters, 2-40

F

factory defaults, returning to, 2-40
FAULT_STATE parameter, 4-16
faultstate parameters, 4-2
FEATURE_SEL/FEATURES parameter (table), 4-16
FF_GAIN parameter, 4-17
FF_SCALE parameter, 4-17
FF_VAL parameter, 4-17
FIELD_VAL parameter, 4-17
FIELD_VAL_D parameter, 4-17
Fieldbus
using with Lookout, B-12
using with Server Explorer, B-13
FREE_SPACE parameter, 4-17
FREE_TIME parameter, 4-17
FSTATE_TIME parameter, 4-18
FSTATE_VAL parameter, 4-18
FSTATE_VAL_D parameter, 4-18
Function Block Application Editor, 2-16
adding additional applications, 2-39
adding blocks, 2-17
connecting function blocks, wiring blocks manually, 2-18

customizing function block application, 2-19
export function block application image, 2-23
opening, 1-7
overview, 1-7
templates, 2-20
creating templates, 2-23
inserting templates, 2-20
function block modes. *See* MODE_BLK parameter
function blocks
See also block configuration window
block diagram, 2-16
bringing online, 2-37
creating basic connections to AI and AO blocks (tutorial), 3-2
determining execution time using loops, 2-25
instantiation and deletion, 2-10
missing function blocks, B-11
missing information, B-10
symbolic information, B-10

G

GAIN parameter, 4-18
GRANT_DENY parameter, 4-18

H

HARD_TYPES parameter, bitmasks (table), 4-18
help
technical support, D-1
window, 1-3
HI_ALM parameter, 4-18
HI_HI_ALM parameter, 4-19
HI_HI_LIM parameter, 4-19
HI_HI_PRI parameter, 4-19
HI_LIM parameter, 4-19
HI_PRI parameter, 4-19
HSE/H1 Linking Device
device window, 1-9
network parameters window disabled, 1-9

I

I/O parameters, 4-7
IN parameter, 4-19
IN_1 parameter, 4-19
Initialization Manual mode (table), 4-23
initialization requirements for parameters, 4-9
instantiation of function blocks, 2-10
instrument drivers (NI resources), D-1
IO_OPTS parameter, operation bitmasks (table), 4-19
ITK_VER parameter, 4-20

K

KnowledgeBase, D-1

L

L_TYPE parameter, 4-20
linearization types (table), 4-21
language, changing, 2-35
LIM_NOTIFY parameter, 4-21
limiting parameters (table), 4-7
link active schedule, changing, 2-33
links
adding links, 2-1
missing links, B-8
LO_ALM parameter, 4-21
LO_LIM parameter, 4-21
LO_LO_ALM parameter, 4-21
LO_LO_LIM parameter, 4-21
LO_LO_PRI parameter, 4-21
LO_PRI parameter, 4-22
Lookout, using with Fieldbus, B-12
loops
See also multiple loops
changing block execution order, 2-26
defining, 2-23
determining function block execution time, 2-25
multiple loop representation in schedule window, 2-34
multiple loops running at different rates, 2-24
setting the Stale Limit field, 2-27
LOW_CUT parameter, 4-22

M

Manual mode
MODE_BLK parameter, 4-23
overview, 2-11
MANUFAC_ID parameter, 4-22
MAX_NOTIFY parameter, 4-22
MEMORY_SIZE parameter, 4-22
methods, using, 2-28
middle frame
examples (figure), 1-6
overview, 1-5
MIN_CYCLE_T parameter, 4-22
missing device troubleshooting, B-4
missing function blocks, B-11
missing information, B-10
missing link, B-8
mode problems, troubleshooting, B-1
mode shedding parameters, 4-7
MODE_BLK parameter
ACTUAL mode (table), 4-23
common operational modes, 2-11
field modes (table), 4-23
NORMAL mode (table), 4-23
operational modes (table), 4-23
overview, 2-11
PERMITTED mode (table), 4-23
TARGET mode (table), 4-23
monitoring parameters, 2-38
multiple loops
representation in schedule window, 2-34
running at different rates, 2-24

N

National Instruments support and services, D-1
network parameters, setting, 2-34
NIFB process
restarting, B-13
starting, 2-1
NI-FBUS Configurator
changing software key, 2-43
main window
components (figure), 1-2
configuration tree, 1-3
device window, 1-3
help window, 1-3
opening, 1-1
overview, 1-1
status window
Download tab, 1-4
Errors tab, 1-4
status tab, 1-4
middle frame
examples (figure), 1-6
overview, 1-5
overview, 1-1
starting, 2-1
tutorials
connecting to network and hardware, 3-1
creating basic connections to AI and AO blocks, 3-2
PID control of simulated process, 3-4
uninstalling, B-13
windows
block configuration window, 1-10
device window, 1-9
Function Block Application Editor window, 1-7
main window, 1-1
middle frame, 1-5
network parameters window, 1-9
NV_CYCLE_T parameter, 4-24

O

option parameters, 4-8
Out of Service (OOS) mode
MODE_BLK parameter (table), 4-23
overview, 2-11
Set to OOS Mode checkbox (figure), 2-3, 2-4, 2-5, 2-6, 2-8
setting new address, 2-7
troubleshooting problems, B-2
OUT parameter, 4-24
OUT_D parameter, 4-24
OUT_HI_LIM parameter, 4-24
OUT_LO_LIM parameter, 4-24
OUT_SCALE parameter (table), 4-25
OUT_STATE parameter, 4-25

P

parameters, 4-1
See also block configuration window
ACK_OPTION parameter, 4-10
AI scaling procedure (figure), 4-4
ALARM_HYS parameter, 4-10
ALARM_SUM parameter, 4-10
alarming parameters, 4-1
ALERT_KEY parameter, 4-10
AO scaling procedure (figure), 4-5
BAL_TIME parameter, 4-11
BIAS parameter, 4-11
BKCAL_HYS parameter, 4-11
BKCAL_IN parameter, 4-11
BKCAL_OUT parameter, 4-11
BKCAL_OUT_D parameter, 4-11
BLOCK_ALM parameter, 4-11
BLOCK_ERR parameter, 4-12
error codes (table), 4-12
BYPASS parameter, 4-13
CAS_IN parameter, 4-13
CAS_IN_D parameter, 4-13
changing settings, 1-11
CHANNEL parameter, 4-13
CLR_FSTATE parameter, 4-14
CONFIRM_TIME parameter, 4-14
CONTROL_OPTS parameter (table), 4-14
CYCLE_SEL parameter, 4-15
CYCLE_TYPE parameter, 4-15
DD_RESOURCE parameter, 4-15
DD_REV parameter, 4-15
DEV_REV parameter, 4-15
DEV_TYPE parameter, 4-15
diagnostic parameters, 4-2
DISC_ALM parameter, 4-15
DISC_LIM parameter, 4-15
DISC_PRI parameter, 4-15
DV_HI_ALM parameter, 4-16
DV_HI_LIM parameter, 4-16
DV_HI_PRI parameter, 4-16
DV_LO_ALM parameter, 4-16
DV_LO_LIM parameter, 4-16
DV_LO_PRI parameter, 4-16
export and import, 2-40
FAULT_STATE parameter, 4-16
faultstate parameters, 4-2
FEATURE_SEL/FEATURES parameter (table), 4-16
FF_GAIN parameter, 4-17
FF_SCALE parameter, 4-17
FF_VAL parameter, 4-17
FIELD_VAL parameter, 4-17
FIELD_VAL_D parameter, 4-17
FREE_SPACE parameter, 4-17
FREE_TIME parameter, 4-17
FSTATE_TIME parameter, 4-18
FSTATE_VAL parameter, 4-18
FSTATE_VAL_D parameter, 4-18
GAIN parameter, 4-18
GRANT_DENY parameter, 4-18
HARD_TYPES parameter bitmasks (table), 4-18
HI_ALM parameter, 4-18
HI_HI_ALM parameter, 4-19
HI_HI_LIM parameter, 4-19
HI_HI_PRI parameter, 4-19
HI_LIM parameter, 4-19
HI_PRI parameter, 4-19
I/O parameters, 4-7
IN parameter, 4-19
IN_1 parameter, 4-19
initialization requirements, 4-9
IO_OPTS parameter, operation bitmasks (table), 4-19
ITK_VER parameter, 4-20
L_TYPE parameter, linearization types (table), 4-21
LIM_NOTIFY parameter, 4-21
limiting parameters (table), 4-7
LO_ALM parameter, 4-21
LO_LIM parameter, 4-21
LO_LO_ALM parameter, 4-21
LO_LO_LIM parameter, 4-21
LO_LO_PRI parameter, 4-21
LO_PRI parameter, 4-22
LOW_CUT parameter, 4-22
MANUFAC_ID parameter, 4-22
MAX_NOTIFY parameter, 4-22
MEMORY_SIZE parameter, 4-22

MIN_CYCLE_T parameter, 4-22
mode shedding parameters, 4-7
MODE_BLK parameter
field modes (table), 4-23
operational modes (table), 4-23
monitoring, 2-38
network parameters, setting, 2-34
NV_CYCLE_T parameter, 4-24
option parameters, 4-8
OUT parameter, 4-24
OUT_D parameter, 4-24
OUT_HI_LIM parameter, 4-24
OUT_LO_LIM parameter, 4-24
OUT_SCALE parameter (table), 4-25
OUT_STATE parameter, 4-25
PID scaling procedure, 4-6
process parameters, 4-9
PV parameter, 4-25
PV_D parameter, 4-25
PV_FTIME parameter, 4-25
PV_SCALE parameter, 4-25
PV_STATE parameter, 4-25
RA_FTIME parameter, 4-26
RATE parameter, 4-26
RCAS_IN parameter, 4-26
RCAS_IN_D parameter, 4-26
RCAS_OUT parameter, 4-26
RCAS_OUT_D parameter, 4-26
READBACK parameter, 4-26
READBACK_D parameter, 4-26
RESET parameter, 4-26
RESTART parameter, restart values (table), 4-27
ROUT_IN parameter, 4-27
ROUT_OUT parameter, 4-27
RS_STATE parameter, device states (table), 4-27
scaling parameters, 4-3
table, 4-6
SEL_1 through SEL_3 parameter, 4-28
SEL_TYPE parameter, 4-28
SET_FSTATE parameter, 4-28
SHED_OPT parameter, 4-28
shed conditions (table), 4-28
SHED_RCAS parameter, 4-28
SHED_ROUT parameter, 4-28
SIMULATE parameter, 4-28
SIMULATE_D parameter, 4-29
SP parameter, 4-29
SP_D parameter, 4-29
SP_HI_LIM parameter, 4-29
SP_LO_LIM parameter, 4-29
SP_RATE_DN parameter, 4-29
SP_RATE_UP parameter, 4-29
ST_REV parameter, 4-29
STATUS_OPTS parameter, status options (table), 4-30
STRATEGY parameter, 4-31
TAG_DESC parameter, 4-31
TEST_RW parameter, 4-31
TRK_IN_D parameter, 4-31
TRK_SCALE parameter, 4-31
TRK_VAL parameter, 4-31
tuning parameters, 4-10
UPDATE_EVT parameter, 4-31
WRITE_ALM parameter, 4-31
WRITE_LOCK parameter, 4-31
WRITE_PRI parameter, 4-31
XD_SCALE parameter, scaling parameter values (table), 4-32
XD_STATE parameter, 4-32
performance problems, B-8
PID control of simulated process (tutorial), 3-4
PID scaling procedure, 4-6
polled addresses, setting number of, B-7
preferences, setting, 2-42
printing project configuration, 2-43
problems. *See* troubleshooting and common questions
process parameters, 4-9
programming examples (NI resources), D-1
projects
creating new project, 2-1
downloading configuration, 2-35
bringing blocks online, 2-37
download options, 2-36
opening existing projects, 2-2
printing, 2-43
problems with opening/finding, B-13

saving configuration, 2-35
uploading project configuration, 2-7
PV parameter, 4-25
PV_D parameter, 4-25
PV_FTIME parameter, 4-25
PV_SCALE parameter, 4-25
PV_STATE parameter, 4-25

Q

quick verification, 2-38

R

RA_FTIME parameter, 4-26
RATE parameter, 4-26
RCAS_IN parameter, 4-26
RCAS_IN_D parameter, 4-26
RCAS_OUT parameter, 4-26
RCAS_OUT_D parameter, 4-26
read/write errors, B-6
READBACK parameter, 4-26
READBACK_D parameter, 4-26
Remote Cascade mode, 4-24, C-2
Remote Output mode (table), 4-24
replacing devices, 2-40
RESET parameter, 4-26
RESTART parameter, restart values (table), 4-27
restarting
devices, B-13
NIFB, B-13
ROUT_IN parameter, 4-27
ROUT_OUT parameter, 4-27
RS_STATE parameter, device states (table), 4-27

S

saving project configuration, 2-35
scaling parameters
AI scaling procedure (figure), 4-4
AO scaling procedure (figure), 4-5
L_TYPE, 4-3
list of parameters (table), 4-6
OUT_SCALE, 4-3
PID scaling procedure (figure), 4-6
table, 4-6
XD_SCALE, 4-3
schedule window, overview, 1-7
schedules
changing link active schedule, 2-33
multiple loop representation, 2-34
SEL_1 through SEL_3 parameter, 4-28
SEL_TYPE parameter, 4-28
Server Explorer, using with Fieldbus, B-13
SET_FSTATE parameter, 4-28
SHED_OPT parameter, shed conditions (table), 4-28
SHED_RCAS parameter, 4-28
SHED_ROUT parameter, 4-28
SIMULATE parameter, 4-28
SIMULATE_D parameter, 4-29
software (NI resources), D-1
software key, changing, 2-43
SP parameter, 4-29
SP_D parameter, 4-29
SP_HI_LIM parameter, 4-29
SP_LO_LIM parameter, 4-29
SP_RATE_DN parameter, 4-29
SP_RATE_UP parameter, 4-29
ST_REV parameter, 4-29
Stale Limit, setting, 2-27
Start Up dialog box, 2-1
starting NI-FBUS Configurator, 2-1
status window
Download tab, 1-4
Errors tab, 1-4
Status tab, 1-4
STATUS_OPTS parameter, status options (table), 4-30
STRATEGY parameter, 4-31
support, technical, D-1
symbolic information, missing, B-10

T

tabs. *See* block configuration window
TAG_DESC parameter, 4-31
tags, setting, 2-10
technical support, D-1

templates
available templates, 2-20
creating templates, 2-23
definition, 2-20
example (figure), 2-22
inserting templates, 2-20
TEST_RW parameter, 4-31
time units, in block configuration window, 1-12
training and certification (NI resources), D-1
trends, configuring, 2-31
TRK_IN_D parameter, 4-31
TRK_SCALE parameter, 4-31
TRK_VAL parameter, 4-31
troubleshooting (NI resources), D-1
troubleshooting and common questions
add/delete tabs, B-10
bad status, B-3
missing device, B-4
missing function blocks, B-11
missing information, B-10
missing link, B-8
mode problems, B-1
open/find project, B-13
performance problems, B-8
read/write errors, B-6
restarting devices, B-13
restarting NIFB, B-13
set address problems, B-7
uninstalling NI-FBUS, B-13
using Fieldbus
with Lookout, B-12
with Server Explorer, B-13
viewing problems, B-11
tuning parameters, 4-10
tutorials
connecting to network and hardware, 3-1
creating basic connections to AI and AO blocks, 3-2
PID control of simulated process, 3-4

U

uninstalling NI-FBUS Configurator, B-13
units of time, in block configuration window, 1-12
UPDATE_EVT parameter, 4-31
uploading project configuration, 2-7
user methods, using, 2-28

V

Verify & Diff option, 2-37
verifying configuration
checking for errors, 2-39
comparing two configurations, 2-39
quick verification, 2-38
view
modifying in block configuration window, 2-14
troubleshooting problems, B-11
viewing device properties, 2-8

W

warnings, A-12
Web resources, D-1
wiring function blocks, wiring blocks manually, 2-18
Write Contained Block Parameters, 2-36
WRITE_ALM parameter, 4-31
WRITE_LOCK parameter, 4-31
WRITE_PRI parameter, 4-31

X

XD_SCALE parameter, scaling parameter values (table), 4-32
XD_STATE parameter, 4-32

Z

Zoom menu, Function Block Application Editor, 2-19