DeviceLogix System

User Manual

Allen-Bradley • Rockwell Software

Rockwell Automation

Important User Information

Solid state equipment has operational characteristics differing from those of electromechanical equipment. Safety Guidelines for the Application, Installation and Maintenance of Solid State Controls (publication SGI-1.1 available from your local Rockwell Automation sales office or online at http://www.rockwellautomation.com/literature/) describes some important differences between solid state equipment and hard-wired electromechanical devices. Because of this difference, and also because of the wide variety of uses for solid state equipment, all persons responsible for applying this equipment must satisfy themselves that each intended application of this equipment is acceptable.

In no event will Rockwell Automation, Inc. be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, Rockwell Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams.

No patent liability is assumed by Rockwell Automation, Inc. with respect to use of information, circuits, equipment, or software described in this manual.

Reproduction of the contents of this manual, in whole or in part, without written permission of Rockwell Automation, Inc., is prohibited.

Throughout this manual, when necessary, we use notes to make you aware of safety considerations.

WARNING 	Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.
IMPORTANT	Identifies information that is critical for successful application and understanding of the product.
ATTENTION 	Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you identify a hazard, avoid a hazard, and recognize the consequence
SHOCK HAZARD 	Labels may be on or inside the equipment, for example, a drive or motor, to alert people that dangerous voltage may be present.
BURN HAZARD 	Labels may be on or inside the equipment, for example, a drive or motor, to alert people that surfaces may reach dangerous temperatures.

Summary of Changes

Introduction

The release of this document contains new and updated information. To find new and updated information, look for change bars, as shown next to this paragraph.

Updated Information

This document contains the following changes.

Topic	Page
Screen format options for download to device	61, 121
PID instruction supported in Function Block Editor and in New Ladder Editor	Chapters 2, 3, 6, and 7
ACC binding for Timer/Counter instructions supported in Function Block Editor and in New Ladder Editor	Chapters 2 and 6
Macro function supported in Function Block Editor and in New Ladder Editor	Chapters 3 and 7
New Ladder Editor introduced	Chapters 6 and 7

Notes:

Table of Contents

Navigate the Old Ladder Editor Interface

Create Logic in the Old DeviceLogix Ladder Editor

Navigate the New Ladder Editor Interface

Create Logic in the New Ladder Editor

Register EDS Files and Add Devices Offline/Online

RSNetWorx for DeviceNet Software and the DeviceLogix Editors

Preface

Purpose of This Manual

This manual describes how to install and configure devices using DeviceLogix. It also describes how to navigate and use the old DeviceLogix Ladder Editor, the new DeviceLogix Ladder Editor, and the DeviceLogix Function Block Editor.

Who Should Use This Manual

This manual is intended for engineers and technicians who use DeviceLogix to control outputs and manage information locally within devices.

This document assumes that you are familiar with one or more of the following working environments:

- RSNetWorx for DeviceNet software (including the configuration of distributed I/O devices)
- Drive Tools (including DriveExplorer, DriveTools SP, and Drive Add-On Profiles)

Related Terms

Refer to the Related Terms table to become familiar with DeviceLogix.

Related Terms

Name	Description
Download	The transfer of logic from the software memory to the device.
Logic	Logic consists of function blocks or ladder logic and their interconnnections that can reside on a DeviceLogix device.
MAC ID	Media Access Control Identifier - An integer identification value assigned to each node on DeviceNet. This value distinguishes a node among all other nodes on the same link.
NAN	Not a Number - Value that is typically produced as a the result of an operation on invalid input operands, especially in floating-point calculations.
Upload	The transfer of logic from the device memory to the software memory.

Common Techniques Used in This Manual

The following conventions are used throughout this manual:

- Bulleted lists provide information, not procedural steps.
- Numbered lists provide sequential steps.
- Pictures of keys and/or dialogs represent the actual keys you press or the dialogs you use.
- Actions you must perform appear in bold and look like the following example: Select **Unregister a device**.
- A menu item in this format **Network > Online** identifies the menu item (**Network**) and the submenu item (**Online**) after the caret (**>**).

TIP Tips contain helpful information.

What is DeviceLogix Functionality?

Introduction

DeviceLogix functionality has been added to a number of Rockwell Automation devices to control outputs and manage status information locally within the device.

The configuration of the DeviceLogix functionality is accomplished through the DeviceLogix Editor. The DeviceLogix Editor includes two kinds of logic configuration tools for DeviceLogix devices to meet different configuration preferences:

- Function Block Editor - provides a graphical interface for configuring function blocks to provide local control within DeviceLogix-capable devices.
- Ladder Editor - provides a ladder-style configuration tool for DeviceLogix-capable devices. Beginning with firmware revision 4, two Ladder Editors are available within DeviceLogix. The Old Ladder Editor supports DeviceLogix, firmware revision 3 and earlier. The New Ladder Editor supports DeviceLogix, firmware revision 4.

The DeviceLogix Editor is an applet of RSNetWorx for DeviceNet software and Drive Tools software, and it can be launched directly from those hosts.

With DeviceLogix-capable devices, you can enable a logic operation using the DeviceLogix Editor to provide local control over the device's operation. A DeviceLogix device consists of:

- a specific number of inputs and/or outputs.
- local logic that determines its behavior.

Inputs and Outputs

Inputs and outputs can be one of two types.

- Physical - Inputs and outputs realized by physical connections to the device. These are referred to as Discrete/Analog Inputs or Discrete/Analog Outputs.
- Networked - Inputs consumed by the device from the network and outputs produced by the device onto the network.

Input and Output Bits

There are five types of DeviceLogix inputs. Inputs are read from the Electronic Data Sheet (EDS) file or are created dynamically during logic configuration. The inputs that are read from the EDS file cannot be modified. The DeviceLogix inputs are:

- Device Input - A physical input of the device. Device inputs represent the actual inputs, such as sensors and switches, attached to a particular device.

- Network Input - Formerly called the Consumed Network Bit (CNB), network input is data sent from a master that can be used in the device's logic.

- Device Status - Status inputs indicate the state of the device. For example, if an explicit message connection exists between the device and a master, an input called 'explicit connection exists' is set to true and possibly affects the logic the device performs.

- Device Fault - Faults are conditions that report device errors. For example, if a device detects a short circuit on an output, a fault input is set to true and possibly affects the logic the device performs.

There are two types of DeviceLogix outputs:

- Device Output - Hardware outputs that are the actual outputs, such as lights and actuators, attached to a particular device. Without DeviceLogix functionality, the master would normally control the outputs via consumed data. In fact, if there is no local logic controlling an output, the master controls the output as it would if DeviceLogix functionality were not running on the device. However, within DeviceLogix functionality, if the local logic controls an output, the master no longer controls the output. The only way the master can affect the state of an output that is under local control is to route requests to the local logic by using network inputs. Some outputs can be under local control while others can still be controlled by the master.

- Network Output - Formerly called Produced Network Bit (PNB), Network outputs report the results of the local logic to a master and are part of the produced data from the device.

Local Function Block Logic

The local logic of a DeviceLogix device consists of function blocks, inputs, outputs, and connections (wires) between them. Function blocks contain connection points (called pins) and perform a specific function. Inputs and outputs also have connection pins and represent the actual hardware devices, networked data, and fault and status bits that are available for use in the local logic.

A connection (wire) between function blocks is defined when an input pin of one function block is bound to an output pin of another function block. A pin can be bound to a:

- pin of another function block.
- physical input/output.
- networked input/output.
- fault or status bit.
- miscellaneous bit.
- block input enable bit and block output enable bit.

Function blocks may also have attributes that influence their function.

Configuring a DeviceLogix device consists of defining or editing the local logic that is present on the device along with the EDS parameters for that device.

Local Ladder Logic

The local logic of a DeviceLogix device consists of rung, branch, contact, output coil, and box instructions. A box instruction performs a specific function (such as Timer or Counter). Contact and coil instructions could be hardware data, networked data, and fault and status bits that are available for use in the local logic. Additionally, contact could also refer the output of a box instruction.

The DeviceLogix Editor is a graphical tool for building DeviceLogix functionality in DeviceLogix-enabled products. With the editor, you can create logic, bind logic input and output, verify logic, upload/download logic and enable/disable logic in DeviceLogix-enabled products. When logic is running (in online mode and when logic is enabled), real-time data is animated in the editor and you can also implement forces or perform online parameter modification for some function types.

DeviceLogix Functionality and Associated Host Software

You configure DeviceLogix features through RSNetWorx for DeviceNet software and Drives Tools software. The DeviceLogix Editor ships as part of RSNetWorx for DeviceNet software, starting with revision 3.0. EDS files that enable DeviceLogix functionality are also shipped with RSNetWorx for DeviceNet software in a separate folder labeled **Additional EDS Files**.

For more information on registering EDS files, see Appendix A.

For more information on configuring RSNetWorx for DeviceNet software for use with the DeviceLogix Editors, see Appendix B.

RSNetWorx for DeviceNet Software Operating Modes

RSNetWorx for DeviceNet software lets you select online or offline mode, as described below.

- Offline - RSNetWorx for DeviceNet software is not connected to the network.
- Online - RSNetWorx for DeviceNet software is connected to the network and is capable of communicating with devices on the network.

Drive Tools Software Operating Modes

Drive Tools software lets you select online mode, as described below.

- Online - Drive Tools software is connected to the network and is capable of communicating with devices on the network.

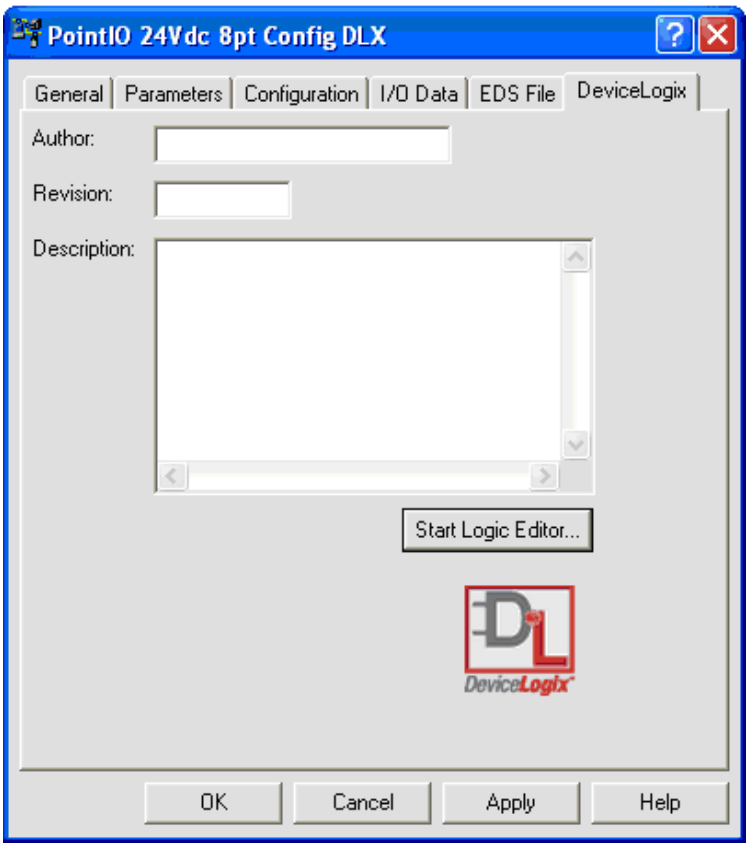
DeviceLogix Operating Modes

The mode that RSNetWorx for DeviceNet software is in directly affects the way the DeviceLogix Editor behaves when it is launched. When online with the device, DeviceLogix functionality provides two alternative states: Pending Edits and Animated.

Mode		Description
Offline		The DeviceLogix Editor does not communicate with the device. If RSNetWorx software is offline, the DeviceLogix Editor is also offline. When offline, you can edit existing DeviceLogix configurations or create new configurations.
Online	Pending Edits	When online with a device, pressing the edit button or selecting **Tools > Edit** enables Pending Edits. Pending Edits allows a device's configuration to be edited while online. When your edits are complete, the configuration must be downloaded to the device.
	Animated	When online and animated, DeviceLogix functionality allows a device's configuration to be monitored in "real time". Real time includes comms throughput latencies. Depending on the device, you may be able to change presets and accumulated values.

Launch the DeviceLogix Editor

After you configure the properties for your DeviceLogix-enabled device (for more information, see Appendix A), you can launch the DeviceLogix Editor. You see an additional tab in the device properties dialog box for all DeviceLogix-enabled devices. This tab is labeled DeviceLogix. This tab provides access to the start-up window for the DeviceLogix Editor. You have the option to fill in your name, a revision number, and a description of your configuration (all optional fields)..



IMPORTANT If you are on line and you click on either the Parameters or the DeviceLogix tab, you may be prompted to upload or download the device. When you are on line, the dialog checks the configuration in the device and compares it to the current configuration. If the configurations are not the same, you must upload from or download to the device to make the configurations the same before you can make changes. If you need to make changes without uploading or downloading, you can exit the dialog box, go off line and re-enter the dialog box to make the desired changes.

To start the DeviceLogix Editor for a DeviceLogix-enabled device, click **Start Logic Editor**. On the DeviceLogix Editor Style Selection dialog, you are prompted to select the editor type that you want to launch. After selecting an editor type, click **OK**.

If the current device does not support one of the editor types, that editor type will be grayed out.

IMPORTANT If you select an editor type for a particular device and that type is committed to the .dnt file (clicking **OK** or **Apply**), that editor style is registered. Therefore, you cannot switch to another editor style in that same .dnt file (the next time you launch the DeviceLogix Editor Style Selection dialog, the other editor style is grayed out). If you want to change the editor type (and a device supports both editor types), you must create a new project file, add this device again, and then select the other editor type.

For more information on the Function Block Editor, refer to Chapters 2 and 3. For more information on the Old Ladder Editor, refer to Chapters 4 and 5. For more information on the New Ladder Editor, refer to Chapters 6 and 7.

Notes:

Navigate the Function Block Editor Interface

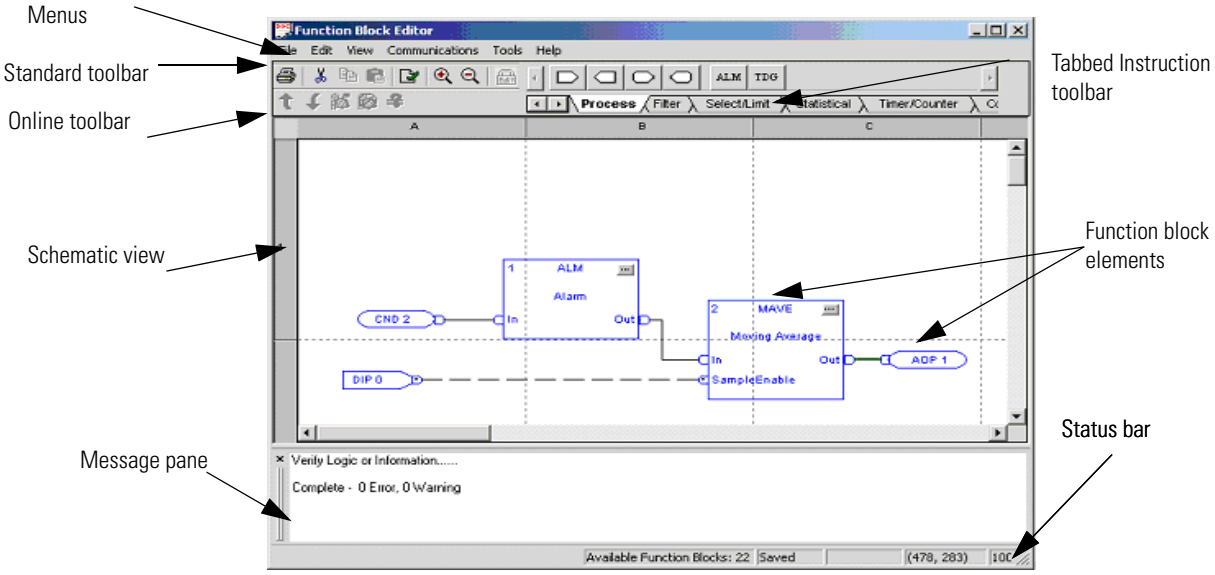
What This Chapter Contains

Read this chapter to learn more information about the Function Block Editor interface. The following table lists what this chapter contains and where to find specific information.

Components

To help you configure your logic, the Function Block Editor consists of:

- Function Block Elements
- Configuration toolbars
- Schematic view
- Message pane
- Status bar
- Menus



DeviceLogix Function Block Elements

Function Block elements consist of:

- I/O components: the input and output source of the product, or I/O information from the network
- Function block instructions: all types of DeviceLogix instructions. An instruction's I/O path needs to be bound with I/O components or the inputs and outputs of another instruction.
- Text comments

I/O Components

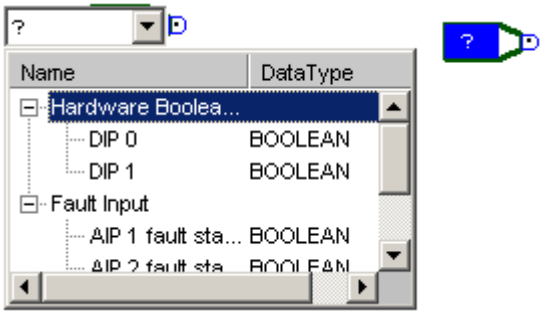
In the following sections, we will briefly describe each of the I/O components and include their graphic from the Function Block Editor.

You can drag each of these I/O components from the instruction toolbar, or click the icon and have it added into the current schematic, or select **Edit > Add Element** to add the I/O component. Each newly added I/O component does not have a binding name; you can click it to display a drop down list related to this component type, and then select the one you needed.

Digital Input Point (DIP)

The following kinds of digital inputs are supported:

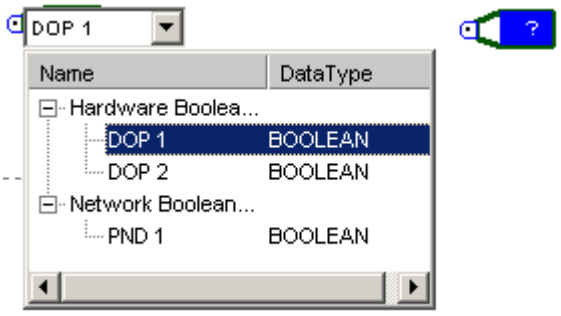
- physical local Boolean input point
- local Boolean fault status
- network Boolean input point
- local Boolean miscellaneous point



Digital Output Point (DOP)

The following kinds of digital outputs are supported:

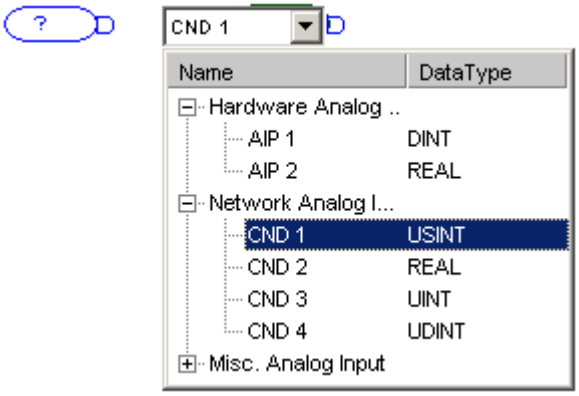
- physical local Boolean output point
- network Boolean output point



Analog Input Point (AIP)

The following kinds of analog inputs are supported:

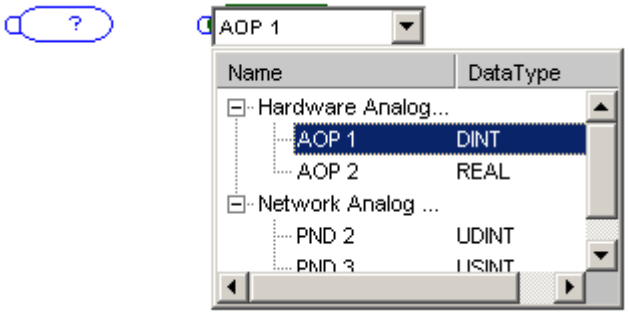
- physical local analog input point
- network analog input point
- local analog miscellaneous point



Analog Output Point (AOP)

The following kinds of analog outputs are supported:

- physical local analog output point
- network analog output point



Function Block Instructions

The DeviceLogix Function Block Editor has several categories of function block types:

- Process
- Filter
- Select/Limit
- Statistical
- Timer/Counter
- Compare
- Compute/Math
- Move/Logical
- Macro Block

Each function block type has the following tabs on its property pages:

- General tab - displays general information about this function block instruction. You can also select the function data type (if available) and input a comment for this block. Once any changes have been applied, a sequence number is allocated for this block.

- Parameter tab - Lists all of the parameters available for this function block type. Preset data can be entered in all editable fields. Once logic runs, the real-time value will be updated in the Value column. Note the read-only data is grayed out and cannot be edited.

Process Category

The Process category includes the following instruction types:

- Alarm
- Timing Diagnosis
- PID

Alarm

The Alarm function block initiates an alert based on the comparison between the input value and the threshold. The output of the DeviceLogix Alarm function block contains these alerts.

- High-High alarm
- High alarm
- Low alarm
- Low-Low alarm

The details of the Alarm function block are outlined in the table.

Condition	Output	Fault State
INPUT ≥ HHLimit (including the case when HLimit==HHLimit)	0x000C	0
HLimit ≤ INPUT < HHLimit	0x0004	0
LLimit < INPUT < HLimit	0x0000	0
LLLimit < INPUT ≤ LLimit	0x0002	0
INPUT ≤ LLLimit (including the case when LLimit==LLLimit)	0x0003	0
INPUT is NAN*	Keep the output unchanged	2
INPUT is positive infinity	0x000C	2
INPUT is negative infinity	0x0003	2
Input value from binding source is out of the object's range	Keep the output unchanged	1
*The condition is only possible when Operation Data Type is REAL.		

The valid parameter range is shown below:

Parameters	Data Range
HHLimit	-2147483648 ~ 2147483647 (DINT) -3.402823466e+38F ~ 3.402823466e+38F (REAL)
HLimit	-2147483648 ~ 2147483647 (DINT) -3.402823466e+38F ~ 3.402823466e+38F (REAL)
LLimit	-2147483648 ~ 2147483647 (DINT) -3.402823466e+38F ~ 3.402823466e+38F (REAL)
LLLimit	-2147483648 ~ 2147483647 (DINT) -3.402823466e+38F ~ 3.402823466e+38F (REAL)
Note: HHLimit >=HLimit >=LLimit >=LLimit	

IMPORTANT When data values are large, switching between REAL and DINT data types may cause a minimal loss of accuracy. For example, a value of 99999999 DINT will be rounded up to 100000000 REAL when you switch to a REAL data type and then back to a DINT data type.

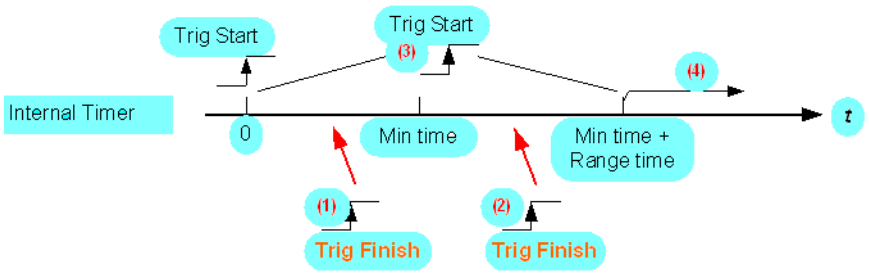
Timing Diagnosis

The DeviceLogix Timing Diagnosis function block object determines whether the occurrence of the expected event is within the preset timing interval.

The following is an operation example:

The rising edge of the Trig Start input indicates the occurrence of a Trig Start event. The rising edge of the Trig Finish input indicates the occurrence of a Trig Finish event.

When the Trig Start event occurs, the Timing Diagnosis function block is started. The internal timer starts timing from 0 as shown below. Meanwhile, the function block reports the triggered status as the output.



The events that occur in the illustration are as follows.

- If the Trig Finish event occurs before the Minimum Time, that is, at the time slot (1), then the function block returns an early finish status. If the occurrence of the Trig Finish event is within the range of Min time and Min time + Range time, as shown in time slot (2), then the Trig Finish event happens within the expected timing slot. Therefore, a normal finish status is returned. If no Trig Finish event occurs at the interval 0 and Min time + Range time, then the Late Finish status is returned, as shown in the time slot (4).
- Within the time interval 0 and Min time + Range time, if the Trig Start event occurs again, that is as shown in the time slot (3), then the Retrigger status is reported as the function block output.

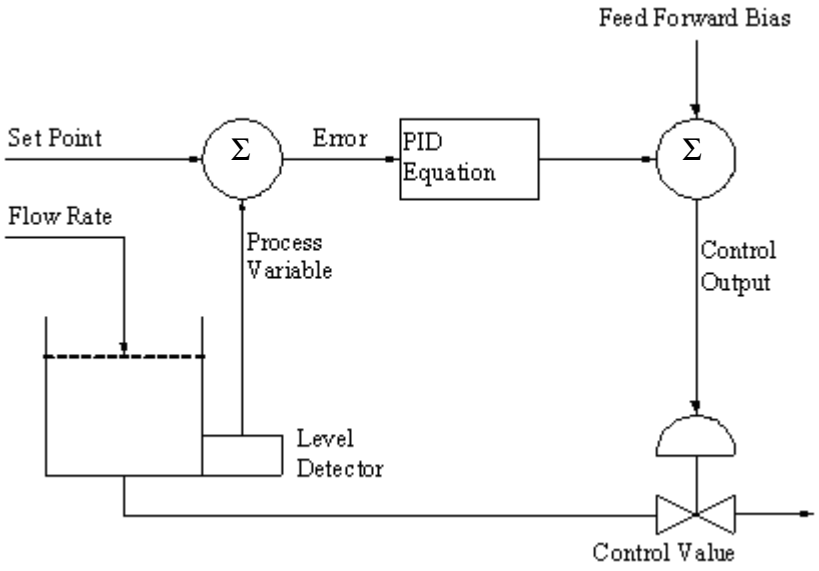
- Once the Trig Finish event occurs, or if a Retrigger event is detected, or the Late Finish status is reported, the function block stops operation, and the internal timer stops timing. The function requires a reset event to perform another operation.
- In all cases, the Reset signal overrides all other function block functionality. If the Reset Binding attribute is not supported or it is not bound, the function block behaves as if it is tied low.
- All input edges that occur during reset are ignored.
- The Elapsed Time attribute should be clear at the time that the Trig Start event triggers the function block.

Parameters	Data Range
MinTime	0 ~ 65535
RangeTime	0 ~ 65535
ElapsedTime	0 ~ 65535

PID

Use the PID function block to control a closed single analog loop.

The PID function block operates only in the timed mode. In this mode, the function block is calculated and updates its output periodically at a user-selectable rate. PID closed loop control holds a process variable at a desired set point. A flow rate/fluid level example is shown in the following figure.



The PID equation controls the process by sending an output to the actuator device. The greater the error between the setpoint and process variable input, the greater the output will be. An additional value (feedforward or bias) can be added to the control output as an offset. The PID result (control variable) drives the process variable toward the setpoint.

The PID function block monitors and controls the process loop for analog process parameters such as pressure, temperature, flow rate, and fluid level. Features of the PID function block include:

- PID equations expressed in Dependent Gains (ISA standard)
- Input scaling in engineering units
- Zero-crossing deadband
- Derivative term acts on PV
- Direct or reverse acting control
- Output alarms
- Output limiting with anti-reset windup
- Manual mode (with bumpless transfer)
- Feedforward or output biasing

The PID function block uses the following equation with dependent gains:

$$CV = K_c[(E) + \frac{1}{T_i}\int_0^t (E)dt + T_d\frac{d(PV)}{dt}] + Bias$$

Where:

K_c is Control Gain

T_i is Reset Term

T_d is Rate Term

SP is Set Point

PV is Process Variable

E is *(SP-PV)* or *(PV-SP)*

CV is Output Control Variable

Δt is Loop Update Time

The PID function has Enable In and Process Variable as inputs, and Enable Out and Control Variable as outputs, as described in the following tables.

Input	Type	Default	Description
Enable In	BOOL	1	Enable In
PV	REAL	0.0	Process Variable

Output	Type	Default	Description
Enable Out	BOOL	0	Enable Out
CV	REAL	0.0	Control Variable

The PID function also provides four parameters that you can modify as needed:

- Tuning
- Configuration
- Scaling
- Status

Tuning Parameters

Tuning Parameters allow you to set the PID algorithm parameters.

Parameter	Type	Range	Default	Description
Setpoint (*SP*)	REAL		0.0	Desired control point of the process variable. It should be scaled in the engineering unit.
Set Output %	REAL	0.0 ~ 100.0	0.0	PID output for the manual set output mode. Use this value to prevent bumps in control when switching control mode back to automatic.
Output Bias %	REAL	0.0 ~ 100.0	0.0	Output bias percentage
Control Gain (K_c)	REAL		0.0	Controller gain
Reset Time (T_i)	REAL		0.0	Reset time
Rate Time (T_d)	REAL		0.0	Rate time
Station Mode	BOOL		0	Indicates the station mode: • 0 = Manual • 1= Automatic

Manual and Automatic Modes

The PID function block automatically provides bumpless transfer from manual mode to auto mode. The PID function block back-calculates the value of the integral accumulation term required to make the CV output track the set output value in manual mode. In this manner, when the loop switches to auto mode, the CV output starts off from the set output value and no "bump" in output value occurs.

Configuration Parameters

Configuration parameters allow you to set control loop features.

Parameter	Type	Range	Default	Description
Control Action	BOOL		0	Indicates the direction of control: • 0 is *E=SP-PV* • 1 is *E=PV-SP* This parameter cannot be configured when the DeviceLogix logic is in the Run mode.
Loop Update Time	UDINT		0	Periodical time interval in microseconds for output update. This parameter cannot be configured when the DeviceLogix logic is in the Run mode.
CV High Limit %	REAL	0.0 ~ 100.0	0.0	System's maximum allowable value for the Control Variable. The PID function block does not output a *CV* that exceeds the High Limit.
CV Low Limit %	REAL	0.0 ~ 100.0	0.0	Sytem's minimum allowable value for the Control Variable. The PID function block does not output a *CV* less than the Low Limit.
Deadband Value	REAL		0.0	Error range above and below the setpoint. Enter "0" to inhibit the deadband. The deadband has the same scaled units as the setpoint.

Deadband

The adjustable deadband is used to select an error range above and below the setpoint where output does not change as long as the error remains within this range. This deadband controls how closely the process variable matches the set point without changing the output.

Zero-crossing is deadband control that lets the function block use the error for computational purposes as the process variable crosses into the deadband until the process variable crosses the setpoint. Once the process variable crosses the setpoint (error crosses zero and changes sign), and as long as the process variable remains in the deadband, the function block considers the error value to be zero. The deadband has the same scaled units as the setpoint.

Output Limit

An output limit (percent of output) can be set on the control output. When the function block detects that the output has reached a limit, the PID function block automatically avoids reset windup by preventing the integral term from accumulating whenever the CV output reaches its maximum or minimum values. The accumulated integral term remains frozen until the CV output drops below its maximum limit or rises above its minimum limit. Normal integral accumulation automatically resumes.

Scaling Parameters

Scaling parameters allow you to set the output scale.

Parameter	Type	Default	Description
PV Max	REAL	0.0	Maximum value for the unscaled Process Variable (*PV*). This parameter cannot be configured when the DeviceLogix logic is in the Run mode.
PV Min	REAL	0.0	Minimum value for the unscaled Process Variable (*PV*). This parameter cannot be configured when the DeviceLogix logic is in the Run mode.
EU Max	REAL	0.0	Maximum engineering unit for the Process Variable (*PV*). This parameter cannot be configured when the DeviceLogix logic is in the Run mode.
EU Min	REAL	0.0	Minimum engineering unit for the Process Variable (*PV*). This parameter cannot be configured when the DeviceLogix logic is in the Run mode.
CV Max (at 100%)	REAL	0.0	Maximum value for the unscaled Control Variable (*CV*). This parameter cannot be configured when the DeviceLogix logic is in the Run mode.

CV Min (at 0%)	REAL	0.0	Minimum value for the unscaled Control Variable (*CV*). This parameter cannot be configured when the DeviceLogix logic is in the Run mode.
Positive Deviation Alarm Limit	REAL	0.0	High deviation alarm from the set point.
Negative Deviation Alarm Limit	REAL	0.0	Low deviation alarm from the set point.

The PID function block scales the process variable, using the following formula:

$$PV_{scaled} = \frac{(PV - PV_{Min})}{PV_{max} - PV_{min}}(EU_{max} - EU_{min}) + EU_{min}$$

The setpoint and zero-crossing deadband should be scaled to engineering units.

Status Parameters

Status parameters allow you to monitor the control algorithm running status, using the Alarm parameter's bit field as described here:

- Bit 0: CV is below minimum output limit (0=no; 1=yes)
- Bit 1: CV is above maximum output limit (0=no; 1=yes)
- Bit 2: Error is within deadband (0=no; 1=yes)
- Bit 3: Deviation is alarmed high (0=no; 1=yes)
- Bit 4: Deviation is alarmed low (0=no; 1=yes)
- Bit 5: SP is out of range (0=no; 1=yes)
- Bit 6: PV is out of range (0=no; 1=yes)
- Bit 7~15: Reserved

Filter Category

The Filter category includes a single instruction type: Low Pass Filter.

Low Pass Filter

The DeviceLogix Low Pass Filter (LPF) function block provides a filter to attenuate input frequency above the cutoff frequency.

Let W_{lag} stand for the value of the attribute WLag and T_{sample} is the value of the sampling period. Therefore, the sampling frequency f_s is:

$$f_s = \frac{1}{T_{sample}}$$

According to $W_{lag,}$ the expected analog cutoff frequency f_p is:

$$f_p = \frac{\omega_{lag}}{2\pi}$$

With the basic equation between the analog input frequency f and the digital frequency Ω

$$\Omega = 2\pi f / f_s \quad [1]$$

we can obtain the digital frequency Ω_p corresponding to f_p

$$\Omega_p = \omega_{lag} \times T_{sample}$$

The bilinear transformation method is applied to convert the analog filter into the digital filter. The bilinear transformation is defined as follows.

$$s \Leftrightarrow 2f_s \frac{z-1}{z+1}$$

To adjust the frequency shift due to the bilinear transformation, you must use the prewarping equation to calculate the prewarping analog frequency. The prewarping equation is as follows.

$$\omega_p = 2 \times f_s \times \tan(\frac{\Omega_p}{2})$$

Then the value of w_p is taken as the real analog cutoff frequency and substitute for w_{lag}..

The above discussion outlined the general process of low-pass filtering. Specifically, the filter discussed order 1 LPF and order 2 LPF.

Order 1 LPF

The target analog filter is as follows.

$$\frac{\omega_{beg}}{s+\omega_{beg}}$$

When bilinear transformation is applied, we take the w_p as the actual cutoff frequency. That is, the target transform function is as follows.

$$\frac{Y(s)}{X(s)}=\frac{\omega_p}{s+\omega_p}$$

Then we apply the bilinear transformation.

$$\frac{Y(z)}{X(z)}=\frac{\omega_p}{2f_s\frac{z-1}{z+1}+\omega_p}$$

Therefore, the corresponding difference equation is as follows.

$$y(n)=\frac{\omega_p}{2f_s+\omega_p}x(n)+\frac{\omega_p}{2f_s+\omega_p}x(n-1)+\frac{2f_s-\omega_p}{2f_s+\omega_p}y(n-1)$$

Order 2 LPF

The target analog filter is as follows.

$$\frac{\omega_{beg}^2}{s^2+\sqrt{2}\times\omega_{beg}\times s+\omega_{beg}^2}$$

This is the same derivation process that is in order 1 LPF. Finally, the corresponding difference equation is as follows.

$$y(n)=\frac{\omega_p^2}{A}x(n)+\frac{2\omega_p^2}{A}x(n-1)+\frac{\omega_p^2}{A}x(n-2)+\frac{8f_s^2-2\omega_p^2}{A}y(n-1)+\frac{2\sqrt{2}\omega_p f_s-4f_s^2-\omega_p^2}{A}y(n-2)$$

where

$$A=4f_s^2+2\sqrt{2}\omega_p f_s+\omega_p^2$$

The valid parameter range is shown below.

Parameters	Data Range
Initialize	0,1
WLag	0 < WLag = 3.402823466e+38F
Order	1,2
Flotation	0, 1, 2, 3
FaultOption	-3.402823466e+38F ~ 3.402823466e+38F

Select/Limit Category

The Select/Limit category includes the following instruction types:

- Select
- High Low Limit

Select

The DeviceLogix Select function block identifies one input within the selected two function block inputs as the output according to the value of the selector. The Select function block is outlined in detail below.

```
IF Selector = 1
    Output = value of Input 2
IF Selector = 0
    Output = value of Input 1
END
```

The valid data ranges are as follows.

Parameters	Data Range
In1	-2147483648 ~ 2147483647 (DINT) -3.402823466e+38F ~ 3.402823466e+38F (REAL)
In2	-2147483648 ~ 2147483647 (DINT) -3.402823466e+38F ~ 3.402823466e+38F (REAL)
SelectorIn	0, 1
FaultOption	0, 1, 2, 3
FaultStateValue	-2147483648 ~ 2147483647 (DINT) -3.402823466e+38F ~ 3.402823466e+38F (REAL)

IMPORTANT When data values are large, switching between REAL and DINT data types may cause a minimal loss of accuracy. For example, a value of 99999999 DINT will be rounded up to 100000000 REAL when you switch to a REAL data type and then back to a DINT data type.

High/Low Limit

The DeviceLogix High/Low Limit function block limits the input value within the specified data range.The High/Low Limit function block is outlined in detail below.

```
IF value of Input > HighLimit
    Output = HighLimit
ELSE IF value of Input < LowLimit
    Output = LowLimit
ELSE
    Output = Input
END
```

Condition	Output		Fault Status
	If Use Fault Checking is RESET	If Use Fault Checking is SET	
INPUT in NAN	NAN	Value of the Fault state value	1
INPUT is the Positive Infinity	HighLimit		0
INPUT is the Negative Infinity	LowLimit		0

The valid data ranges are as follows.

Parameters	Data Range
HighLimit	2147483648 ~ 2147483647 (DINT) -3.402823466e+38F ~ 3.402823466e+38F (REAL)
LowLimit	2147483648 ~ 2147483647 (DINT) -3.402823466e+38F ~ 3.402823466e+38F (REAL)
FaultOption	0, 1, 2, 3, 4, 5
FaultStateValue	-2147483648 ~ 2147483647 (DINT) -3.402823466e+38F ~ 3.402823466e+38F (REAL)
Note: HighLimit > LowLimit.	

IMPORTANT When data values are large, switching between REAL and DINT data types may cause a minimal loss of accuracy. For example, a value of 99999999 DINT will be rounded up to 100000000 REAL when you switch to a REAL data type and then back to a DINT data type.

Statistical Category

The Statistical category includes a single instruction type: Moving Average.

Moving Average

The DeviceLogix Moving Average function block calculates a time average value for the input signal.

When sampling is enabled, the object executes this formula as the main function.

$$Output_k = \sum_{n=k}^{n=k-NumberOfSamples+1} Input_n \quad / NumberOfSamples$$

When the object starts executing, the moving average is initialized, as shown in the example that follows.

Example

Number of samples = 3
Scan 1: Output1 = Input1
Scan 2: Output2 = (Input2+Input1)/2
Scan 3: Output3 = (Input3+Input2+Input1)/3

The table lists the conditions and corresponding output with special input values.

	Output			
Condition	**Fault Option = 1**	**Fault Option = 2**	**Fault Option = 3**	**Fault Code**
INPUT in NAN	NAN	Hold last output	Value of instance attribute Fault State Value	2
INPUT is the Positive Infinity	Positive Infinity			2
INPUT is the Negative Infinity	Negative Infinity			2
Input value from binding source is out of range for the object	The calculated result using the truncated input value			1

The valid data ranges are as follows.

Parameters	Data Range
SampleEnable	0, 1
NumberOfSamples	1 ~ 65535
SampleRate	0 ~ 65535
FaultOption	0, 1, 2, 3
FaultStateValue	-2147483648 ~ 2147483647 (DINT) -3.402823466e+38F ~ 3.402823466e+38F (REAL)

IMPORTANT When data values are large, switching between REAL and DINT data types may cause a minimal loss of accuracy. For example, a value of 99999999 DINT will be rounded up to 100000000 REAL when you switch to a REAL data type and then back to a DINT data type.

Timer/Counter Category

The Timer/Counter category includes the following instruction types:

- Timers - The Timer controls the state of a single output based on the value of an Accumulator and a Preset. The Timer has two inputs called Input and Reset which control the counting of the Accumulator. The three timers available are:
  - On-Delay Timer
  - Off-Delay Timer
  - Pulse Timer

  All three timers have a selectable time base of either 1 millisecond or 10 milliseconds. The timers are non-retentive. Only the output status is available. No Timer Timing or Timer Done bits are available. The different timers are described in the next sections.

- Counters - There are two kinds of Counters:
  - Up Counter
  - Up/Down Counter

  The difference is the Up/Down Counter has two input path: one for up counting and the other for down counting. The Reset path binding is optional for Counters. You can also change element type by editing the name field.

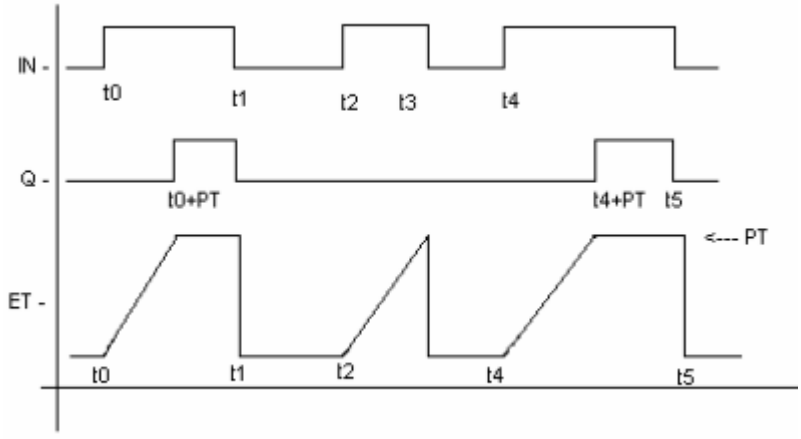
If the device supports retentive type Timer or Ccounter, then the accumulate value of the Timer/Counter will be retained during power-down. When the power is cycled and the logic resumes running, it updates based on the previous accumulate value.

On Delay Timer

The On Delay Timer delays the output response to an input by a desired amount of time. When the input is TRUE, the timer increments the accumulator and when the input goes FALSE the timer resets the accumulator. For each time base unit of time, the timer increments the accumulator. When the accumulator reaches the preset value, the timer sets the output to TRUE. The timer maintains the output TRUE status as long as the input remains TRUE.

When the input changes from TRUE to FALSE, the timer resets both the output and the accumulator. If the input goes FALSE before the time period specified by the time base and preset, the output remains FALSE and the accumulator is cleared, essentially ignoring the input.

If the timer senses a TRUE level on the reset input at any time during the operation of the timer, it resets the output to FALSE and clears the accumulator. Because the reset line is level sensitive, the timer remains reset until the timer detects a FALSE on the reset input. Also, because the input is level sensitive, the timer again begins to increment the accumulator if the reset line goes FALSE while the input remains TRUE.



The valid data ranges are as follows.

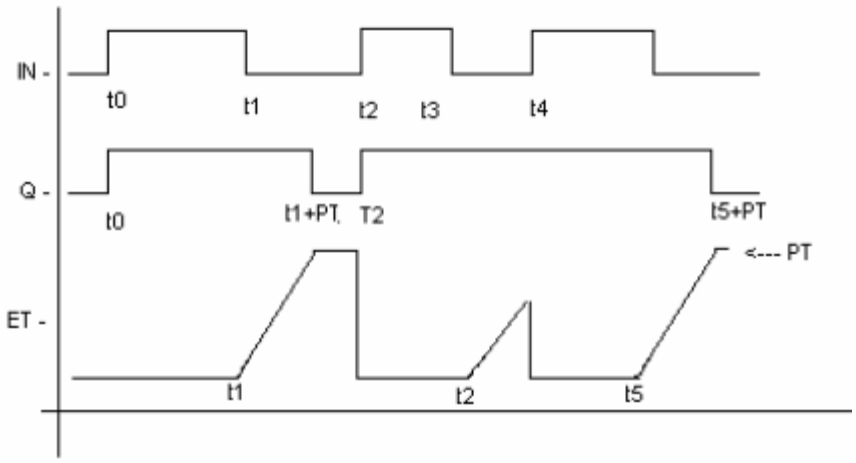
Parameters	Data Range
PRE	0 ~ 65535
ACC	0 ~ 65535
Time Base	0 (1 ms), 1 (10 ms)

Off Delay Timer

The Off-Delay Timer works the same way as the On-Delay Timer but instead of delaying the TRUE status of the output, it delays the FALSE status of the output. The input to this timer is a level sensitive FALSE with an edge-triggered reset on the FALSE to TRUE transition. This means that when the input is FALSE, the timer increments the accumulator and when the input goes TRUE the timer resets the accumulator. For each time base unit of time, the timer increments the accumulator.

When the accumulator reaches the preset value, the timer sets the output to FALSE. The timer maintains the output FALSE status as long as the input remains FALSE. When the input changes from FALSE to TRUE, the timer sets the output to TRUE and resets the accumulator. If the input goes TRUE before the time period specified by the time base and preset, the output remains TRUE, and the accumulator is cleared, essentially ignoring the input.

If the timer senses a TRUE level on the reset input at any time during the operation of the timer, it resets the output to FALSE and clears the accumulator. Because the reset line is level sensitive, the timer remains reset until the timer detects a FALSE on the reset input. Also, because the input is level sensitive, the timer again begins to increment the accumulator if the reset line goes FALSE while the input remains FALSE. However, because the reset logic already set the output to FALSE, the time delay causes no effect because the output is already FALSE. If the input is TRUE when the reset goes FALSE, the timer sets the output to TRUE.



The valid data ranges are as follows.

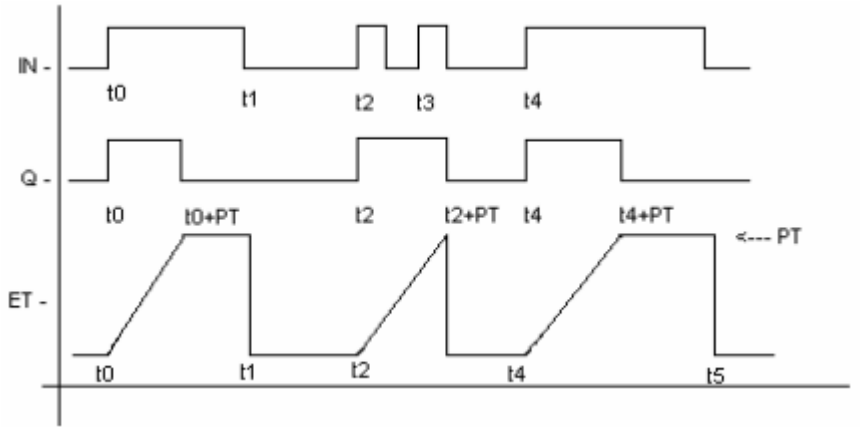
Parameters	Data Range
PRE	0 ~ 65535
ACC	0 ~ 65535
Time Base	0 (1 ms), 1 (10 ms)

Pulse Timer

The Pulse Timer generates a TRUE value on its output for a fixed amount of time. The duration of the TRUE pulse is determined by the preset value along with the time base of the timer. When the input to the timer changes from FALSE to TRUE, it sets the output to TRUE and starts the accumulator counting. It then increments the accumulator each time the time base number of milliseconds has expired. When the accumulator reaches the preset value, the timer resets the output to FALSE.

In the Pulse Timer, the input acts only as a trigger to start the accumulator counting. Once the accumulator starts timing, it continues to rise regardless of the state of the input. As long as the output is TRUE and the accumulator is counting, additional triggers of the input do not affect the state of the output or the count of the accumulator. Once the accumulator reaches the preset value and the timer resets the output to FALSE, the Pulse Timer can again trigger the process by sensing a FALSE to TRUE transition on the input pin. Even if the input remains on the entire time the accumulator is counting, when the accumulator reaches the preset value, the timer resets the output to FALSE.

At any point during the operation of the timer, if it detects a TRUE level on the reset input, it will disable the timer and set the output to FALSE. The timer must again be triggered by a FALSE to TRUE transition on the input. This means that if the input is TRUE and the timer is reset, the timer will remain inactive. Even if the reset changes back to FALSE while the input is TRUE, the timer remains inactive. In order to start a new pulse operation, the input must change to FALSE and then back to TRUE.



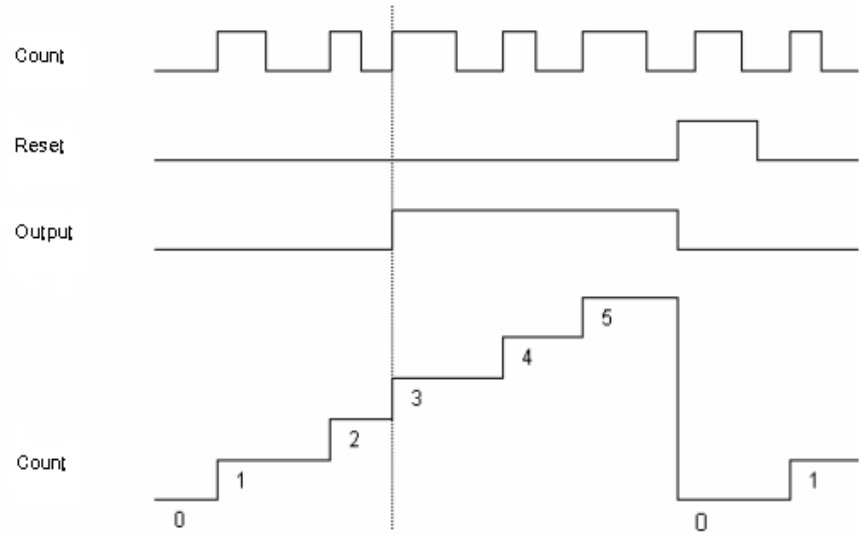
The valid data ranges are as follows.

Parameters	Data Range
PRE	0 ~ 65535
ACC	0 ~ 65535
Time Base	0 (1 ms), 1 (10 ms)

Up Counter

The Up Counter has two inputs called Input and Reset and one output called Output. The Up counter simply counts up on a FALSE to TRUE transition and sets its output to TRUE when the accumulator reaches the preset value. The accumulator continues to count up until the counter is reset or it reaches 65,535. A TRUE value on the reset input zeros the accumulator and sets the output to FALSE.

An illustration of the counter function with a preset value set to 3 is below.



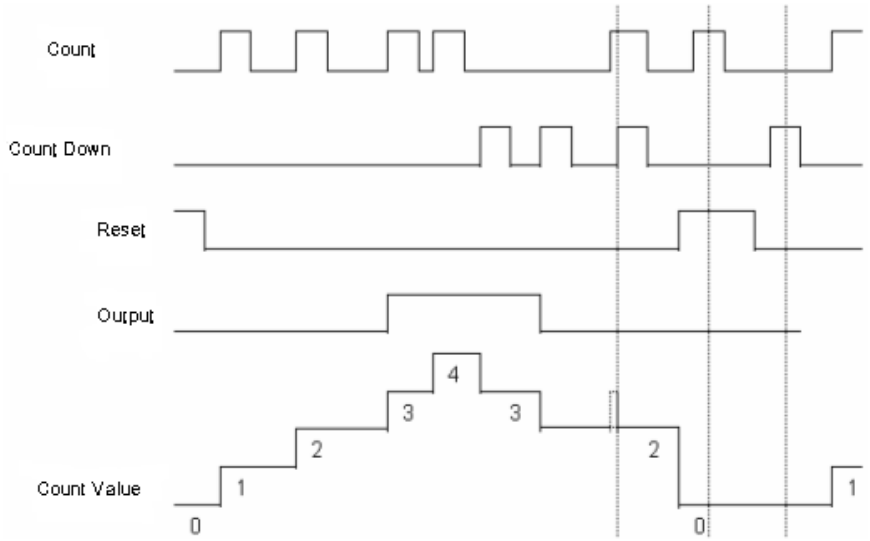
The valid data ranges are as follows.

Parameters	Data Range
PRE	0 ~ 65535
ACC	0 ~ 65535

Up/Down Counter

The Up/Down counter has three inputs called Input, Reset, and Count Down Input and one output called Output. The counter increments the accumulator any time the Input changes from FALSE to TRUE and decrements the counter any time the Count Down Input changes from FALSE to TRUE. When the accumulator is above or equal to the preset value, the counter sets its output to TRUE. When the accumulator falls below the preset, the counter resets its output to FALSE. Like the Up Counter, the Up Down Counter resets when a TRUE level is detected on the Reset input. When a reset occurs, the counter zeros the accumulator and sets the output to FALSE.

An illustration of the Up Down counter function with a preset value set to 3 is below:



The valid data ranges are as follows.

Parameters	Data Range
PRE	0 ~ 65535
ACC	0 ~ 65535

Compare Category

The Compare category includes basic comparison functions, including the the instruction types:

- Greater Than (GRT)
- Greater Than or Equal To (GEQ)
- Equal (EQU)
- Not Equal (NEQ)
- Less Than (LES)
- Less Than or Equal (LEQ)
- Mask (MEQ)

Operation rules [all except Mask (MEQ)]

The conditions and outputs of each function block are described below.

Function Block Type	Condition	Output
Greater Than (GRT)	Source A > Source B	1
	Source A <= Source B	0
Greater Than or Equal (GEQ)	Source A >= Source B	1
	Source A < Source B	0
Equal (EQU)	Source A == Source B	1
	Source A != Source B	0
Not Equal (NEQ)	Source A != Source B	1
	Source A == Source B	0
Less Than (LES)	Source A < Source B	1
	Source A >= Source B	0
Less Than or Equal (LEQ)	Source A <= Source B	1
	Source A > Source B	0

Data ranges

The valid data ranges for all of these instructions are as follows.

Parameters	Data Range
Source A	-2147483648 ~ 2147483647 (DINT) -3.402823466e+38F ~ 3.402823466e+38F (REAL)
Source B	-2147483648 ~ 2147483647 (DINT) -3.402823466e+38F ~ 3.402823466e+38F (REAL)
FaultOption	0, 1, 2

IMPORTANT When data values are large, switching between REAL and DINT data types may cause a minimal loss of accuracy. For example, a value of 99999999 DINT will be rounded up to 100000000 REAL when you switch to a REAL data type and then back to a DINT data type.

Comparison output for Infinity Input

Value of Source A	Value of Source B	Comparison Output					
		GRT	GEQ	EQU	NEQ	LES	LEQ
(+) Infinity	(+) Infinity	0	1	1	0	0	1
(+) Infinity	(-) Infinity	1	1	0	1	0	0
(+) Infinity	Any finite number	1	1	0	1	0	0
(-) Infinity	(+) Infinity	0	0	0	1	1	1
(-) Infinity	(-) Infinity	0	1	1	0	0	1
(-) Infinity	Any finite number	0	0	0	1	1	1
Any finite number	(+) Infinity	0	0	0	1	1	1
Any finite number	(-) Infinity	1	1	0	1	0	0
Any finite number	Any finite number	Refer to operation rules					

Operation rules [Mask (MEQ) only]

The Mask process is outlined below.

```
IF (Source AND Mask) == (Compare AND Mask)
    Output is Set
ELSE
    Output is Clear
END
```

Data ranges [Mask (MEQ) only]

The valid data ranges for the MEQ instructions are as follows.

Parameters	Data Range
Source	16#0 ~ 16#FFFFFFFF
Mask	16#0 ~ 16#FFFFFFFF
Compare	16#0 ~ 16#FFFFFFFF

Compute/Math Category

The Compute/Math category includes fundamental arithmetic operations, including the instructions types:

- Add (ADD)
- Multiply (MUL)
- Subtract (SUB)
- Divide (DIV)
- Modulus (DINT)
- Modulus (REAL)
- Absolute (ABS)
- Negative (NEG)
- Square Root (SQR)
- Power (XPY)

Operation rules

The conditions and outputs of each function block are described below.

Function Block Type	Output
Add (ADD)	= Source A + Source B
Multiply (MUL)	= Source A x Source B
Subtract (SUB)	= Source A - Source B
Divide (DIV)	= Source A / Source B
Modulus (DINT)	= Source A - (Source A/ Source B) x Source B
Modulus (REAL)	= Source A - (DINT)(Source A/ Source B) x Source B
Absolute (ABS)	= Absolute value of (Source A)
Negative (NEG)	= - Source A
Square Root (SQR)	= Square root of (Source A) *
Power (XPY)	= Source A * * Source B
* If Source A is negative, the operation takes the absolute value of the Source A before calculating the square root and no fault is reported.	

Fault State Conditions and Rules

Fault state conditions and rules are listed below.

Operation Data Type	Condition	Output			Fault Code
		Fault Option = 1	**Fault Option = 2**	**Fault Option = 3**	
DINT	The result from the operations ADD, SUB or MUL exceeds the range of the DINT data type	Truncates	Hold last output	Value of instance attribute Fault State value	3
	(Any DINT) / 0	= Dividend			3
	Input data from the binding path is out of range	Use the Truncates input value			1
REAL	(±) Infinity x (±) Infinity	(±) Infinity	Hold last output	Value of instance attribute Fault State value	3
	(±) Nonzero / 0	(±) Infinity			3
	Infinity + Infinity	Infinity			3
	0 / 0	Infinity			3
	Infinity - Infinity	NAN			3
	(±) Infinity / (±) Infinity	NAN			3
	(±) Infinity x 0	NAN			3
	Sqrt ((±) Infinity)	Infinity			3
	0 x x 0	NAN			3
	In XPY operation, source A is negative while source B is not an integer value	NAN			3
	NAN operand for any operation	NAN			3
	Input data from the binding path is out of range	Use the Truncates input value			1

Data ranges

The valid data ranges for the compute instructions are as follows.

Parameters	Data Range
SourceA	-2147483648 ~ 2147483647 (DINT) -3.402823466e+38F ~ 3.402823466e+38F (REAL)
Source B	-2147483648 ~ 2147483647 (DINT) -3.402823466e+38F ~ 3.402823466e+38F (REAL)
FaultOption	0, 1, 2, 3
FaultStateValue	-2147483648 ~ 2147483647 (DINT) -3.402823466e+38F ~ 3.402823466e+38F (REAL)

IMPORTANT When data values are large, switching between REAL and DINT data types may cause a minimal loss of accuracy. For example, a value of 99999999 DINT will be rounded up to 100000000 REAL when you switch to a REAL data type and then back to a DINT data type.

Move/Logical Category

The Move/Logical category includes fundamental bit type logic operations, including the instructions types:

- Boolean
  - AND (BAND)
  - Not AND (BNAND)
  - OR (BOR)
  - Not OR (BNOR)
  - Exclusive OR (BXOR)
  - Exclusive Not OR (BXNOR)
  - NOT (BNOT)
- Latch
  - Set Latch (SETD)
  - Reset Latch (RESD)

Boolean Functions

The Boolean functions are as follows. Note that the number of inputs can be changed on some functions and is product-specific.

Function Block Type	Set	Reset	Output Value
AND (BAND	0	0	0
	0	1	0
	1	0	0
	1	1	1
Not AND (BNAND)	0	0	1
	0	1	1
	1	0	1
	1	1	0
OR (BOR)	0	0	0
	0	1	1
	1	0	1
	1	1	1
Not OR (BNOR)	0	0	1
	0	1	0
	1	0	0
	1	1	0
Exclusive OR (BXOR)	0	0	0
	0	1	1
	1	0	1
	1	1	0
Exclusive Not OR (BXNOR)	0	0	1
	0	1	0
	1	0	0
	1	1	1
NOT (BNOT) *	0	N/A	1
	1	N/A	0

Note: The Input number can be set in a property page; the default number is 2. BNOT has only one input.

Latch Functions

There are two kinds of Latch functions: Set Dominant Latch (SR) and Reset Dominant Latch (RS). Unlike the Latch in RS Logix 5000, DeviceLogix Latch requires the Reset binding. So the Set/Reset appears in pair and the RS type and SR type have different element order. See its truth table below.

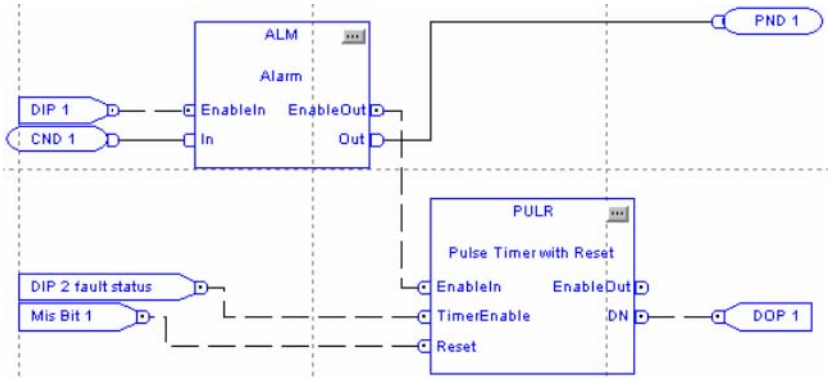
Function Block Type	Input 1	Input 2	Value at time + t_0	Value at time + t_{0+1}
SETD (Set Dominant)	0	0	0	0
	0	1	0	1
	1	0	0	0
	1	1	0	1
	0	0	1	1
	0	1	1	1
	1	0	1	0
	1	1	1	1
RS (Reset Dominant)	0	0	0	0
	0	1	0	1
	1	0	0	0
	1	1	0	0
	0	0	1	1
	0	1	1	1
	1	0	1	0
	1	1	1	0

Enable Line Feature

Enable Line can be supported by each instruction type in the Function Block Editor. When an instruction supports the Enable Line feature, that particular instruction can only be executed when the Enable Line feature is on; otherwise, that instruction maintains the data from the last data scan. Each instruction that uses Enable Line must configure both an input and an output; the output has the same real time data as the input, passing the enable information to the next instruction.

The Enable Line feature can have two data sources:

- If an input uses Enable Line to bind with an input point or other function block's output, the input retains this point's data value.
- If an input uses Enable Line in an unbound capacity, the input uses the default constant value (which you can set on each instruction's property page).



TIP The EDS file for a device must explicitly specify support for this functionality to be available. If a device's EDS file does not support this feature, this functionality will not be available.

Configuration Toolbars

The Function Block Editor toolbars are:

- Standard - Lets you perform general editing functions, verify logic, and toggle edit mode.
- Tabbed Instruction - Lets you enter Boolean or Analog type I/O, and add functional elements to the schematic.
- Online - Lets you perform online functions. This toolbar is enabled only when you are working on line.

Standard Toolbar

Standard Toolbar

Function	Description
	Sends the logic schematic to a printer.
	Removes the selected element.
	Duplicates the selected element and send it to the clipboard.
	Places the element in the clipboard at current cursor position.
	Verifies the logic you have configured on the schematic sheet.
	Makes the configuration smaller.
	Makes the configuration bigger.
	Toggles between the DeviceLogix Function Block Editor software being in the Edit mode or not being in the Edit mode. In the Edit mode, you can modify the logic but you cannot communicate with the device. That is, you cannot upload or download logic or turn the logic On or Off. When you are not in Edit mode, you can perform online animation, if the device is online. You can set the preset value, force I/O, or accumulator value of counters or timers, and download to a device, but you cannot modify logic.

Tabbed Instruction Toolbars

The tabbed instruction toolbars are:

- Process
- Filter
- Select/Limit
- Statistical
- Timer/Counter
- Compare
- Compute/Math
- Move/Logical
- Macro Block

Independent of what instruction type you have selected, the following I/O components toolbar icons are always available.

I/O Components Toolbar

Function	Description
	Boolean bit input
	Boolean bit output
	Analog bit input
	Analog bit output

Process Category Toolbar

Process Category Toolbar

Function	Description
ALM	Alarm Instruction
TDG	Timing Diagnosis Instruction
PID	PID Instruction

Filter Category Toolbar

Filter Category Toolbar

Function	Description
LPF	Low Pass Filter Instruction

Select/Limit Category Toolbar

Select/Limit Category Toolbar

Function	Description
SEL	Select Instruction
HLL	High Low Limit Instruction
LPF	Low Pass Filter Instruction

Statistical Category Toolbar

Statistical Category Toolbar

Function	Description
MAVE	Moving Average Instruction

Timer/Counter Category Toolbar

Timer/Counter Category Toolbar

Function	Description
PULR	Pulse Timer Instruction
TONR	On-Delay Timer Instruction
TOFR	Off-Delay Timer Instruction
CTU	Up Counter Instruction
CTUD	Up Down Counter Instruction

TIP When using the ACC feature, you may notice some minor differences in the Timer/Counter instructions. Depending on the firmware implementation, some devices consider this an output and bind it to an analog output tag, while other devices consider it a parameter.

Compare Category Toolbar

Compare Category Toolbar

Function	Description
GRT	Greater Than Instruction
GEQ	Greater Than or Equal To Instruction
EQU	Equal Instruction
NEQ	Not Equal Instruction
LES	Less Than Instruction
LEQ	Less Than or Equal Instruction
MEQ	Mask Instruction

Compute/Math Category Toolbar

Compute/Math Category Toolbar

Function	Description
ADD	Add Instruction
MUL	Multiply Instruction
SUB	Subtract Instruction
DIV	Divide Instruction
MOD	Modulus (DINT) Instruction
MOD	Modulus (REAL) Instruction
ABS	Absolute Instruction
NEG	Negative Instruction
SQR	Square Root Instruction
EXP	Power Instruction

Move/Logical Category Toolbar

Move/Logical Category Toolbar

Function	Description
BAND	AND Instruction
BNAND	Not AND Instruction
BOR	OR Instruction
BNOR	Not OR Instruction
BXOR	Exclusive OR Instruction
BXNOR	Exclusive Not OR Instruction
BNOT	NOT Instruction
SETD	Set Latch Instruction
RSTD	Reset Latch Instruction

Macro Block Category Toolbar

Beginning with DeviceLogix firmware revision 4, certain devices support Macro function block instructions. Once you define a Macro instruction, a new tabbed instruction toolbar appears. For more information on Macro function block instructions, refer to Chapter 3, Bind Function Blocks with I/O.

Online Toolbar

Use the Online toolbar to perform functions when you are working on line in the Function Block Editor.

Online Toolbar

Function	Description
	Upload the logic configuration from the device to the Function Block Editor configuration tool.
	Download the logic configuration from the Function Block Editor to the device. The configuration must pass the logic verification process for the download to be successful.
	Run the DeviceLogix logic configuration that you have downloaded to the device.
	Stop the DeviceLogix logic configuration running in the device.
	Compare the logic in the device with that in the Function Block Editor configuration tool.

Schematic View

The schematic sheet is the area in which you place function blocks to create logic. The area is laid out on a grid with letters A through L representing the columns and numbers 1 through 12 representing the rows. This area is large enough to easily place all the function blocks, I/O tags, and associated connecting wires needed to create the desired logic.

TIP The zoom level controls how much of the schematic you see at any given time. If you want to see more of the schematic, zoom out. If you want more detail, zoom in. You can also use the scroll bars to move the schematic sheet around to display parts of the schematic that do not fit on the display.

The printed schematic uses 12 size A sheets of paper. The editor prints the schematic on four rows of three sheets. The schematic is printed at full size, regardless of the setting of the current zoom level. To make adjustments to the schematic before printing, you can use the print preview feature to see how the schematic will print. The schematic can also be sent to a plotter for easier viewing.

TIP Because a device has limited memory to store logic, the display of uploaded information may not match the appearance of information downloaded to the device. Therefore, once your logic configuration is complete, you should print a copy of it for your records.

Message Pane

The message pane on the bottom of the Function Block Editor displays the results when you verify logic. If the Function Block Editor finds an error in your logic, click on the error or warning message in the message pane and the cursor goes to the place in the Function Block Editor where the error is in the logic.

The message pane also displays the number of function blocks that are remaining in the device.

From the View menu, you can toggle between displaying the message pane or not displaying it.

Status Bar

The Status bar, located on the bottom of the Function Block Editor, provides a view of the current working status of the Function Block Editor. You can toggle between displaying the Status Bar and not displaying it by selecting **View > Status Bar**.

There are six panes that provide unique information about the Function Block Editor.

Status Bar

Pane	Description
Help	Indicates how to launch the online help.
Capacity	Displays the number of function blocks you may add to the schematic before maximum capacity has been reached.
Schematic Saved	Indicates whether changes to the schematic have been saved to the RSNetWorx for DeviceNet software (*.dnt) file. If no edits occurred, then Saved appears. If edits did occur, then Not Saved appears.
Schematic Matched	Indicates whether the schematic matches the configuration in the device. Displays Pending Edit if changes exist, Animated if there is a match (the schematic reflects the live status of the running logic), or is blank if you are offline.
Coordinates	Displays the x and y coordinates of the cursor on the schematic page.
Zoom	Displays the current zoom percentage (by default, 100%).

Menus

The Function Block Editor has six menu options that allow you to create and maintain a schematic:

- File
- Edit
- View
- Communications
- Tools
- Help

File Menu

The File menu lets you perform printing functions and lets you exit the DeviceLogix Function Block Editor .

File Menu

Function	Description
Print	Sends the logic schematic to the printer. All vital components (blocks, comments, and I/O points) are not spilt across printed pages.
Print Preview	Preview the logic schematic before sending it to the printer.
Print Setup	Choose printer and printing options.
Close	Exit the DeviceLogix Function Block Editor and return to RSNetWorx for DeviceNet software.

Edit Menu

The Edit menu lets you modify elements in the schematic. .

Edit Menu

Function	Description
Undo	Cancels the last action.
Redo	Performs again the previously cancelled action.
Cut[1]	Removes the selected function block element.
Copy[1]	Duplicates the selected content to save it in the clipboard.
Paste[1]	Places the content in the clipboard into the Function Block Editor at the position that your cursor is resting.
Delete	Permanently removes the selected content from the schematic.
Add Element	Adds the element at the cursor position. The List of Elements dialog box opens for you to select an element type.
Select All	Highlights all the elements in the Function Block Editor (to copy or cut).
Properties	Opens the Configuration Properties dialog box to modify properties, such as Author, Revision, and Description.
Recovery Mode	Opens the Recovery Mode dialog box to determine if logic is automatically enabled following a module replacement. When you use an Allen-Bradley master, choose the recovery mode option to determine how the I/O device recovers from an automatic download by a master device. When an I/O device fails and a new device is added to replace it, a master device on the network can automatically download the stored configuration and logic to the device. This feature is known as Auto Device Replace (ADR). The recovery mode determines whether or not to enable the local logic when a download of this type occurs. The recovery mode is disabled by default. ADR in the scanner is also disabled by default.

[1] You can cut/copy/paste across multiple instances of the Function Block Editor when the source version and the target version of the Function Block Editor are the same and the target device supports the element that is being copied.

View Menu

The View menu lets you modify your view of the schematic in the Function Block Editor.

View Menu

Function	Description
Toolbars	Opens the Toolbars dialog box to determine which toolbars display in the DeviceLogix Function Block Editor software.
Status Bar	Toggles between displaying and not displaying the status bar at the bottom of the Function Block Editor.
Message Log Window	Toggles between displaying and not displaying the Message Log window at the bottom of the Function Block Editor.
Zoom In	Increases (make larger) the view of the configuration. You lose view of part of the schematic of the Function Block Editor when you use this function.
Zoom Out	Decreases (make smaller) the view of the configuration. You see more of the schematic of the Function Block Editor when you use this function.
Fit to Page	Displays the entire logic diagram schematic within the application window. This causes the logic elements to appear very small, but displays the entire schematic.
Zoom to 100%	Causes the configuration pane to mirror what is seen on the printed version when the configuration pane is printed.

Communication Menu

The Communications menu lets you work on line in the Function Block Editor.

Communication Menu

Function	Description
Upload	Copies the configuration in the DeviceLogix-enabled device and displays it in the Function Block Editor. You lose any changes you made in the configuration and online animation starts.
Download	Transfers the configuration in the Function Block Editor to the DeviceLogix-enabled device and online animation starts. For the download to begin, the configuration must pass the verification process. After the download completes, you have the choice to enable the logic or leave the logic disabled.
Logic Enable On	Runs the logic configuration that you have downloaded to the device.
Logic Enable Off	Stops running the logic configuration that you downloaded to the device.

Tools Menu

The Tools menu lets you modify logic in the Ladder Editor.

Tools Menu

Function	Description
Logic Verify	Checks for mistakes in a schematic. Confirms that: • each function block has the minimum number of pins bound • each attribute has the correct data type associated with it • the two ends of a binding are of the same type
Compare	Compares the logic configured in the device with the logic configured in the Function Block Editor. You must be working in the online mode to use this function.
Edit Mode	Toggles between working in the Edit mode and not working in the Edit mode. In the Edit mode, it is possible to modify the logic but you cannot communicate with the device. (That is, you cannot upload or download logic or use the Logic Enable On or Logic Enable Off functions.) When you are not working in the Edit mode, it is possible to perform online animation. If you are working in online mode, you can set the preset value, force I/O, or accumulator value of counters or timers, but you cannot modify logic.
Resource	Displays the total function block resources and currently available resources. Use the Screen Format Resource and Download Option section to indicate what screen elements (such as comments or description) you want to download to the device.
Macro Block	Allows you to create and manage Macro types within your logic configuration through several sub-menu options: • Create Macro Block • Macro Block Manager • Open Macro Block Definition • Delete Macro Block • Open Macro Block Logic

Help Menu

The help menu provides you with assistance when you are working in the Function Block Editor.

Help Menu

Function	Description
Help Topics	Accesses the help files available for the Function Block Editor.
Release Notes	Accesses release notes pertaining to DeviceLogix functionality or the Function Block Editor.
About DeviceLogix	Opens the About Function Block Editor window to learn revision and copyright information about the Function Block Editor. File revision identifies the revision of the Function Block Editor DLL application.

Bind Function Blocks with I/O

What This Chapter Contains

Read this chapter to learn more information about the binding function blocks with I/O. The following table lists what this chapter contains and where to find specific information.

TIP For more information on the tasks you can perform in the Function Block Editor, see the Function Block Editor online help.

Overview of Inputs and Outputs

Function blocks contain both inputs and outputs. Inputs to function blocks can be attached to any of the input types or to the output of another function block. Only one connection can be made to one function block input pin and an input cannot be tied to another input. Outputs from function blocks can be attached to either hardware or network outputs or to the inputs of other function blocks. A single function block output can be the source of (and connected to) any number of hardware outputs or function block inputs. Hardware outputs cannot be tied to other hardware outputs, and hardware inputs cannot be tied to other hardware inputs.

Inputs

Device inputs can connect to any function block input or can drive an output directly. The number and type of inputs varies from device to device. However, devices may support the following five types of inputs (supported categories based on device) and are described below.

Input Category	Description
Hardware (physical)	Hardware inputs represent the actual inputs (such as sensors and switches) attached to a particular device.
Network	Network inputs represent data sent from a master that can be used in the device's logic.
Status	Status inputs indicate the state of the device. For example, if an explicit message connection exists between the device and a master, an 'Explicit connection exists' input would be set to true, possibly impacting the logic that the device performs.
Fault	Fault inputs are conditions that report device errors. For example, if a device detects a short circuit on an output, it can set a fault input to true. The fault input can then impact the logic that the device performs. **Note:** Currently, faults can be of the Boolean type only which indicates a device fault status, while the other input types can be Boolean or Analog.
Miscellaneous	Miscellaneous inputs reflect a status or a condition that is specific to each individual product.

Outputs

DeviceLogix device outputs (either Boolean or Analog) can connect to the output of any function blocks or can be driven directly by an input. There are two types of outputs described below.

Output Category	Description
Hardware	Hardware outputs are the actual outputs (such as lights and actuators) attached to a particular device. Without DeviceLogix functionality, the master would normally control the outputs via consumed data. In fact, if there is no local logic controlling an output, the master controls the output as it would if DeviceLogix functionality were not running on the device. However, within DeviceLogix functionality, if the local logic controls an output, the master no longer controls the output. The only way the master can affect the state of an output (under local control) is to route requests to the local logic by using network inputs. Some outputs can be under local control, while others can still be controlled by the master.
Network	Network outputs report the results of the local logic to a master. These outputs can be attached to any output point on any function block, and are part of the produced data from the device. They can also be connected to status, faults, or inputs.

Connect I/O points and function block instructions

Once you have I/O points and function blocks represented in your workspace, you want to connect them to actually create the flow of the logic. You can:

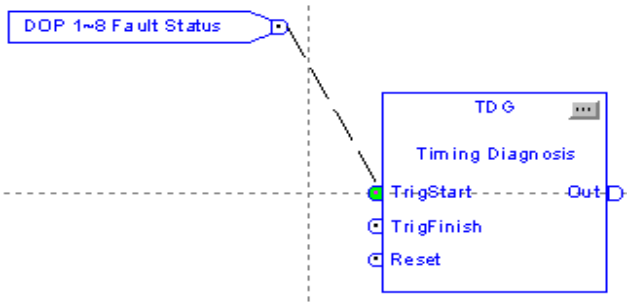
- bind the function block's input to input components
- bind the input component directly to output component
- bind the function block's output to function block's input

Connection rules

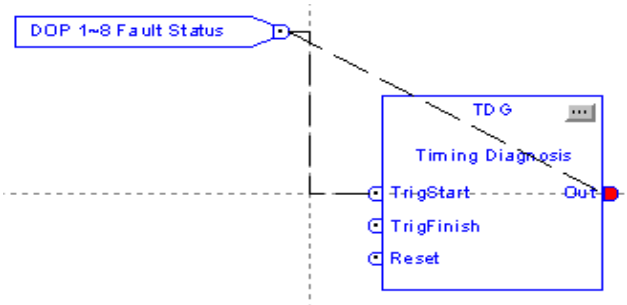
When making connections, keep the following rules in mind:

- You can attach a function block output pin to any function block input pin. This gives you feedback capability.
- You can tie a hardware input or any other input pin directly to a hardware or network output pin.
- You cannot tie hardware or network output connection pin to a function block input pin.
- You cannot attach a function block output pin to any input pin.

- A green circle indicates that the connection is valid.



- A red circle indicates that the connection is invalid.



Determine the status of a connection

Once a connection is made, you can use the Negate and Assume Data Available options to determine the state of that connection.

Negate Data

The Negate function causes the status of the data to be negated before it enters the function block. The negate operation cannot be used for analog types.

TIP Using the Negate option does not require the use of one of the available function blocks (as would a Not function block) .

Set Assume Data Available

The **Assume Data Available** feature should be used when feedback paths are involved in your logic; this feature assists the Function Block Editor in determining the function block process order. When feedback paths are used, the Function Block Editor's execution order algorithm may be unable to determine which function block needs to be resolved first. In this case, you must specify which function block will be evaluated first by designating that the Function Block Editor should assume that data is available at a specific input. Once selected, a double-headed arrow appears at the end of the wire indicating that it has precedence in the logic.

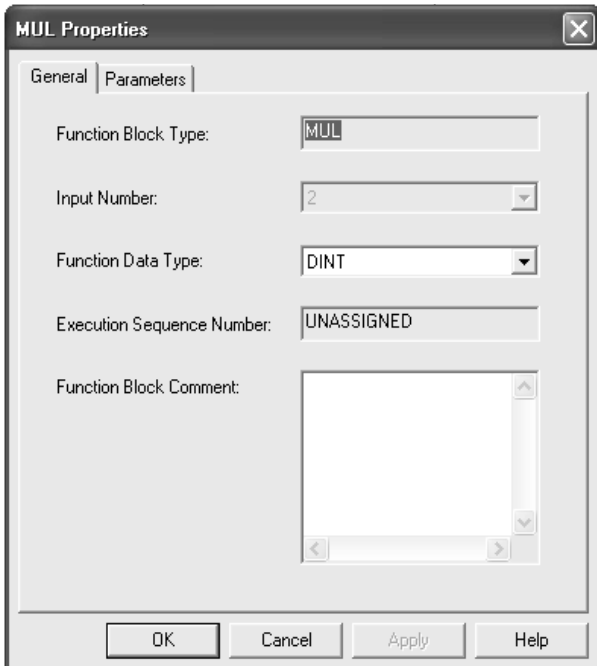
TIP Following a verify operation, the software may indicate that it could not resolve the execution order. In this case, you just use the Assume Data Available feature to resolve the conflict.

Function Block Properties and Parameters

To open a function block's property page, double-click the function block on the schematic page.

General tab

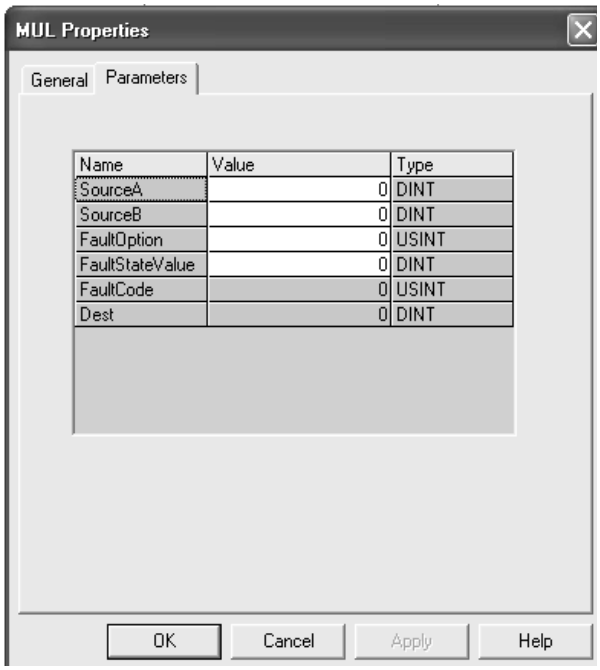
The following is an example General tab:



On this tab, the following fields are available:

Field	Description
Function Block Type	Displays the current function block type (read only).
Input Number	Displays the number of available inputs with this function block (Boolean function blocks are configurable; other function blocks are read only).
Function Data Type	Allows you to selct the function data type. Select between DINT and REAL. For some function blocks, this field is read-only.
Execution Sequence Number	Displays the execution sequence number for this block (once the project has been verified).
Function Block Comment	Allows you to include a comment with this function block (up to 100 characters).

Parameters tab



On this tab, the following fields are available:

Field	Description
Name	Displays the list of parameter names (read only). **Tip:** EnableIn and EnableOut parameters are only available for those devices that support DeviceLogix v3.00.
Value	Displays the value of each listed parameter. In offline mode, some fields may be editable; those fields appear in white. In online mode (when logic is enabled), some fields are editable (appear in white), while are others are not - this appearance is dependent on the Function Data Type selected on the General tab. Any new values will only be accepted after you click **Apply** or **OK**. **Tip:** For more information on each instruction and the associated parameters, click **Help**.
Type	Displays the function data type (read only). You can select the type in the **General** tab.

Configure the Macro Instruction

Beginning with DeviceLogix firmware revision 4, certain devices support the Macro instruction, which defines a set of logic routines. Once defined, a Macro instruction behaves like other instructions in logic.

A Macro definition contains an instruction and may reference any output inside this logic and wiring. It does not contain any product scope I/O. The name of a Macro definition cannot duplicate any existing DeviceLogix instruction name or another existing Macro definition name.

The maximum number of Macro definitions permitted is defined in the DeviceLogix capability information, which is included in the EDS file.

Work with the Macro Instruction

You can access, edit, and manage Macro instructions through new Tools menu options:

- Create Macro Block
- Macro Block Manager
- Open Macro Block Definition
- Delete Macro Block
- Open Macro Block Logic

Using a Macro instruction encompasses these general steps, each of which is described in following sections.

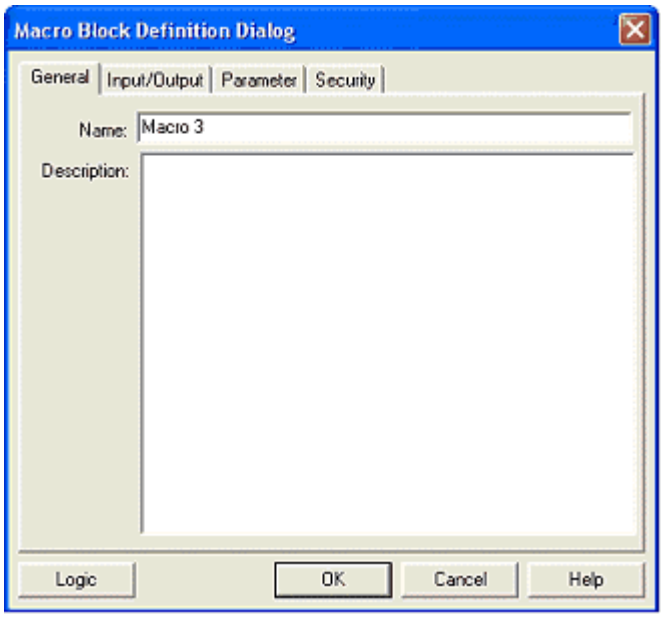
1. Create the Macro definition
2. Define the external I/O binding
3. Add parameters
4. Set security (optional)
5. Apply the Macro definition
6. Add the Macro instruction to logic
7. Manage the Macro

Create the Macro Definition

Complete these steps to create a new Macro definition.

1. Choose **Tools > Macro > Create Macro Block**.

   The Macro Block Definition dialog box appears, containing General, I/O, Parameter and Security tabs.



2. On the General tab, assign a unique and valid Name for the Macro block.

   Valid names are alpha-numeric and are limited to 16 characters.

3. Enter a Description.

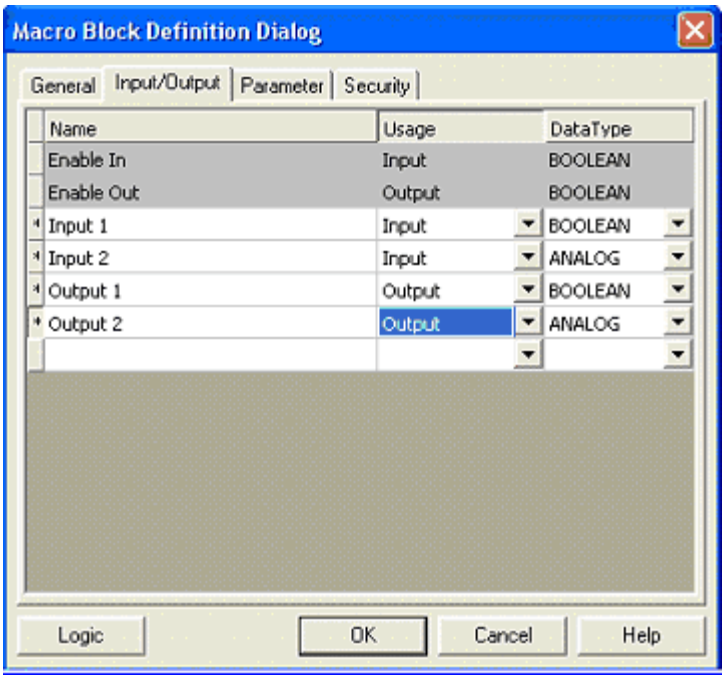
4. Click **OK**.

   If the Name you assigned is not unique, you will be prompted at this time to enter a different Name.

Define the External I/O Binding

To definte the input/output bindings of your new Macro block, complete these steps.

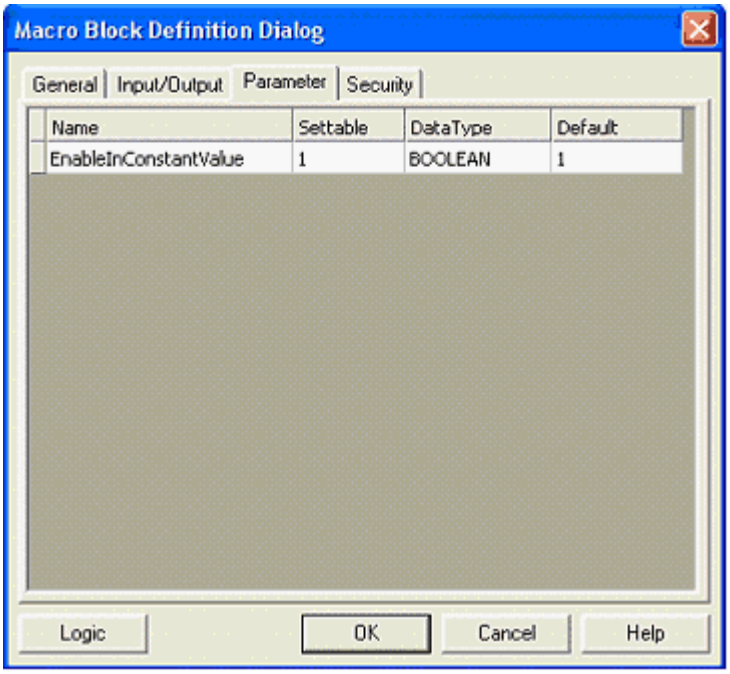
1. Choose **Tools > Macro > Open Macro Block Definition**.
2. Click the Input/Output tab.



3. Assign a Name for the I/O binding.
4. Select the Usage.
5. Select the Data Type.
   - BOOL: the defined I/O can be bound only with Boolean data.
   - ANALOG: the defined I/O can be bound with a wide class of basic data types.
6. Click **OK**.

Add Parameters

While the Parameter tab in the Macro Block Definition dialog box allows you to edit parameters, you cannot add Macro instruction parameters here. You add parameters when you add an instance of the Macro instruction type to your logic.

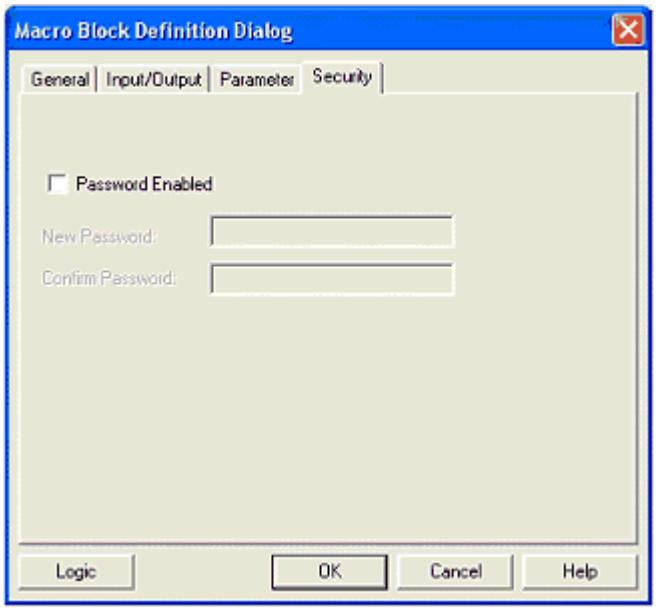


Set Macro Security

Password protection provides the capability for Macro Block designers to protect definitions and parameters. You may enable Macro security when you create the Macro block; by default, password protection is disabled on a new Macro block. The security scope extends to each instance of the Macro block that you add to the logic.

To enable security on a Macro block, complete these steps.

1. Choose **Tools > Macro > Open Macro Block Definition**.
2. Click the Security tab.



3. Select the Password Enabled box.
4. Type the New Password.
5. Confirm the password.
6. Click **OK**.

When editing security-enabled Macro definitions or parameters, you may be prompted to enter the password.

IMPORTANT If you forget the password for the Macro block, you will not be able to access it. In this situation, you must create a new Macro block to overwrite the locked one.

Apply the Macro Definition

When you click **OK** on any tab in the Macro Block Definition dialog box, your edits will be applied if the logic is valid. When you click **OK** or **Logic** on this dialog box, a tabbed editing window for the new Macro block displays. To close this tabbed window, right-click the tab, and choose Close.

Your newly-defined Macro instruction will appear in the new Macro tab in the Instruction toolbar, along with other defined Macro instructions. If you have not created and defined a new Macro instruction, then this tab will be empty. If a Macro instruction is invalid or has not been verified, then it will be grayed-out in the Macro tab.

Add Macro Instruction to Logic

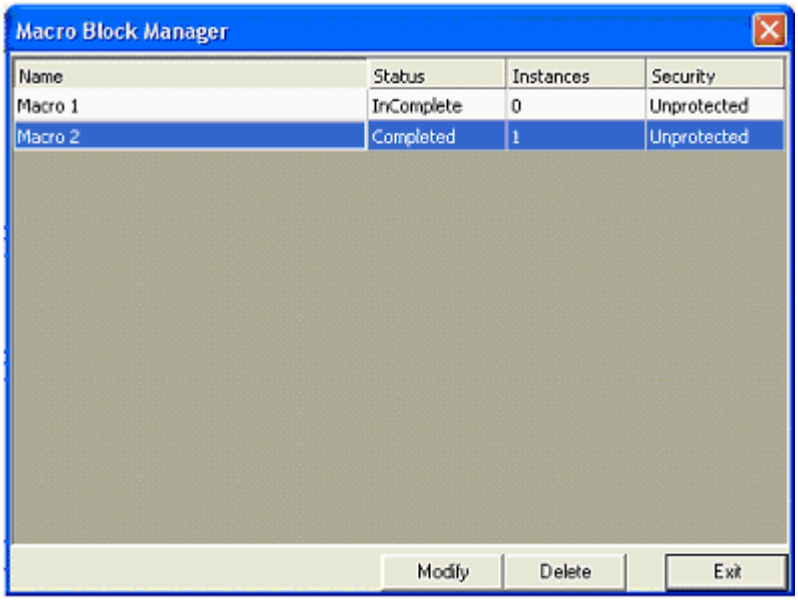
Add the Macro instruction into your logic configuration just as you would add a standard instruction.

- Choose **Edit > Add Element**, select the Macro instruction element in the Add Element dialog box, and click **OK.**
- Click the instruction icon in the Macro tab of the Instruction toolbar.
- Drag the instruction from the Macro tab of the Instruction toolbar and drop it into the logic.

Only valid Macro definitions will appear in the Instruction toolbar and in the Add Element dialog box.

Manage the Macro

Choose **Tools > Macro Block > Macro Block Manager** to open the Macro Block Manager dialog box, where you can select Macro blocks to modify or delete.



You can delete a Macro block only if there are no instances of the Macro in the logic. You cannot undo a Macro block deletion.

Edit the Macro Definition and Parameters

Once you have created a new Macro definition, you can choose **Tools > Macro > Open Macro Definition** to edit the Macro definition and parameters. In addition, with the Macro tabbed window open, you can edit logic for this Macro instruction just as you would a standard instruction.

To edit a Macro's parameters, complete the following steps.

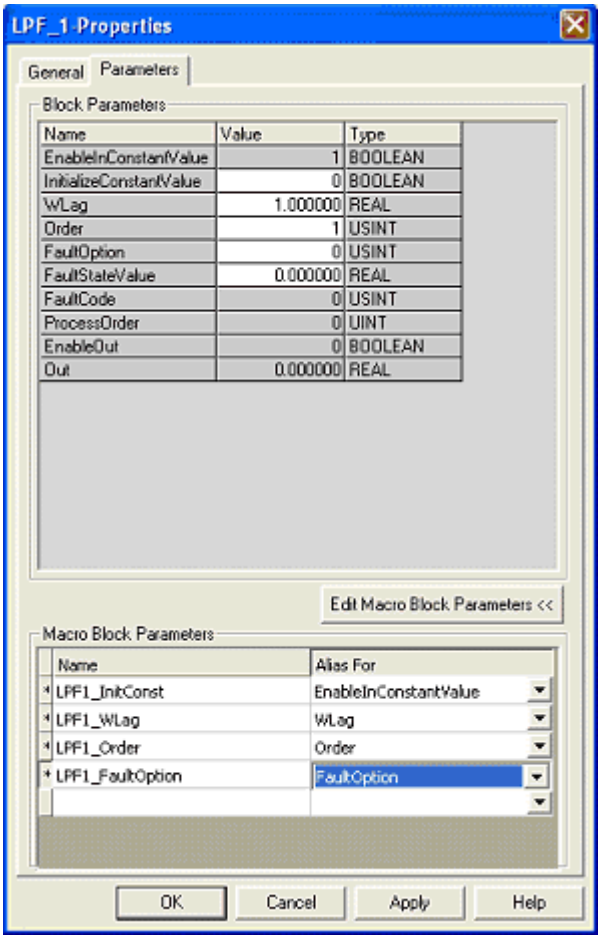
1. In the logic, click the Properties button for the target instructions.

   The Properties dialog box displays.

2. Click the Parameters tab.

3. Click the Edit Macro Block Parameters button.

   The dialog box expands to show the list of externally-viewable Macro Block Parameters.

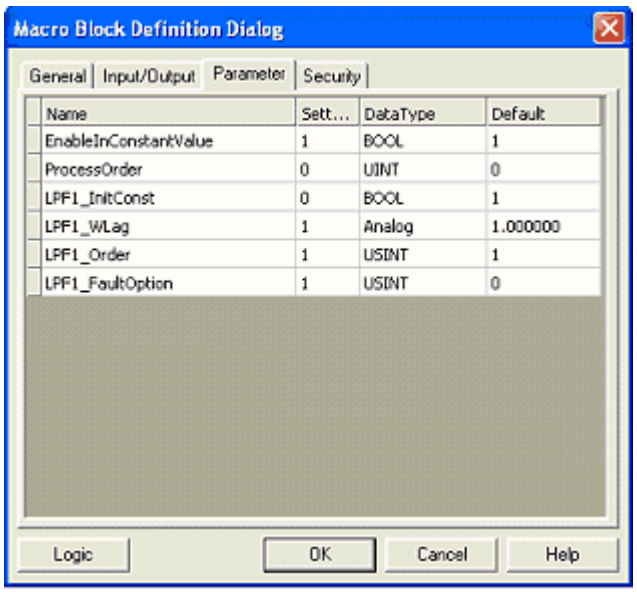


4. Assign a unique Name for the Macro block parameter.

5. Select the block parameter that this Macro block parameter references from the Alias For drop-down list.

6. Click **OK**.

To view the Macro block parameters, choose **Tools > Macro > Open Macro Definition**, and click the Parameter tab.



Consider these guidelines when editing the Macro definition.

- Nested Macro block configuration is not supported. While you configure Macro block logic, all Macro block instructions in the Instruction toolbar are grayed-out so that you cannot add a Macro within a Macro definition.
- You may copy and paste logic from the main logic window to the Macro tabbed window, assuming that the copied logic doesn't contain any Macro instructions. When you copy and paste from the main logic window, all I/O bindings will be removed.
- You may copy and paste logic from the Macro tabbed window to the main logic window. When you copy and paste from the Macro tabbed window, all I/O bindings will be removed.

Offline Operations

Once configured, products containing DeviceLogix functionality can operate without a network to perform small local control functions. Rockwell Automation advises that you use standard DeviceNet cabling in these applications to accommodate the device's power requirements and future configuration needs.

When a network is not used, enable the Comm Status Override Parameter on the Device Parameters Tab. This will override any errors that are generated because there is no network. If this parameter is not enabled, the Outputs will not react to the local logic.

Online Operations

You can perform the following functions when you work on line:

- Communicate with devices
- View the status of I/O with online animation
- Change logic
- Change the value of timers and counters
- Enable and disable logic
- Verify logic
- Compare logic
- Upload and download logic
- Force
- Clear latched hardware faults

Go Online

Once you've created your logic and set up your Scanlist or Override Parameters, you are ready to go online. While the system is online, you can make changes (for example, to the Timer and Counter values) and to the program itself.

Perform the following steps to go online in RSNetWorx software:

1. Click the **Online** icon on the RSNetWorx software toolbar.

2. Once the network browse is completed, double-click the device that you want to connect to.
   The Device Properties dialog is displayed.

3. Click the **DeviceLogix** tab.
   You may be asked to Upload or Download if you haven't already done so. Click **OK**.

4. Once the upload or download is completed, click the **Start Logic Editor** button.

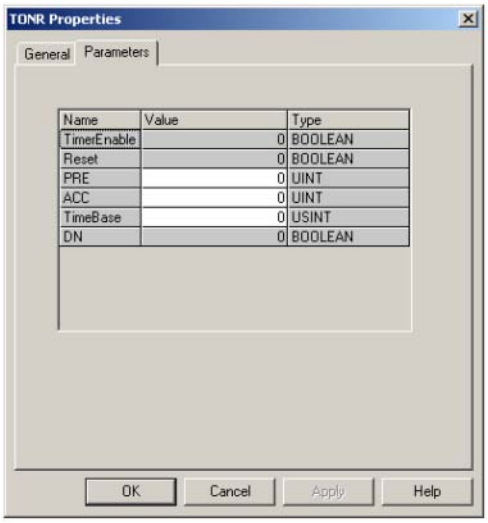
You see:

- The local logic resident in the device
- The real-time status of the inputs and outputs indicated by data on the connecting wires.
- Logic Enabled or Logic Disabled displayed in the online toolbar, depending on the state of the device. You can change the state by simply selecting the desired state.
- The word Animated displayed in the status bar, which indicates that the logic on the screen matches the logic in the device. If you do not see Animated and you want the logic on the screen to match the logic in the device, you must perform an upload or download to synchronize the logic.

5. Double-click on an element. Then select the Parameter tab to view the updating parameter values.

Change Function Block Attribute Values

Many function blocks have attributes that can be modified online. For example you can change Timer block's Preset (PRE) and Accumulator (ACC) while the logic is running. See the following figure.

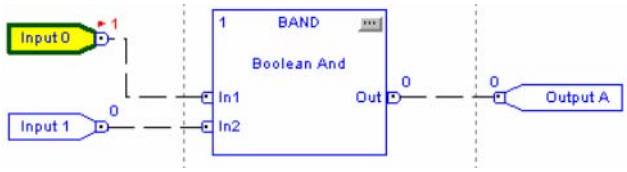


To change the value, perform the following steps:

1. Double-click on the face of the function block you want to change.
2. Select the Parameters tab.
3. Locate the value that you want to change (values that can be changed are white). Enter the new value in the box.
4. Click **OK**.
   The new value takes immediate effect. If you are changing a preset, remember to **Save** the configuration the next time you exit from the Function Block Editor.

Online Animation

Once you go on line and Logic Enable is set to ON, online animation starts. The status of the I/O can be seen in the Function Block Editor schematic. Note that in the online animation mode, you cannot edit instructions.



There are a few remote cases with analog values that result in an error (for example, divide by zero). In these cases, the error value expression has the following format:

Function Block Editor Display Value	Real Value in the Device
1.#R	0x7FC00000 (+NAN)
-1.#R	0xFFC00000 (-NAN)
1.#J	0x7F800000 (+INFINITY)
-1.#J	0xFFC00000 (-INFINITY)

Change Logic

While the Function Block Editor is running and animated, you can change the logic that appears on the schematic by first pressing the Edit button or selecting **Tools > Edit**. When you change the logic, the Function Block Editor will change from the Animated mode into Pending Edits mode. The logic in the device will not be affected and the current program will continue to run. However, the screen will not reflect the real-time status of the logic. The data on the wires will be frozen to whatever it was when you modified the logic. To ensure your changes take affect, perform the following steps:

1. Select **Tools > Edit** so that a check mark appears on the menu, indicating that you are in Edit mode.

2. Select **Communications > Download**.

   You see a message that tells you that logic is running and asks you if you would like to stop it. If you:

   - Select **No** - the logic is not downloaded to the device and the current logic continues to run in the device.
   - Select **Yes** - the logic does not continue to run in the device and new logic is downloaded to the device. When the download is complete, you are asked if you want to start the logic running again. If you:
     - Select **No** - the online toolbar displays Logic Enable Off.
     - Select **Yes** - the online toolbar displays Logic Enabled On; the Function Block Editor is animated. The message Not Saved displays in the status bar until you exit the Function Block Editor and save the new configuration in RSNetWorx for DeviceNet software.

TIP Note that you can use the edit mode icon  on the standard toolbar to enter edit mode.

You can use the download icon  on the online toolbar to download logic to the device.

Enable and Disable Logic

Logic Enable On

Use the Logic Enable On function to notify the device to start processing the stored logic diagram. To set Logic Enable On, select **Communications > Logic Enable On**.

Logic Enable Off

Use the Logic Enable Off option to notify the device to stop executing the logic diagram. When you select this option, the device enters an idle state, turning off the Logic Enable Bit in the produced data. The Logic Enable bit in the Produced I/O assembly of the device reflects a 0 or 1 for Logic Enable Off or Logic Enable On. To set Logic Enable Off, select **Communications > Logic Enable Off.**

Verify Logic

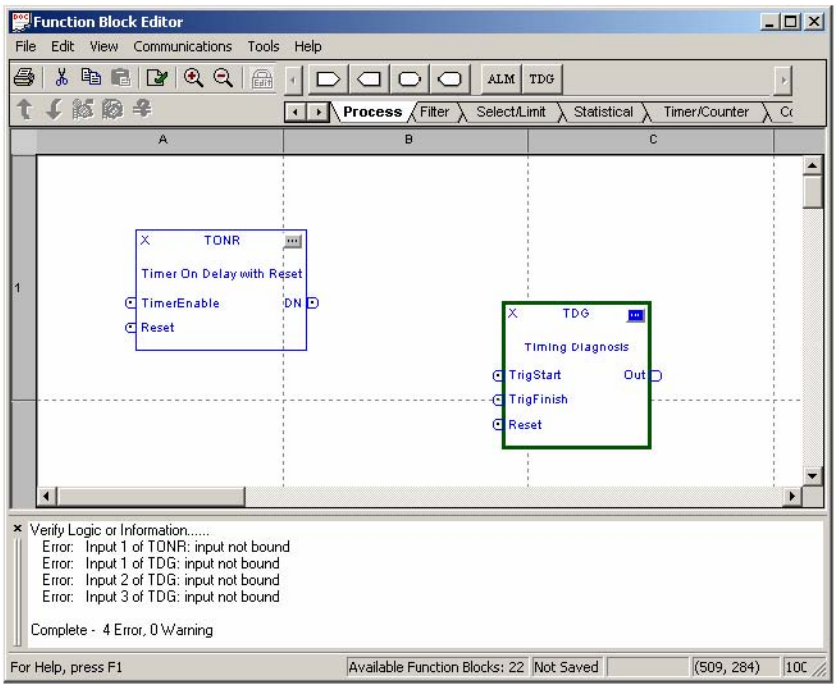
To verify the logic's syntax, click the Verify icon in the standard toolbar or **Tools > Verify**. The verify results will be displayed in the message log window. If an error is identified, double-click the error message to have the current caret jump to the error block.

TIP You can only download the logic to the device after verifying and passing the verification without an error.

TIP For devices running firmware revision 3, each instruction's instance ID number is assigned and valid only after verification. This number may change if the placement of the instruction changed.

For devices running firmware revision 4 or higher, each instruction's instance ID number is assigned once and will never change.

The following figure shows an example verifcation in process:



Compare Logic

Use the Compare option to determine if the logic in the Function Block Editor matches the logic in the device. The results of the comparison display in a pop up window.

To use the Compare function, select **Tools > Compare**.

Upload and Download Logic

Upload Logic

To upload logic, select **Communications > Upload**. The logic configuration in the device is read and displayed in the Function Block Editor. Any unsaved changes will be discarded and Online Animation begins.

Note that some devices are capable of storing screen format information. If a particular DeviceLogix-enabled device supports this feature (dependent on the static memory size), the function block logic and layout position information is stored and will be returned with an upload. Further, the text comments are also saved, but the quality of the content cannot be guaranteed.

When you upload, the uploaded information is not automatically stored into the RSNetWorx for DeviceNet software .dnt file. You must make sure you save any changes after you exit the DeviceLogix Editor to ensure that any changes made to the schematic are saved, including changes made as a result of an upload.

Download Logic

To download logic, select **Communications > Download**. The logic configuration in the Function Block Editor is duplicated and transferred to the device. Before a download can be performed, the logic must pass the verification process. See Verify Logic for more information.

You can download when logic is enabled or disabled. If you download logic when logic is enabled, the download process occurs as follows:

- The Function Block Editor checks to see if logic is enabled in the device.
- If the logic is enabled, you are asked if you want to disable logic. If you:
  - Select **No** - the logic is not downloaded to the device and an error message displays.
  - Select **Yes** - the logic is downloaded to the device.
- If logic was enabled before you downloaded, when the download is complete, you are asked if you want to re-enable the logic. If for some reason the screen format cannot be downloaded into the product, an error message will be displayed and the screen format will not be saved. If logic was not enabled before you downloaded, you are not prompted to enable or disable the logic.
- When the download is complete, you are notified that the download was successful. Note that if the master is currently scanning the device to which you are downloading, you must first access the DeviceNet scanner's scanlist to remove the device from the scanlist. If this new logic results in different connection sizes, you must adjust the master as follows:
  - Put the master in Idle mode (This may mean turning a processor's key switch to Program.)
  - Access the device parameter's to change the size of the connections.
  - Download the device to the scanlist.
  - Click the Edit I/O Parameters button and verify the Rx and Tx size corresponds to the new connection size.
  - Put the master in Run mode for normal operation.

You can click the Cancel button anytime during the download process to abort downloading the logic to the device. If you click the Cancel button, an error message displays and neither old logic nor new logic exists in the device. You must let the download complete for logic to be contained in the device.

If logic is enabled during the download process, once the download is complete, online animation starts. You see the real time values on the wires connecting the elements. If logic is disabled, there is not any activity; the current state of connections is displayed and inputs can be changed, but online animation does not start.

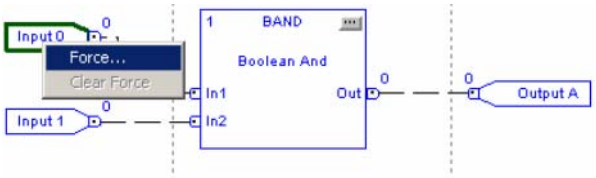
When you download, the downloaded information is not automatically stored into the RSNetWorx for DeviceNet software .dnt file. You must make sure you save any changes after you exit the DeviceLogix Editor to ensure that any changes made to the schematic are saved, including changes made as a result of an upload.

Forcing

The DeviceLogix Function Block Editor provides troubleshooting capabilities by allowing you to force inputs and outputs that allow you to verify the run time result of your logic.

To aid in troubleshooting and debugging of your schematic, you can force hardware inputs and hardware outputs. No other inputs or outputs can be forced within the Function Block Editor. Network inputs can be forced in the device from where they originate. If you exit the Function Block Editor with a force enabled, that force will stay in effect until you remove it.

To apply forces, right-click on the input or output element you wish to force. A pop-up appears that lists the forcing options .



TIP You cannot use the Force function in offline mode.

TIP If Logic Enable is Off (logic is not running in the product), the Force function has no effect. In this case, if the Force value is modified, it will not take effect until Logic Enable is On.

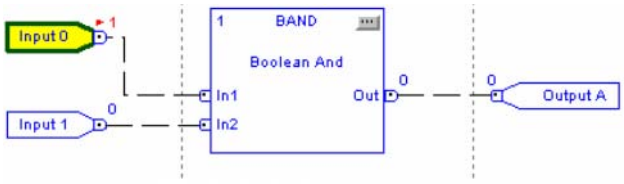
TIP You do not have to be in Edit mode to use forces.

Forcing Inputs

The following list describes the input force options.

Input Force Option	Description
Force On	Forces the input ON.
Force Off	Forces the input OFF.
Remove Force	Returns control of the input to the hardware device and turns the instruction color back to white.

The following figure shows the result of placing a force on an input. When forced, an input element turns yellow with a red triangle indicator and the status value reflects the force state chosen.

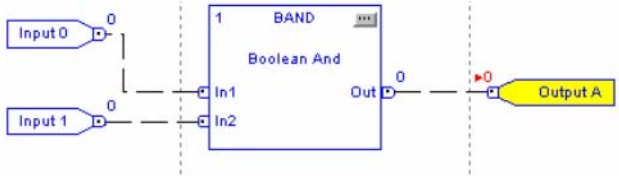


Forcing Outputs

The following list describes the output force options.

Output Force Option	Description
Force On	Forces the input ON.
Force Off	Forces the input OFF.
Remove Force	Returns control of the input to the hardware device and turns the instruction color back to white.
Clear Fault	When a device supports latching of faults, this selection clears a hardware output fault indication (for example, off-wire or short circuit). The actions related to Clear Fault are product specific.

The following figure shows the result of placing a force on an output. When forced, an output element turns yellow with a red triangle indicator and the status value reflects the force state chosen.



Notes:

Navigate the Old Ladder Editor Interface

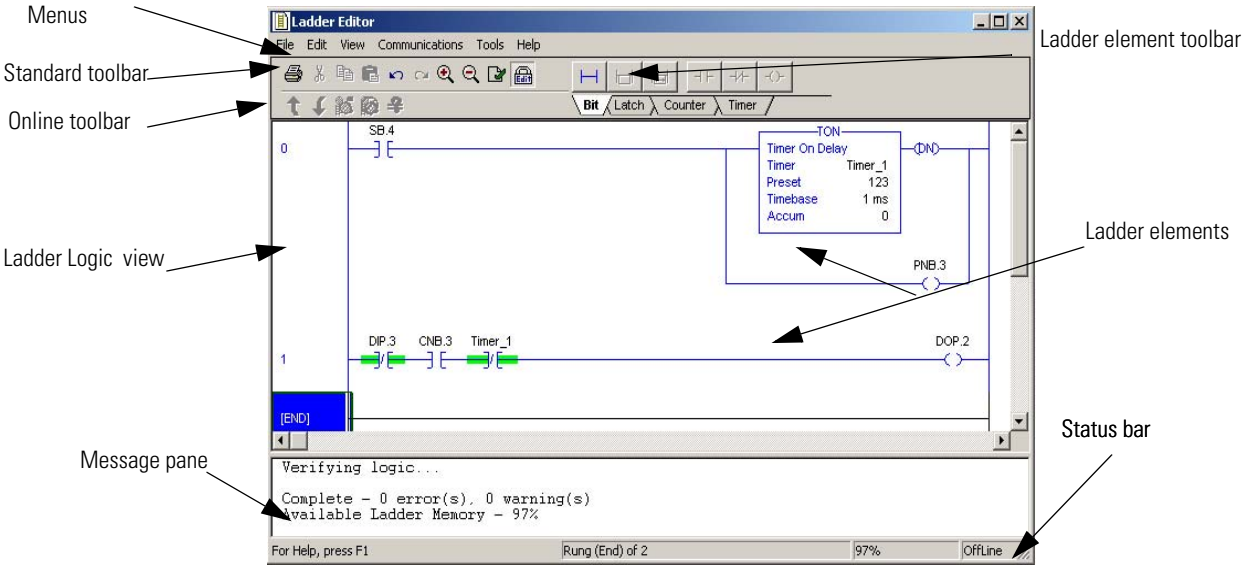
What This Chapter Contains

Read this chapter to learn more information about the old Ladder Editor interface. The following table lists what this chapter contains and where to find specific information.

Components

To help you configure your ladder logic, the old Ladder Editor consists of:

- Ladder elements
- Configuration toolbars
- Ladder Logic view
- Message pane
- Status bar
- Menus



Ladder Elements

The ladder logic in the old Ladder Editor consists of rungs. The rung consists of functional instruction elements which include bits, latches, counters, and timers. Logic combinations are displayed on the left part of the rung and logic outputs are displayed on the right side of the rung. Logic outputs are determined by the type of instruction (bit, latch, counter, or timer). For each rung, you can enter rung comments which describe the logic that you created.

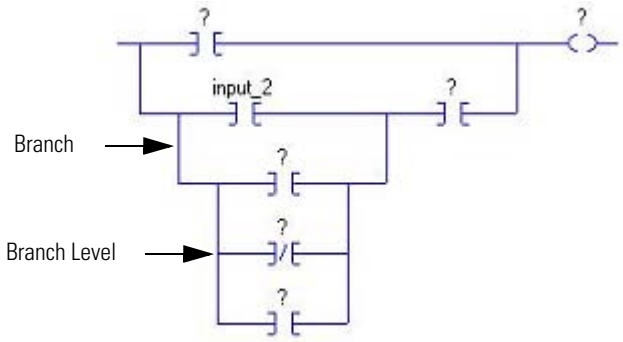
Ladder elements are:

- Rung
- Bit
- Latch
- Counter
- Timer

Rung Element

Rung Elements

Element	Description
	Rung Element -The basic executable unit in the old Ladder Editor. Each rung has at least one input condition and one output condition. The True or False value of an input determines the output value. You can add a comment to a rung to clarify or describe the instruction.
	Branch Element - Connection element that adds OR logic in the old Ladder Editor. The branch element always resides on a rung element and, therefore, cannot exist without the rung element.
	Branch Level Element - Connection element that adds OR logic in the old Ladder Editor. The branch level element can reside on a branch element or a branch level element. The branch level element can also store instructions.



Bit Element

Bit Elements

Element	Description
	Examine If Open (or XIO) Element
	Examine If Close (or XIC) Element
	Output Coil Bit (or Output Energize Bit) Element

Latch Element

Latch Elements

Element	Description
	Reset Dominant (RS) Latch Element
	Set Dominant (SR) Latch Element
	Latch Reset Element

The old DeviceLogix Ladder Editor provides standard latching capabilities. These include the set dominant latch (SRL) and the reset dominant latch (RSL).

The RS and SR latch elements require an accompanying reset element. The latch/reset pair must use the same tag name.

Set Dominant Latch

When using a set dominant latch, the accompanying reset element must appear before the SRL element. Refer to the truth table for information about the set dominant latch.

Function	Reset	Input (Set)	Output Value at time = t_0	Output Value at time = t_{0+1}
SR set dominant (A set dominant block goes to the set state if both inputs are true.)	0	0	0	0
	0	1	0	1
	1	0	0	0
	1	1	0	1
	0	0	1	1
	0	1	1	1
	1	0	1	0
	1	1	1	1
RS reset dominant (A reset dominant block goes to the false state if both inputs are true.)	0	0	0	0
	0	1	0	1
	1	0	0	0
	1	1	0	0
	0	0	1	1
	0	1	1	1
	1	0	1	0
	1	1	1	0

Reset Dominant Latch

When using a reset dominant latch, the accompanying reset element must appear after the RSL element. Refer to the truth table for information about the reset dominant latch.

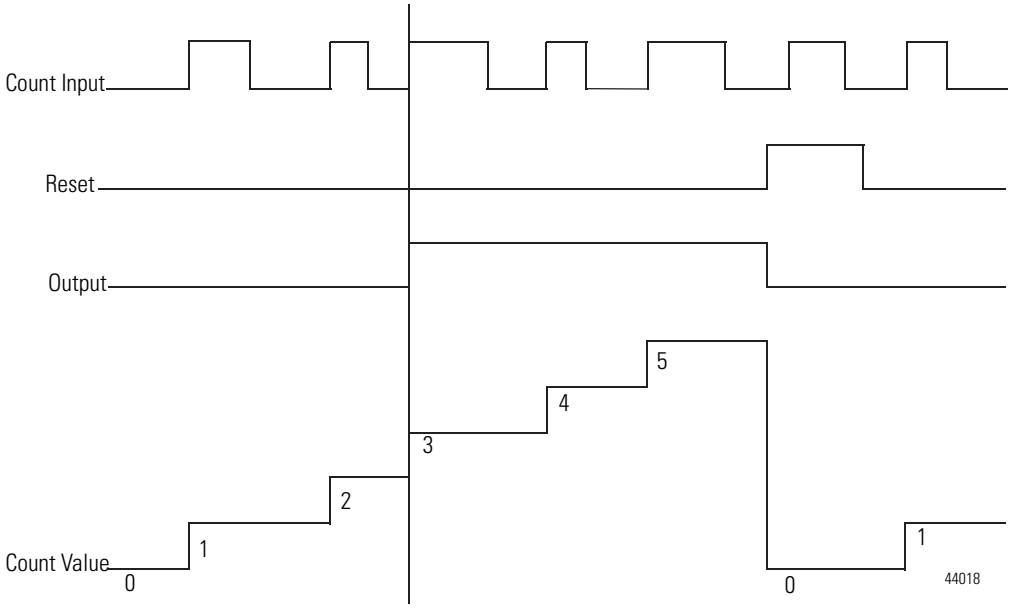
Counter Element

Counter Elements

Element	Description
CTU	Count Up Element - The up counter increments its accumulator when the input is true. It has a preset value and an accumulator value.
CTD	Count Down Element - The counter decrements its accumulator when the input is true. It has a preset value and an accumulator value.
RES	Count Reset Element - Resets the accumulator and output value.

Up Counter

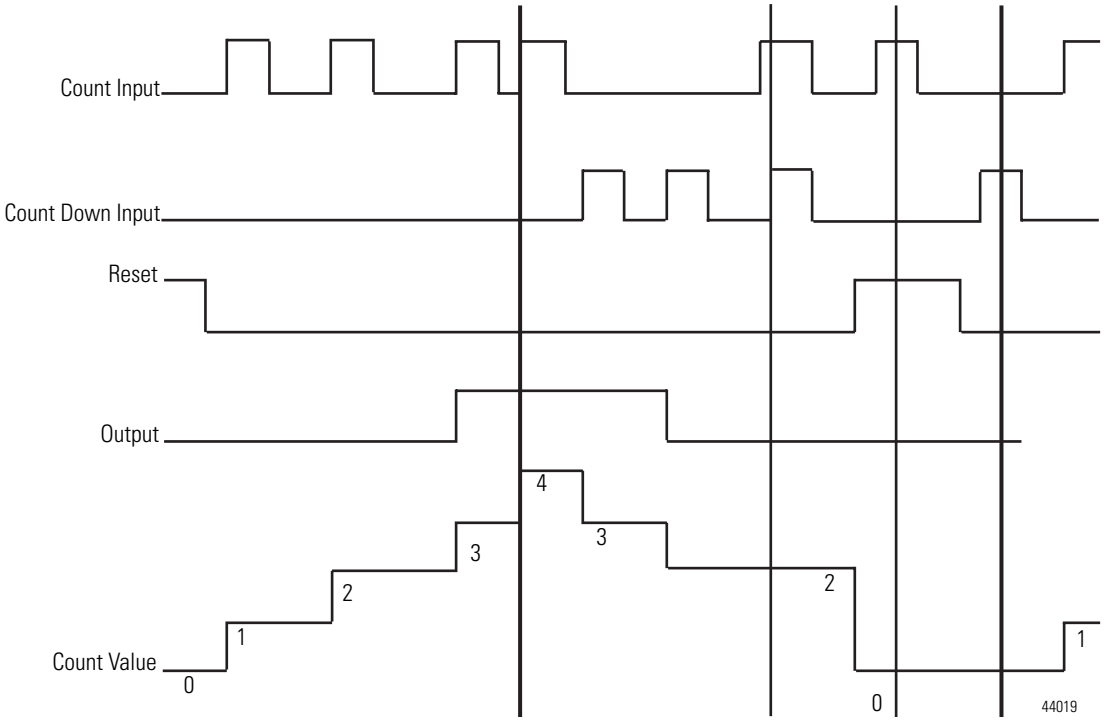
The up counter counts up on a false to true transition of the count input and sets its output to true when the accumulator reaches the preset value. The accumulator continues to count up until the counter is reset or until the counter reaches 65,535. A reset causes the accumulator to be set to zero and sets the output to false. The reset (RES) element must reference the same tag name as the counter that it will be resetting. The illustration shows an up counter with a preset value of 3.



Down Counter

The down counter can be an independent down counter or it can work in combination with an up counter as an up down counter. The down only counter can be consider an up down counter, but without count up input.

The counter increments the accumulator any time the count up input changes from false to true and decrements the counter any time the count down input changes from false to true. When the accumulator is above or equal to the preset value, the counter sets its output to true. When the accumulator falls below the preset value, the counter resets its output to false. As with the up counter, the up down counter resets when a true level is detected on the reset input. When a reset occurs, the counter causes the accumulator to be set to zero and sets the output to false. Refer to the illustration of the counter function for clarification.The illustration shows an up counter with a preset value of 3.



Timer Element

Timer Elements

Element	Description
PUL	Pulse Trig Timer - It has a preset value and an accumulator value. The timing base can be selected as 1 ms or 10 ms.
TON	On Trig Timer (On-delay Timer) - It has a preset value and an accumulator value. The timing base can be selected as 1 ms or 10.
TOF	Off Trig Timer (Off-delay Timer). It has a preset value and an accumulator value. The timing base can be selected as 1 ms or 10.
RES	Timer Reset Element - Reset the timer element

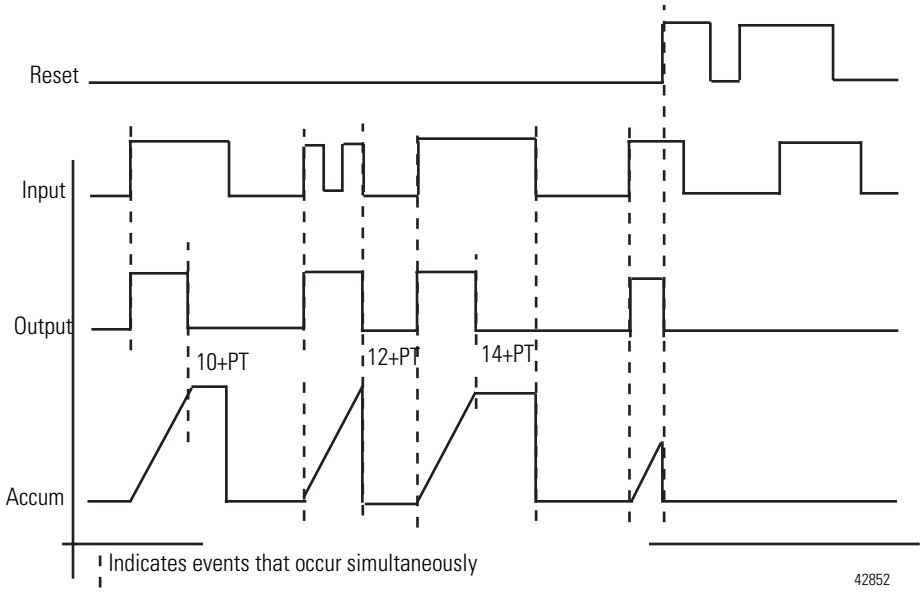
Pulse Timer

The Pulse Timer generates a true value on its output for a fixed amount of time. The duration of the true pulse is determined by the Preset value along with the Timebase of the timer. When the input to the Timer changes from false to true, it sets the output to true and the Accumulator starts counting. The Timer then increments the Accumulator each time the Timebase number of milliseconds has expired. When the Accumulator reaches the preset value, the Timer resets the output to false.

In the Pulse Timer, the input acts only as a trigger to start the Accumulator counting. Once the Accumulator starts timing, it continues to rise regardless of the state of the input. As long as the output is true and the Accumulator is counting, additional triggers of the input do not affect the state of the output or the count of the Accumulator. Once the Accumulator reaches the preset value and the Timer resets the output to false, the Pulse Timer can again trigger the process by sensing a false to true transition on the input contact.

Even if the input remains on the entire time, the Accumulator is counting. When the Accumulator reaches the preset value, the Timer resets the output to false.

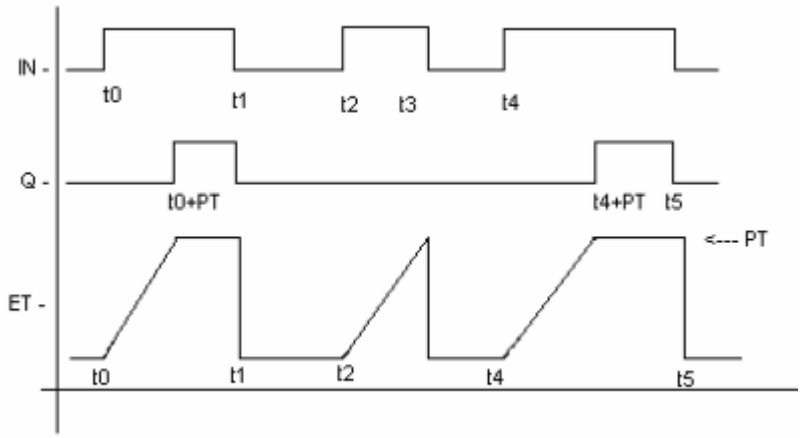
At any point during the operation of the Timer, if it detects a true level on the Reset input, it will disable the Timer and set the output to false. The Timer must again be triggered by a false to true transition on the input. This means that if the input is true and the Timer is reset, the Timer will remain inactive. Even if the Reset changes back to false while the input is true, the Timer remains inactive. In order to start a new pulse operation, the input must change to false and then back to true. Refer to the illustration of the timer function for clarification.



On Delay Timer

The On Delay Timer delays the output response to an input by a desired amount of time. When the input is true, the Timer increments the Accumulator and when the input goes false, the Timer resets the Accumulator. Each Timebase unit of time, the Timer increments the Accumulator. When the Accumulator reaches the preset value, the Timer sets the output to true. The Timer maintains the output true status as long as the input remains true. When the input changes from true to false, the Timer resets both the output and the Accumulator. If the input goes false before the time period specified by the Timebase and Preset, the output remains false and the Accumulator is cleared, essentially ignoring the input.

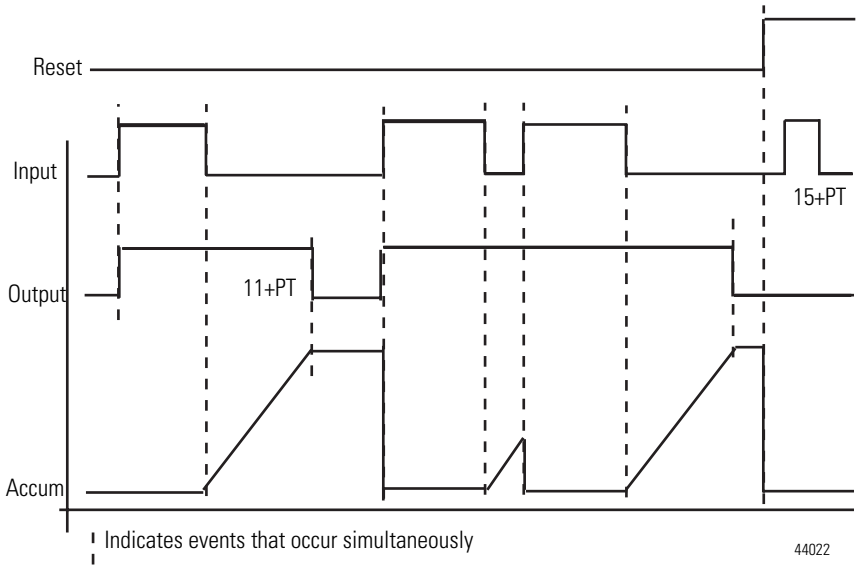
If the Timer senses a true level on the Reset input at any time during the operation of the Timer, it resets the output to false and clears the Accumulator. Because the reset line is level sensitive, the timer remains reset until the Timer detects a false on the Reset input. Also, because the input is level sensitive, the Timer will again begin to increment the Accumulator if the reset line goes false while the input remains true. Refer to the illustration of the timer function for clarification.



Off Delay Timer

The Off Delay Timer works the same way as the On Delay Timer but instead of delaying the true status of the output, it delays the false status of the output. The input to this timer is a level sensitive false with a level-triggered reset. This means that when the input is false, the Timer increments the Accumulator and when the input goes true, the Timer resets the Accumulator. Each Timebase unit of time, the Timer increments the Accumulator. When the Accumulator reaches the preset value, the Timer sets the output to false. The Timer maintains the output false status as long as the input remains false. When the Input changes from false to true, the Timer sets the output true and resets the Accumulator. If the Input goes true before the time period specified by the Timebase and Preset, the output remains true and the Accumulator is cleared, essentially ignoring the input.

If the Timer senses a true level on the Reset Input at any time during the operation of the Timer, it resets the output to false and clears the Accumulator. Because the reset line is level sensitive, the Timer remains reset until the Timer detects a false on the Reset input. Also, because the input is level sensitive, the Timer will again begin to increment the Accumulator if the reset line goes false while the input remains false. However, because the reset logic already set the output to false, the time delay has no effect because the output is already false. If the input is true when the reset goes false, the Timer sets the output to true. Refer to the illustration of the timer function for clarification.



Configuration Toolbars

The old Ladder Editor toolbars are:

- Standard - Lets you perform general editing functions, verify logic, and toggle edit mode.
- Ladder Element - Lets you add ladder elements to the old Ladder Editor. The toolbar changes based on the element you are adding (bit, latch, timer, or counter). You can drag and drop elements from this toolbar into the old Ladder Editor.
- Online - Lets you perform online functions. This toolbar is enabled only when you are working on line.

Standard Toolbar

Standard Toolbar

Function	Description
	Sends the ladder logic to a printer.
	Removes the selected ladder element.
	Duplicates the selected content to save it in the clipboard.
	Removes the saved content in the clipboard and puts the content in the old Ladder Editor where your cursor is resting.
	Cancels the last action.
	Performs again the previously cancelled action.
	Increases (make larger) the view of the configuration. You lose view of part of the configuration pane of the old Ladder Editor when you use this function.
	Decreases (make smaller) the view of the configuration. You see more of the configuration pane of the old Ladder Editor when you use this function.
	Confirms that the logic that you configured is valid.
	Toggles between working in the Edit mode and not working in the Edit mode. In the Edit mode, you can make changes to the Logic. You must exit the Edit mode to download the logic to the device.

Ladder Element Toolbars

The ladder element toolbars are:

- Bit Element
- Latch Element
- Counter Element
- Timer Element

TIP It is not required for all DeviceLogix-enabled products to support all of the element types. The EDS file for each product provides information on what element are supported. If an element is not supported, it will not appear in the associated toolbar.

Bit Element Toolbar

Bit Element Toolbar

Function	Description
	Rung Element
	Branch Element
	Branch Level Element
	Examine If Open (or XIO) Element
	Examine If Close (or XIC) Element
	Output Coil Bit (or Output Energize Bit) Element

Latch Element Toolbar

Latch Element Toolbar

Function	Description
	Rung Element
	Branch Element
	Branch Level Element
	Reset Dominant Latch Element
	Set Dominant Latch Element
	Latch Reset Element

Counter Element Toolbar

Counter Element Toolbar

Function	Description
	Rung Element
	Branch Element
	Branch Level Element
	Count Up Element
	Count Down Element
	Count Reset Element

Timer Element Toolbar

Timer Element Toolbar

Function	Description
	Rung Element
	Branch Element
	Branch Level Element
	Pulse Timer Element
	On Delay Timer Element
	Off Delay Timer Element
	Timer Reset Element

Online Toolbar

Use the Online toolbar to perform functions when you are working on line in the old Ladder Editor.

Online Toolbar

Function	Description
	Upload the logic configuration from the device to the old Ladder Editor configuration tool.
	Download the logic configuration from the old Ladder Editor to the device. The configuration must pass the logic verification process for the download to be successful.
	Run the DeviceLogix logic configuration that you have downloaded to the device.
	Stop the DeviceLogix logic configuration running in the device.
	Compare the logic in the device with that in the old Ladder Editor configuration tool.

Ladder Logic View

The Ladder Logic View is the area in which you place functional elements to create logic.

The printed ladder logic uses A4 sheets of paper. The ladder logic is printed at full size, regardless of the setting of the current zoom level. To make adjustments to the ladder logic before printing, you can use the print preview feature to see how it will print. The ladder can also be sent to a plotter for easier viewing.

TIP

Because a device has limited memory to store logic, the display of uploaded information may not match the appearance of information downloaded to the device. Therefore, once your logic configuration is complete, you should print a copy of it for your records. Additionally, Rockwell Automation also suggests that you save the RSNetWorx for DeviceNet software (*.dnt) project file.

Message Pane

The Message Pane on the bottom of the old Ladder Editor displays the results when you verify logic. If the old Ladder Editor finds an error in your logic, click on the error or warning message in the message pane and the cursor goes to the place in the old Ladder Editor where the error is in the logic.

The message pane also displays the amount of memory that is available in the device (only after you verify the logic).

From the View menu, you can toggle between displaying the message pane or not displaying it.

Status Bar

The Status bar, located on the bottom of the old Ladder Editor, provides a view of the current working status of the old Ladder Editor. You can toggle between displaying the Status Bar and not displaying it by selecting **View > Status Bar**.

There are four panes that provide unique information about the old Ladder Editor.

Status Bar

Pane	Description
Rung Index	Displays the rung the cursor is resting on and the total number of rungs in the configuration.
Ladder Available Memory Count	Indicates the available percentage of memory for the ladder logic. When available memory is less than 5%, Low Memory will be displayed.
Online Indication	Indicates if you are working on line or off line in the old Ladder Editor.
Logic Enable Indication	Indicates if the Logic Enable function is on or off. If you are working off line, this pane does not display.

Menus

The old Ladder Editor has six menu options

- File
- Edit
- View
- Communications
- Tools
- Help

File Menu

The File menu lets you perform printing functions and lets you exit the old DeviceLogix Ladder Editor.

File Menu

Function	Description
Print	Sends the ladder logic to a printer.
Print Preview	View the ladder logic before sending the logic to the printer.
Print Setup	Choose printer and printing options.
Close	Exit the old DeviceLogix Ladder Editor.

Edit Menu

The Edit menu lets you modify elements in the old Ladder Editor.

Edit Menu

Function	Description
Undo	Cancels the last action.
Redo	Performs again the previously cancelled action.
Cut	Removes the selected ladder element.
Copy	Duplicates the selected content to save it in the clipboard.
Paste	Places the content in the clipboard into the old Ladder Editor at the position that your cursor is resting.
Select All	Highlights all of the elements in the old Ladder Editor (to copy or cut). This menu option has different behavior depending upon the current selection in the old Ladder Editor: • When a rung is selected, Select All selects all of the rungs. • When a branch leg is selected, Select All selects the entire branch. • When a branch level is selected, Select All selects the entire branch level.
Add Ladder Element	Insert a ladder element at the place where the cursor is positioned. The Add Ladder Element window opens for you to select the type of element you wish to insert.
Delete Ladder Element	Removes the selected element from the old Ladder Editor.
Edit Ladder Element	Modifies the selected ladder element by you changing the instruction type or the instruction type parameters. Note that it is possible to change only instruction types of the same category (bits, latch, timers, counters).

Edit Menu

Function	Description
Edit Element	Modifies the selected ladder element. It is possible to change the instruction type or the instruction type parameters. Note that it is possible to change only instruction types of the same category (bits, latch, timers, counters).
Edit Element Comment	Modifies the comment attached to the selected rung or instruction type. Note that you cannot add comments to branch or branch level elements.
Properties	Access the Configuration Properties dialog to modify or add configuration information.
Recovery Mode	Accesses the Recovery Mode dialog to select between having logic enabled or disabled following the restoration of a configuration. The Recovery Mode is used in conjunction with the Auto Device Replace Configuration Recovery feature.

View Menu

The View menu lets you modify your view of the old Ladder Editor.

View Menu

Function	Description
Status Bar	Toggles between displaying and not displaying the status bar at the bottom of the old Ladder Editor.
Zoom In	Increases (make larger) the view of the configuration. You lose view of part of the configuration pane of the old Ladder Editor when you use this function.
Zoom Out	Decrease (make smaller) the view of the configuration. You see more of the configuration pane of the old Ladder Editor when you use this function.
Zoom to 100%	Causes the configuration pane to mirror what is seen on the printed version when the configuration pane is printed.

Communication Menu

The Communications menu lets you work on line in the old Ladder Editor.

Communication Menu

Function	Description
Upload	Reads the logic configuration in the device and displays it in the old Ladder Editor. You lose any changes you made in the configuration and online animation starts.
Download	The logic configuration in the old Ladder Editor is transferred to the device and online animation starts. For the download to begin, the configuration must pass the verification process.
Logic Enabled On	Runs the logic configuration that you have downloaded to the device.
Logic Enabled Off	Stops running the logic configuration that you downloaded to the device.

Note that these communication options are only available when you are working in the online mode.

Tools Menu

The Tools menu lets you modify logic in the old Ladder Editor.

Tools Menu

Function	Description
Logic Verify	Confirms that the logic that you configured is valid. Confirms that: • each rung has a minimum input and output element • each branch has a minimum element • each element has the minimum number of bound pins • each attribute has the correct data type associated with it • the two ends of a binding are of the same type
Compare	Compares the logic configured in the device with the logic configured in the old Ladder Editor. You must be working in the online mode to use this function.
Edit Mode	Toggles between working in the Edit mode and not working in the Edit mode. In the Edit mode, it is possible to modify the logic but you cannot communicate with the device. (That is, you cannot upload or download logic or use the Logic Enable On or Logic Enable Off functions.) When you are not working in the Edit mode, it is possible to perform online animation. If you are working in online mode, it is possible to set the preset value or accumulator value of counters or timers, but you cannot modify logic.
Edit Tags	Creates latch, counter, and timer tag names.

Help Menu

The Help menu provides you with assistance when you are working in the old Ladder Editor.

Help Menu

Function	Description
Ladder Editor Help	Accesses the help files available for the old Ladder Editor.
Release Notes	Accesses release notes pertaining to DeviceLogix functionality or the old Ladder Editor.
About DeviceLogix	Opens the About Ladder Editor window to learn revision and copyright information about the Ladder Editor. File revision identifies the revision of the Ladder Editor DLL application.

Create Logic in the Old DeviceLogix Ladder Editor

What This Chapter Contains

Read this chapter to learn how to create logic in the old DeviceLogix Ladder Editor and about the options available to you when you work on line.

TIP For more information on the tasks you can perform in the old Ladder Logic Editor, see the Ladder Logic Editor online help.

Understanding and Working With I/O Tags

There are two kinds of I/O tags: static and dynamic.

Static tags are:

- Device Input (DIP) - physical input point of the device
- Consumed Network Bit (CNB) - network input bit
- Fault Bit (FB)- device fault bit
- Status Bit (SB) - device status bit
- Device Output (DOP) - physical output point of the device
- Produced Network Bit (PNB) - network output bit

Dynamic tags are those created dynamically during configuration. They are the referenced output bits of latch, counter, and timer elements.

Online Operations

You can perform the following functions when you work on line:

- Communicate with devices
- View the status of I/O with online animation
- Change logic
- Change the value of timers and counters
- Enable and disable logic
- Verify logic
- Compare logic
- Upload and download logic
- Force
- Clear latched hardware faults

Communication with Devices

The old DeviceLogix Ladder Editor communicates with the devices to accomplish the following:

- Determine device type
- Get communication parameters
- Set communication parameters
- Download configuration to the device
- Upload configuration from the device
- Modify function block parameters during animation (for example, Preset value and Accumulated counts for counters or Preset time and Elapsed time for timers)
- Force I/O values
- Start or Stop local logic (toggle Logic Enabled)
- Display device faults during animation
- Clear I/O faults
- Reset the device

Note that all communication between the old DeviceLogix Ladder Editor and the device is done using explicit messaging connections.

Go On Line

Once you have created your logic and set up your scanlist or override parameters, you are ready to go on line.

To go on line:

1. Click the Online icon on the RSNetWorx software toolbar. The RSNetWorx software will browse the network.

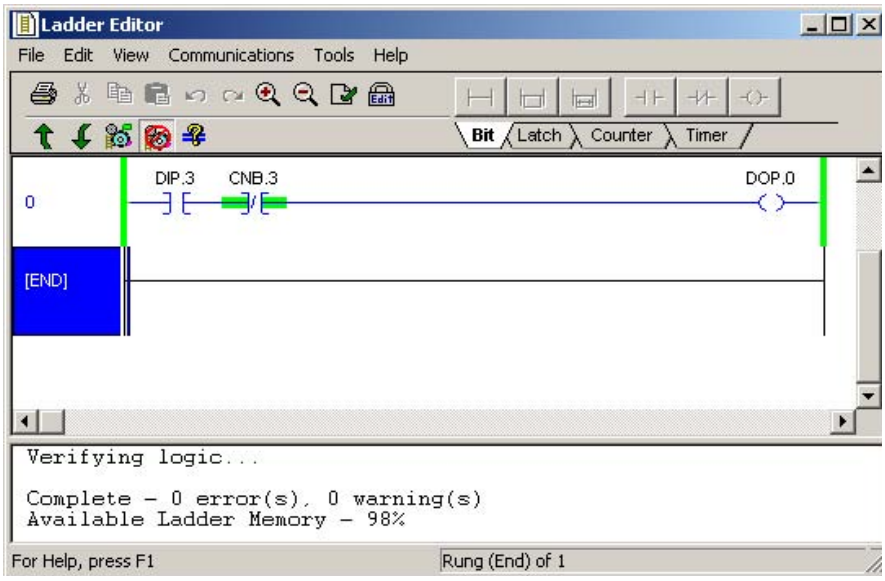
2. When the browsing is complete, double click the DeviceLogix-enabled device.

3. Click the DeviceLogix tab in the device properties window. You may be asked to upload or download if you have not done so already.

4. Once the upload or download is complete, click the **Start Logic Editor** button.

You see:

- Local logic resident in the device.
- Real-time status of inputs and outputs indicated by 0s and 1s on the connected wires.
- The value of the accumulators change based on the logic (if you have timers and/or counters).
- Logic Enabled or Logic Disabled displayed in the online toolbar, depending upon the state of the device.
- The word Animated displayed in the status bar, which means the logic on the screen matches the logic in the device. If you do not see the word Animated displayed, perform an upload or a download to synchronize the logic.



Online Animation

Once you go on line and Logic Enable is set to on, online animation starts. The status of the I/O can be seen in the old Ladder Editor configuration pane. Note that in the online animation mode, you cannot edit instructions.

Change Logic

While the logic is running and animated, you can change the logic that appears in the old Ladder Editor. The old Ladder Editor changes from the animated mode to the pending edits mode. The logic in the device is not affected and the current program continues to run. However, the screen does not reflect the real-time status of the logic. The data on the wires is frozen to what it was when you entered Edit Mode. To cause your changes to take affect:

1. Select **Tools > Edit Mode** so that a check mark appears on the menu, indicating that you are in Edit mode.

2. Select **Communications > Download**.

   You see a message that tells you that logic is running and asks you if you would like to stop it. If you:

   - Select **No** - the logic is not downloaded to the device and the current logic continues to run in the device.

- Select **Yes** - the logic does not continue to run in the device and new logic is downloaded to the device. When the download is complete, you are asked if you want to start the logic running again. If you:
  - Select **No** - the online toolbar displays Logic Enable Off.
  - Select **Yes** - the online toolbar displays Logic Enabled On; the old Ladder Editor is animated. The message Not Saved displays in the status bar until you exit the Ladder Editor and save the new configuration.

TIP

Note that you can use the edit mode icon  on the standard toolbar to enter edit mode.

You can use the download icon  on the online toolbar to download logic to the device.

Change the Value of Timers and Counters

When working in the online mode, it is possible to change the preset and accumulator values of timers and counters.

To change the value:

1. Double click on the value you want to change.
2. Enter the new value into the box.
3. Press the Enter key.

The new value takes effect immediately. If you change a preset value, remember to save the configuration the next time you exit the old Ladder Editor.

Enable and Disable Logic

Logic Enable On

To cause the device to start processing the stored logic configuration:

Click **Communications > Logic Enable On**.

Logic Enable Off

To cause the device to stop executing the logic configuration:

Click **Communications > Logic Enable Off**.

If the logic in the old Ladder Editor does not match the logic in the device, the Logic Enable On and Logic Enable Off functions are disabled (grayed out). You must verify the logic and download it to the device for the Logic Enable On and Logic Enable Off functions to be enabled.

Verify Logic

When you use the Verify Logic function, you test your configuration for mistakes.

To verify your logic, click **Tools > Logic Verify**.

TIP Another way to verify logic is to click the verify logic icon  on the toolbar.

The following is verified during the logic verify process:

- Each rung has the minimum number of input and output instructions
- Each branch has the minimum number of instructions
- Each parameter has the correct data type associated to it
- Latches have been correctly paired with reset instructions
- Outputs have only been used once
- Counters, timers, and latches have unique tag names

Once the Verify Logic process is complete, you see the results in the pane on the bottom of the screen. If any of the checks fail the verification process, an error message displays in the pane. Click the error or warning line to cause the cursor to go to the place in the old Ladder Editor where the error is in your logic. Your logic configuration must pass the logic verification process before you can download the logic to the DeviceLogix device.

Note that in the Message pane you see the amount of memory that is available in the device. The amount of available memory displays only if the logic passes verification.

Compare Logic

Use the Compare option to determine if the logic in the old Ladder Editor matches the logic in the device. The results of the comparison display in a pop up window.

To use the Compare function, select **Tools > Compare**.

Upload and Download Logic

Upload Logic

To upload logic, select **Communications > Upload**. The logic configuration in the device is read and displayed in the old Ladder Editor. Any unsaved changes will be discarded and Online Animation begins.

Note that because a device has limited memory to store logic, the layout of the configuration is not downloaded to the device. Therefore, the information that is uploaded from the device will not match in appearance the information you downloaded to the device (that is, the user interface will look different although the logic is the same). If you want a record of the layout of your configuration, we suggest that you print a copy of your logic configuration before you download it to the device.

Also note that uploaded information is not automatically stored in the RSNetWorx for DeviceNet softtware .dnt file. In order for updated information and any other changes you make in the logic configuration to be retained, you must use the Save function in RSNetWorx for DeviceNet software after you exit the old Ladder Editor.

Download Logic

To download logic, select **Communications > Download**. The logic configuration in the old Ladder Editor is duplicated and transferred to the device. Before a download can be performed, the logic must pass the verification process. See Verify Logic for more information.

You can download when logic is enabled or disabled. If you download logic when logic is enabled, the download process occurs as follows:

- The old Ladder Editor checks to see if logic is enabled in the device.
- If the logic is enabled, you are asked if you want to disable logic. If you:
  - Select **No** - the logic is not downloaded to the device and an error message displays.
  - Select **Yes** - the logic is downloaded to the device.
- If logic was enabled before you downloaded, when the download is complete, you are asked if you want to re-enable the logic. If logic was not enabled before you downloaded, you are not prompted to enable or disable the logic.
- When the download is complete, you are notified that the download was successful. Note that if the master is currently scanning the device to which you are downloading, you must first access the DeviceNet scanner's scanlist to remove the device from the scanlist. If this new logic results in different connection sizes, you must adjust the master as follows:
  - Put the master in Idle mode (This may mean turning a processor's key switch to Program.)
  - Access the device parameter's to change the size of the connections.
  - Download the device to the scanlist.
  - Click the **Edit I/O Parameters** button and verify the Rx and Tx size corresponds to the new connection size.
  - Put the master in Run mode for normal operation.

You can click the **Cancel** button anytime during the download process to abort downloading the logic to the device. If you click the **Cancel** button, an error message displays and neither old logic nor new logic exists in the device. You must let the download complete for logic to be contained in the device.

If logic is enabled during the download process, once the download is complete, online animation starts. You see the color green on the wires connecting the elements. If logic is disabled, the current state of connections is displayed and inputs can be changed, but online animation does not start.

When you download, the downloaded information is not automatically stored into the RSNetWorx for DeviceNet software .dnt file. You must make sure you save any changes after you exit the old DeviceLogix Editor to ensure that any changes made to the schematic are saved, including changes made as a result of an upload.

Forcing

To aid in the troubleshooting and debugging of your configuration, you can force hardware inputs and outputs. No other inputs or outputs can be forced from within the old Ladder Editor. Network inputs and outputs can be forced in the device from where they originated. If you exit the old Ladder Editor, the forced status will remain in effect until you remove it.

TIP If you force a Digital Output Point (DOP), the Comms Status Override parameter on the Configuration tab in RSNetWorx for DeviceNet software will be set to Override Enabled.

The following list describes the input force options.

Input Force Option	Description
Force On	Forces the input ON.
Force Off	Forces the input OFF.
Remove Force	Returns control of the input to the hardware device and turns the instruction color back to white.

The following list describes the output force options.

Input Force Option	Description
Force On	Forces the input ON.
Force Off	Forces the input OFF.
Remove Force	Returns control of the input to the hardware device and turns the instruction color back to white.
Clear Fault	When a device supports latching of faults, this selection clears a hardware output fault indication (for example, off-wire or short circuit). The actions related to Clear Fault are product specific.

When forced, the instruction turns yellow and the status value on the connection reflects the forced state chosen.

To force hardware inputs and outputs (Force On or Force Off):

1. Right click on the element you want to force on or force off.
2. From the menu, select **Force On** or **Force Off** (whichever action you want to perform).

When a bit is forced, its bit element will have brown text behind it indicating that forcing is on or off.

To remove the Force On or Force Off function:

1. Right click on the element you want to force on or force off.
2. From the menu, select **Remove Force**.

Note that the Force operations can only be done when the configuration is not in the Edit mode.

Clear Latched Hardware Faults

If the device supports latching of faults, the Clear Fault function clears a hardware output fault indication, such as an off wire or short circuit condition.

To clear a fault condition:

1. Right click on the device that you want to clear the fault condition.
2. From the menu, select **Clear Fault**.

Recovery Mode

Access the Recovery Mode window to select between having logic enabled or disabled following the restoration of a configuration. Use the Recovery Mode in conjunction with the Auto Device Replace Configuration Recovery feature.

- When using Allen-Bradley DeviceNet scanners, choose the recovery mode option to determine how the I/O device should recover from an automatic download by a master device.
- When an I/O device fails and a new device is added to replace it, a master device on the network can automatically download the stored configuration and logic to the device. This feature is known as Auto Device Replacement (ADR). The recovery mode determines whether or not to enable the local logic when a download of this type occurs. The recovery mode is disabled by default. ADR in the scanner is also disabled by default.

Navigate the New Ladder Editor Interface

What This Chapter Contains

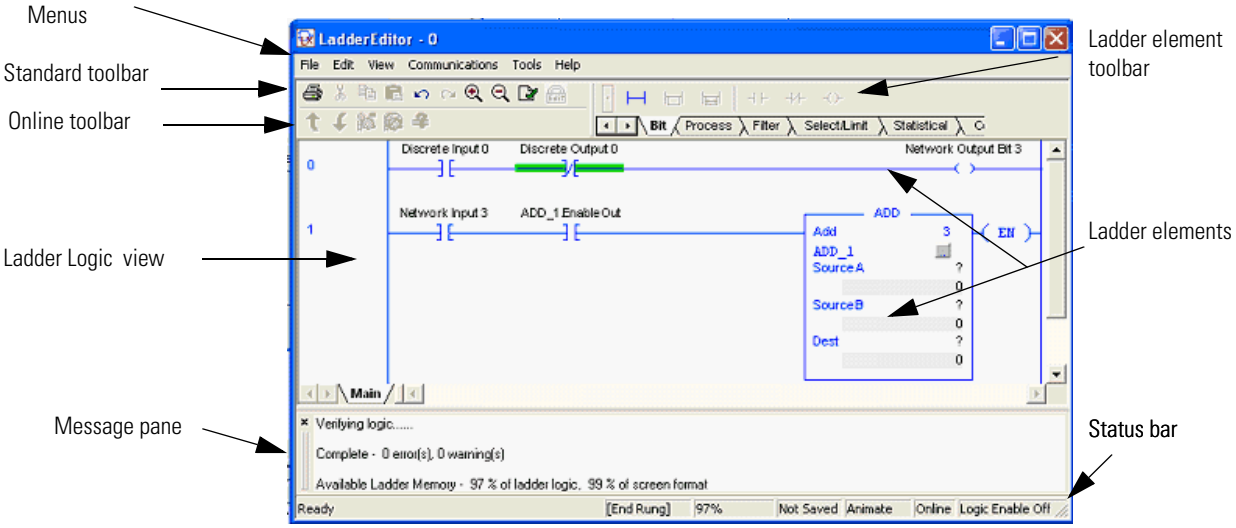
Read this chapter to learn more information about the new Ladder Editor interface. The following table lists what this chapter contains and where to find specific information.

Components

The new DeviceLogix Ladder Editor lets you create the ladder logic you want the device to perform by using DeviceLogix functional elements (rung, branch or branch level and instructions).

The new Ladder Editor tool includes:

- Configuration toolbars
- Ladder logic view
- Status bar
- Message pane
- Menus



Ladder Elements

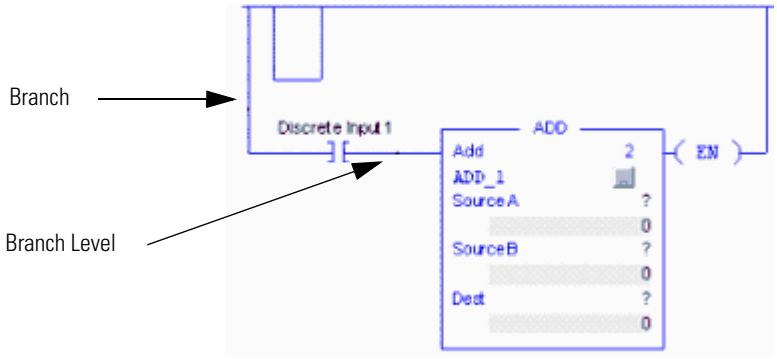
The ladder logic in the new Ladder Editor consists of rungs. The rung consists of functional instruction elements which include bits, latches, counters, and timers. Logic combinations are displayed on the left part of the rung and logic outputs are displayed on the right side of the rung. Logic outputs are determined by the type of instruction (bit, latch, counter, or timer). For each rung, you can enter rung comments which describe the logic that you created.

Ladder elements are:

- Rung
- Bit
- Latch
- Counter
- Timer

Rung Element

Element	Description
	Rung Element - The basic executable unit in the new Ladder Editor. Each rung has at least one input condition and one output condition. The True or False value of an input determines the output value. You can add a comment to a rung to clarify or describe the instruction.
	Branch Element - Connection element that adds OR logic in the new Ladder Editor. The branch element always resides on a rung element and, therefore, cannot exist without the rung element.
	Branch Level Element - Connection element that adds OR logic in the new Ladder Editor. The branch level element can reside on a branch element or a branch level element. The branch level element can also store instructions.



Configuration Toolbars

The new Ladder Editor toolbars include:

- Standard - Lets you perform general editing functions, verify logic, and toggle edit mode.
- Ladder Element - Lets you add ladder elements to the new Ladder Editor. The toolbar changes based on the element you are adding (bit, latch, timer, or counter). You can drag and drop elements from this toolbar into the new Ladder Editor.
- Online - Lets you perform online functions. This toolbar is enabled only when you are working on line.

Standard Toolbar

Function	Description
	Sends the ladder logic to a printer.
	Removes the selected ladder element.
	Duplicates the selected content to save it in the clipboard.
	Removes the saved content in the clipboard and puts the content in the new Ladder Editor where your cursor is resting.
	Cancels the last action.
	Performs again the previously cancelled action.
	Increases (make larger) the view of the configuration. You lose view of part of the configuration pane of the new Ladder Editor when you use this function.
	Decreases (make smaller) the view of the configuration. You see more of the configuration pane of the new Ladder Editor when you use this function.
	Confirms that the logic that you configured is valid.
	Toggles between working in the Edit mode and not working in the Edit mode. In the Edit mode, you can make changes to the Logic. You must exit the Edit mode to download the logic to the device.

Tabbed Instruction Toolbar

The ladder element toolbars include:

- Bit Element
- Process Element
- Filter Element
- Select/Limit Element
- Statistical Element
- Timer/Counter Element

- Compare Element
- Compute/Math Element
- Misc Element

TIP It is not required for all DeviceLogix-enabled products to support all of the element types. The EDS file for each product provides information on what element are supported. If an element is not supported, it will not appear in the associated toolbar.

Regardless of the ladder element toolbar you have selected, the following icons are always available.

Function	Description
	Rung Element
	Branch Element
	Branch Level Element

Bit Element Toolbar

Function	Description
	Examine If Open (or XIO) Element
	Examine If Close (or XIC) Element
	Output Coil Bit (or Output Energize Bit) Element

Process Element Toolbar

Function	Description
	Alarm Element
	Time Diagnostic Element
	PID Element

Filter Element Toolbar

Function	Description
LPF	Low-Pass Filter Element - The total number of this instruction type that you can use is limited due to resources and is device-specific.

Select/Limit Element Toolbar

Function	Description
SEL	Select Element
HLL	High-Low Limit Element

Statistical Element Toolbar

Function	Description
MAVE	Moving Average Element - The total number of this instruction type that you can use is limited due to resources and is device-specific.

Timer/Counter Element Toolbar

Function	Description
TONR	On-Delay Timer Element
TOFR	Off-Delay Timer Element
PULR	Pulse Timer Element
CTU	Up Counter Element
CTUD	Up Down Counter Element
RES	Timer/Counter Reset Element

TIP You can apply the Up Counter and Down Counter instructions to the same tag name to create an Up-Down Counter.

TIP When using the ACC feature, you may notice some minor differences in the Timer/Counter instructions. Depending on the firmware implementation, some devices consider this an output and bind it to an analog output tag, while other devices consider it a parameter.

Compare Element Toolbar

Function	Description
MEQ	Mask Element
EQU	Equal Element
NEQ	Not Equal Element
LES	Less Than Element
GRT	Greater Than Element
LEQ	Less Than or Equal To Element
GEQ	Greater Than or Equal To Element

Compute/Math Element Toolbar

Function	Description
ADD	Add Element
SUB	Subtract Element
MUL	Multiply Element
DIV	Divide Element
MOD	Modulus Element

Function	Description
SQR	Square Root Element
NEG	Negative Element
ABS	Absolute Element

Misc. Element Toolbar

Function	Description
SETD	Set Dominant Latch Element
RSTD	Reset Dominant Latch Element
RES	Reset Latch Element

For detailed descriptions of instruction behaviors, refer to Chapter 2, Navigate the Function Block Editor Interface.

Online Toolbar

Use the Online toolbar to perform functions when you are working on line in the new Ladder Editor.

Function	Description
	Upload the logic configuration from the device to the new Ladder Editor configuration tool.
	Download the logic configuration from the new Ladder Editor to the device. The configuration must pass the logic verification process for the download to be successful.
	Run the DeviceLogix logic configuration that you have downloaded to the device.
	Stop the DeviceLogix logic configuration running in the device.
	Compare the logic in the device with that in the new Ladder Editor configuration tool.

TIP Some host software programs control the upload and download communications, so these functions are not available to you in the new Ladder Editor.

Ladder Logic View

The Ladder Logic View is the area in which you place functional elements to create logic.

The printed ladder logic uses A4 sheets of paper. The ladder logic is printed at full size, regardless of the setting of the current zoom level. To make adjustments to the ladder logic before printing, you can use the print preview feature to see how it will print. The ladder can also be sent to a plotter for easier viewing.

TIP Because a device has limited memory to store logic, the display of uploaded information may not match the appearance of information downloaded to the device. Therefore, once your logic configuration is complete, you should print a copy of it for your records. Additionally, Rockwell Automation also suggests that you save the project file.

Message Pane

The Message Pane on the bottom of the new Ladder Editor window displays the results when you verify logic by clicking the **Verify** button in the toolbar or by choosing **Tools > Logic Verify**. If the new Ladder Editor finds an error in your logic, click on the error or warning message in this pane, and the cursor moves to the place in the new Ladder Editor where the error appears in the logic.

The Message Pane also displays the amount of memory that is available in the device (only after you verify the logic).

From the View menu, you can toggle between displaying the Message Pane or not displaying it.

Status Bar

The Status bar, located on the bottom of the new Ladder Editor, provides a view of the current working status of the new Ladder Editor. You can toggle between displaying the Status Bar and not displaying it by selecting **View > Status Bar**.

Six panes provide unique information about the new Ladder Editor.

Pane	Description
Rung Index	Displays the rung the cursor is resting on and the total number of rungs in the configuration.
Available Logic Memory Resource	Indicates the available percentage of memory for the ladder logic. When available memory is less than 5%, Low Memory will be displayed.
Logic Saved Indication	Indicates whether or not the current logic has been saved in the host software's project file.

Pane	Description
Edit Mode	Refer to table below.
Online Indication	Indicates if you are working online or offline in the new Ladder Editor.
Logic Enabled Indication	Indicates if the Logic Enable function is on or off. If you are not in Edit mode and go online with a device, then it displays Logic Enabled status.

Edit Mode

Status	Editor Mode	Description	Pre-Conditions
Online	Edit Enabled	You can modify logic.	Read-write status. Edit toolbar button is toggled ON.
	Animate	Logic in the new Ladder Editor is synched with logic in the device.	Edit toolbar button is toggled OFF.
	Pending Edit	Logic in the new Ladder Editor is not synched with logic in the device	Edit toolbar button is toggled OFF.
Offline	Edit Enabled	You can modify logic.	Read-write status. Edit toolbar button is toggled ON.
	Edit Disabled	You cannot modify logic.	Read-only or online-edit status. Edit toolbar button is toggled OFF.

Menus

Use menu options to create and maintain ladder logic. Although the toolbar provides convenient shortcuts to many of the features available in the menus, not all of the menu functions are available from the toolbar. The new Ladder Editor provides six menus, each of which is described with its features in the sections that follow.

- File
- Edit
- View
- Communications
- Tools
- Help

File Menu

The File menu lets you perform printing functions and lets you exit the new Ladder Editor.

Function	Description
Print	Sends the ladder logic to a printer.
Print Preview	Displays the ladder logic before sending the logic to the printer.
Print Setup	Allows you to choose printer and printing options.
Close	Exits the new Ladder Editor.

Because the new Ladder Editor is not a standalone application but rather is part of the host software, this File menu does not provide options for creating, opening, or saving new ladder logic configurations.

Edit Menu

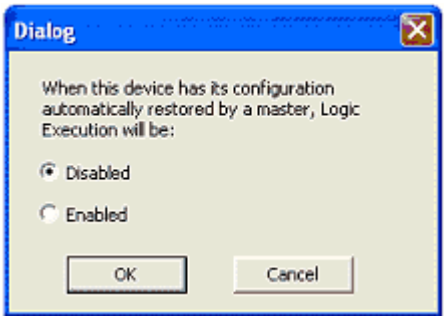
The Edit menu lets you modify elements in the new Ladder Editor.

Function	Description
Undo	Cancels the last action.
Redo	Performs again the previously cancelled action.
Cut	Removes the selected ladder element.
Copy	Duplicates the selected content to save it in the clipboard.
Paste	Places the content in the clipboard into the Ladder Editor at the position that your cursor is resting.
Select All	Highlights all of the elements in the Ladder Editor (to copy or cut). This menu option has different behavior depending upon the current selection in the Ladder Editor: • When a rung is selected, Select All selects all of the rungs. • When a branch leg is selected, Select All selects the entire branch. • When a branch level is selected, Select All selects the entire branch level.
Add Ladder Element	Insert a ladder element at the place where the cursor is positioned. The Add Ladder Element window opens for you to select the type of element you wish to insert.
Delete Ladder Element	Removes the selected element from the Ladder Editor.

Function	Description
Edit Ladder Element	Modifies the selected ladder element by you changing the instruction type or the instruction type parameters. Note that it is possible to change only instruction types of the same category (bits, latch, timers, counters).
Edit Element	Modifies the selected ladder element. It is possible to change the instruction type or the instruction type parameters. Note that it is possible to change only instruction types of the same category (bits, latch, timers, counters).
Edit Element Comment	Modifies the comment attached to the selected rung or instruction type. Note that you cannot add comments to branch or branch level elements.
Properties	Access the Configuration Properties dialog to modify or add configuration information.
Recovery Mode	Accesses the Recovery Mode dialog to select between having logic enabled or disabled following the restoration of a configuration. The Recovery Mode is used in conjunction with the Auto Device Replace Configuration Recovery feature.
Logic Enable Selection Mode	Opens the Logic Enable Selection dialog box, where you can indicate your preferred behavior for enabled logic running in the device.

Recovery Mode

If the host software supports Recovery Mode, choose **Edit > Recovery Mode**. In the Recovery Mode dialog box, you can choose to enable or disable the recovery mode.

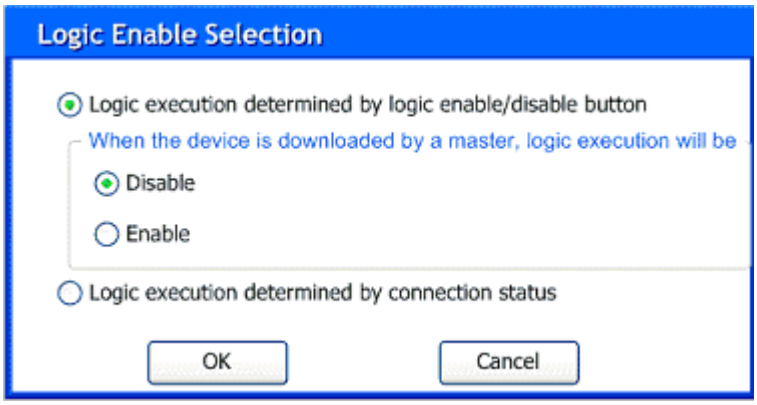


When using the Allen-Bradley master, choose the recovery mode option to determine how the I/O device should recover from an automatic download by a master device.

When an I/O device fails and a new device is added to replace it, a master device on the network can automatically download the stored configuration and logic to the device. This feature is known as Auto Device Replace (ADR). The recovery mode determines whether or not to enable the local logic when a download of this type occurs. The recovery mode is disabled by default. ADR in the scanner is also disabled by default.

Logic Enable Selection Mode

If the host software supports the Logic Enable Selection Mode, choose **Edit > Logic Enable Selection Mode**. In the Logic Enable Selection dialog box, indicate your preference for enabling logic in the device.



If you want the logic execution to be determined by the logic enable/disable button, select the appropriate radio button and either:

- select Disabe to add a stop logic service, or
- select Enable to add a run logic service.

If you want the logic execution to be determined by the connection status, select the appropriate radio button. You will not be permitted to run or stop the logic in the new Ladder Editor; you will be able to monitor the status of the logic in the device.

View Menu

The View menu lets you modify your view of the new Ladder Editor.

Function	Description
Standard Bar	Toggles between displaying and not displaying this toolbar.
Online Bar	Toggles between displaying and not displaying this toolbar.
Instruction Bar	Toggles between displaying and not displaying this toolbar.

Function	Description
Status Bar	Toggles between displaying and not displaying the status bar at the bottom of the new Ladder Editor.
Message Log	Toggles between displaying and not displaying the Message Pane at the bottom of the new Ladder Editor.
Zoom In	Increases (make larger) the view of the configuration. You lose view of part of the configuration pane of the new Ladder Editor when you use this function.
Zoom Out	Decrease (make smaller) the view of the configuration. You see more of the configuration pane of the new Ladder Editor when you use this function.
Zoom to 100%	Causes the configuration pane to mirror what is seen on the printed version when the configuration pane is printed.

Communications Menu

The Communications menu lets you work online in the new Ladder Editor.

Function	Description
Upload	Reads the logic configuration in the device and displays it in the new Ladder Editor. You lose any changes you made in the configuration, and online animation starts.
Download	Transfers the logic configuration in the new Ladder Editor to the device and commences online animation. For the download to begin, the configuration must pass the verification process. If logic is enabled in the device, you will be prompted to disable it. If you do not disable the logic in the device, then the download process will stop, and you will receive an error.
Logic Enable Off	Stops running the logic configuration that you have downloaded to the device. This option is grayed-out when the logic configuration in the new Ladder Editor does not match the logic configuration in the device.
Logic Enable On	Runs the logic configuration that you have downloaded to the device. This option is grayed-out when the logic configuration in the new Ladder Editor does not match the logic configuration in the device.

Note that these communication options are only available when you are working in the online mode.

TIP

In some host software, the Logic Enable status is determined by the connection status. Choose **Edit > Logic Enable Selection Mode** to indicate your preferred behavior for enabled logic running in the device.

Tools Menu

The Tools menu lets you modify logic in the new Ladder Editor.

Function	Description
Logic Verify	Confirms that the logic that you configured is valid. Confirms that: • each rung has a minimum input and output element • each branch has a minimum element • each element has the minimum number of bound pins • each attribute has the correct data type associated with it • the two ends of a binding are of the same type • the logic and its screen format do not exceed the memory limit in the device
Compare	Compares the logic configured in the device with the logic configured in the Lnew adder Editor. You must be working in the online mode to use this function.
Edit Mode	Toggles between working in the Edit mode and not working in the Edit mode. In the Edit mode, it is possible to modify the logic but you cannot communicate with the device. (That is, you cannot upload or download logic or use the Logic Enable On or Logic Enable Off functions.) When you are not working in the Edit mode, it is possible to perform online animation. If you are working in online mode, it is possible to set the preset value or accumulator value of counters or timers, but you cannot modify logic.
Show Tag Database	Displays the Tag Database, which provides details about product scope tags, program scope tags, and macro scope tags.
Screen Format	Opens the Screen Format Download Option dialog box, where you can indicate the portions of the logic configuration that you want to be downloaded to the device.
Macro Block	Allows you to create and manage Macro types within your logic configuration through several sub-menu options: • Create Macro Block • Macro Block Manager • Open Macro Block Definition • Delete Macro Block • Open Macro Block Logic

Help Menu

The Help menu provides you with assistance when you are working in the new Ladder Editor.

Function	Description
Ladder Editor Help	Accesses the help files available for the new Ladder Editor.
Release Notes	Accesses release notes pertaining to DeviceLogix functionality or the new Ladder Editor.
About DeviceLogix	Opens the About Ladder Editor window to learn revision and copyright information about the new Ladder Editor. File revision identifies the revision of the Ladder Editor DLL application.

Create Logic in the New Ladder Editor

What This Chapter Contains

Read this chapter to learn more about the creating and managing logic in the new Ladder Editor. The following table lists what this chapter contains and where to find specific information.

About I/O Tags

There are two kinds of I/O tags: static and dynamic.

Static tags include:

- Device Input (DIP) - physical input point of the device
- Consumed Network Bit (CNB) - network input bit
- Fault Bit (FB)- device fault bit
- Status Bit (SB) - device status bit
- Device Output (DOP) - physical output point of the device
- Produced Network Bit (PNB) - network output bit

Dynamic tags are those created dynamically during configuration. They are the referenced output bits of latch, counter, and timer elements. Using reference bits can simplify your ladder logic.

Working with the Tag Database

Devices have two kinds of tags:

- product scope tag
- program scope tag

If you edit Macro logic, then the product scope tag will be replaced with the Macro scope tag.

Product scope tags are the product I/O defined in DeviceLogix. You can assign alias names to product scope tags. In the logic configuration, only alias tag names are used. Alias tag names are not case-sensitive.

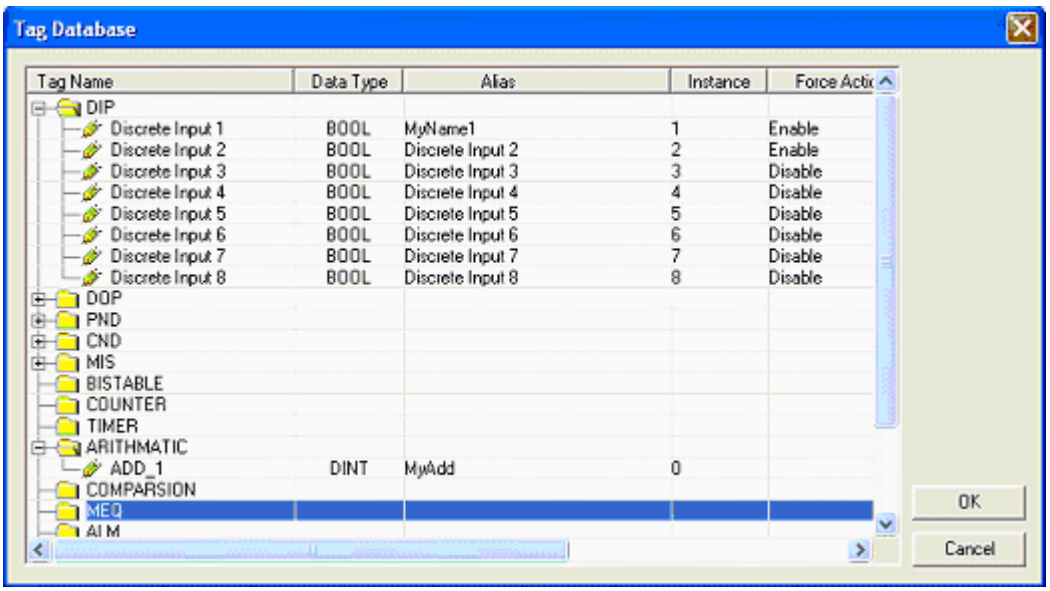
Product Scope Tags

I/O Data Group	I/O Data	Data Type	Reference Rule
Hardware Input	Device Input (DIP)	Bit	Input
	Analog Input (AIP)	DINT/REAL	Input
Hardware Output	Device Output (DOP)	Bit	Output/Input
	Analog Output (AOP)	DINT/REAL	Output/Input
Soft Input	Fault Bit (FB)	Bit	Input
	Consumed Network Data (CNB)	Bit/DINT/REAL	Input
	Status Bit (SB)	Bit	Input
Soft Output	Produced Network Data (PND)	Bit/DINT/REAL	Output/Input

Use Bit type tags for input contact or output coil binding, depending on the tag type. Use Analog input type tags only in instruction input binding. You can use Analog type output tags in instruction output binding, as long as the data type matches.

Use the program scope tag when you want an instruction to reference the results of other instructions. It is a "soft" tag and is created dynamically during the logic editing.

When you add an instruction to your logic, all outputs of this instruction are added to the Tag Database automatically. Choose **Tools > Show Tag Database** to open the window. In the Tag Database, you may modify the alias names of the new tags, edit forcing options (if the device supports forcing), select data types, and select fault options.



When you add an instruction to your logic, a tag name is automatically assigned (its default alias name), following the naming rules: [InstructionName_Index], where Index is numeric and automatically increases by 1 when you add a new instruction of the same type to your logic. If you delete an instruction from your logic, this Index number becomes available for new instructions (of the same type) that you add to your logic.

Output tag names are assigned as follows: [Instruction name.Output Name]. For example, Timer_1.DN refers to the output for Timer 1.

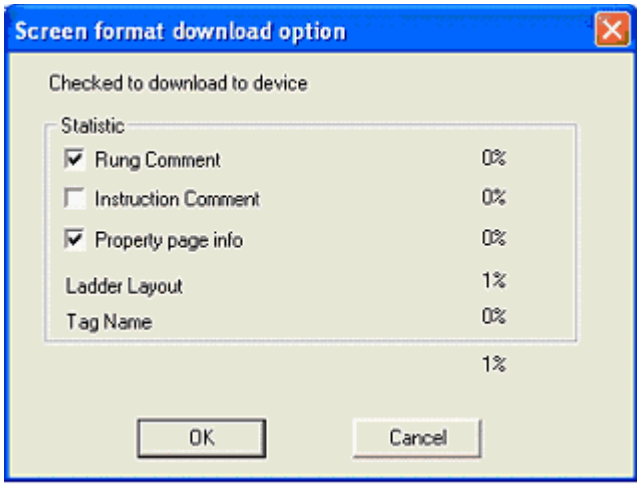
About Screen Format Elements

During the download process, the new Ladder Editor downloads all of the logic configuration data to the device, including:

- Logic
- Ladder layout
- Tag Database information
- Rung comments and instruction comments
- Properties page content

The first three items - logic, ladder layout, and Tag Database information - are required for download to the device. If the device memory cannot store these items, then an error message will appear during logic verification, indicating that the download cannot be completed. The last two items - comments and Properties page content - are optional for download to the device. If the device memory cannot store these items, then you can opt not to download them.

Choose **Tools > Screen Format** to open the Screen Format Download Option dialog box, where you can indicate which items you want to include in the download. After you complete the logic verification, this dialog box shows the percentage of total screen format memory that is used by each item; if the device does not have enough memory to store all of the items, use this information to help you determine which items to download.



Configure the Macro Instruction

Beginning with DeviceLogix revision 4, certain devices support the Macro instruction, which defines a set of logic routines. Once defined, the Macro instruction behaves like other instructions in logic.

A Macro definition contains an instruction and may reference any output inside this logic and wiring. It does not contain any product scope I/O. The name of a Macro definition cannot duplicate any existing DeviceLogix instruction name or another existing Macro definition name.

The maximum number of Macro definitions permitted is defined in the DeviceLogix capability information, which is included in the EDS files.

Work with the Macro Instruction

You can access, edit, and manage Macro instructions through new Tools menu options:

- Create Macro Block
- Macro Block Manager
- Open Macro Block Definition
- Delete Macro Block
- Open Macro Block Logic

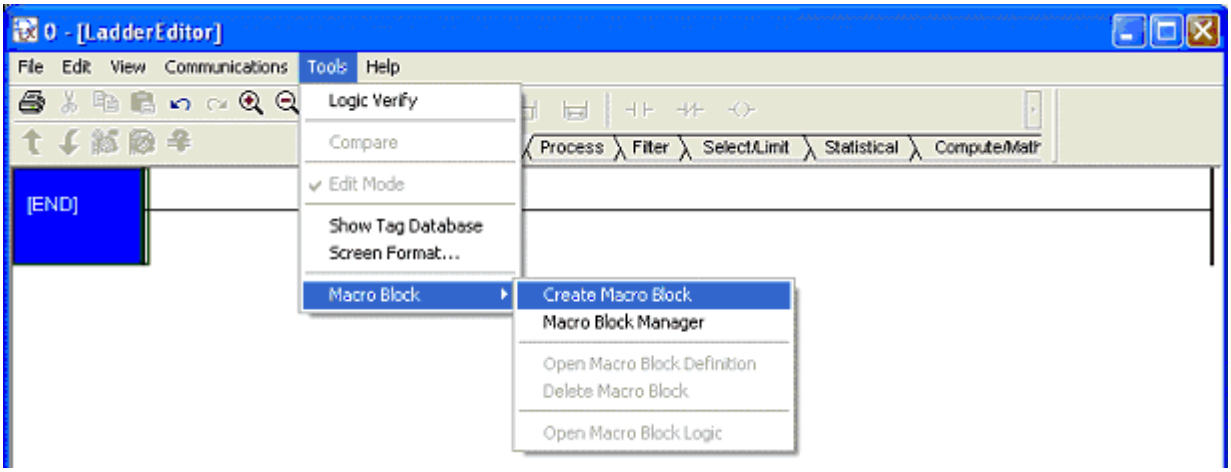
Using a Macro instruction encompasses these general steps, each of which is described in following sections.

1. Create the Macro definition
2. Define the external I/O binding
3. Add parameters
4. Set security (optional)
5. Apply the Macro definition
6. Add the Macro instruction to ladder logic
7. Manage the Macro

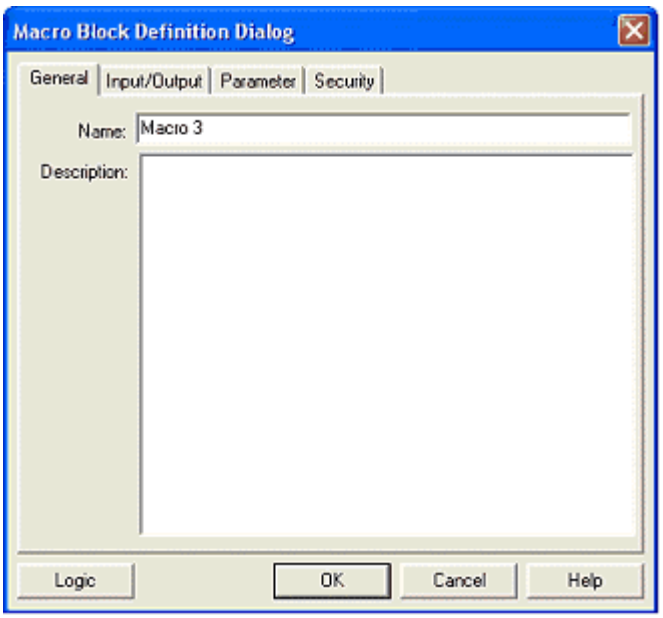
Create the Macro Definition

Complete these steps to create a new Macro definition.

1. Choose **Tools > Macro > Create Macro Block**.



The Macro Block Definition dialog box appears, containing General, I/O, Parameter and Security tabs.



2. On the General tab, assign a unique and valid Name for the Macro block.

   Valid names are alpha-numeric and are limited to 16 characters.

3. Enter a Description.

4. Click **OK**.

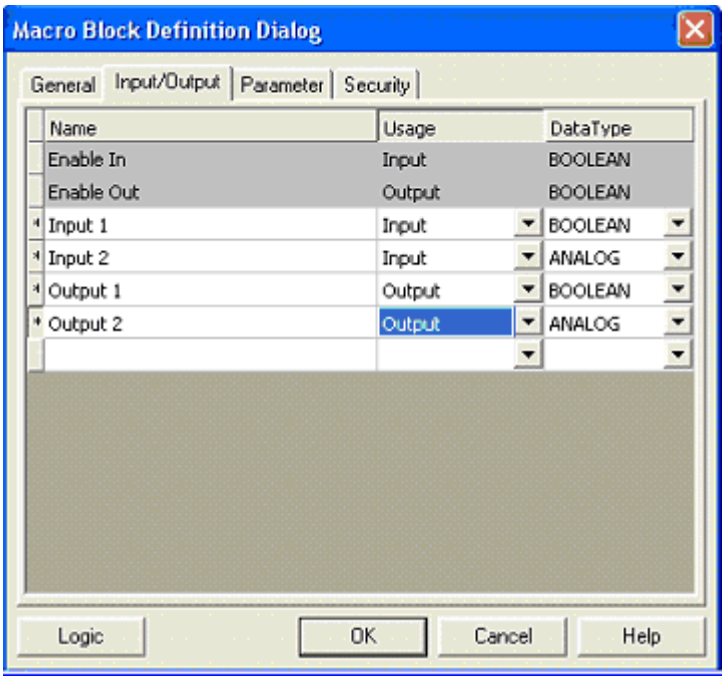
   If the Name you assigned is not unique, you will be prompted at this time to enter a different Name.

Define the External I/O Binding

To definte the input/output bindings of your new Macro block, complete these steps.

1. Choose **Tools > Macro > Open Macro Block Definition**.

2. Click the Input/Output tab.



3. Assign a Name for the I/O binding.

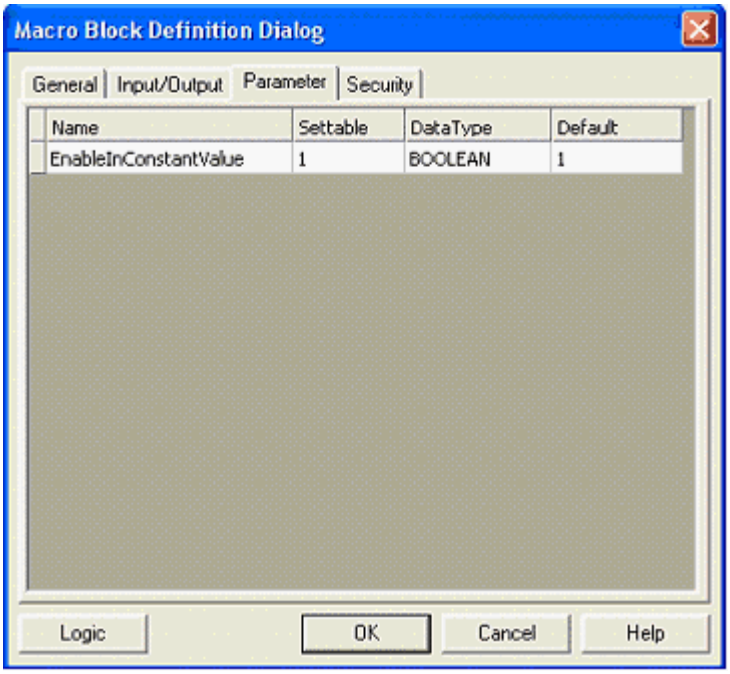
4. Select the Usage.

5. Select the Data Type.
   - BOOL: the defined I/O can be bound only with Boolean data.
   - ANALOG: the defined I/O can be bound with a wide class of basic data types.

6. Click **OK**.

Add Parameters

While the Parameter tab in the Macro Block Definition dialog box allows you to edit parameters, you cannot add Macro instruction parameters here. You add parameters when you add an instance of the Macro instruction type to your ladder logic.

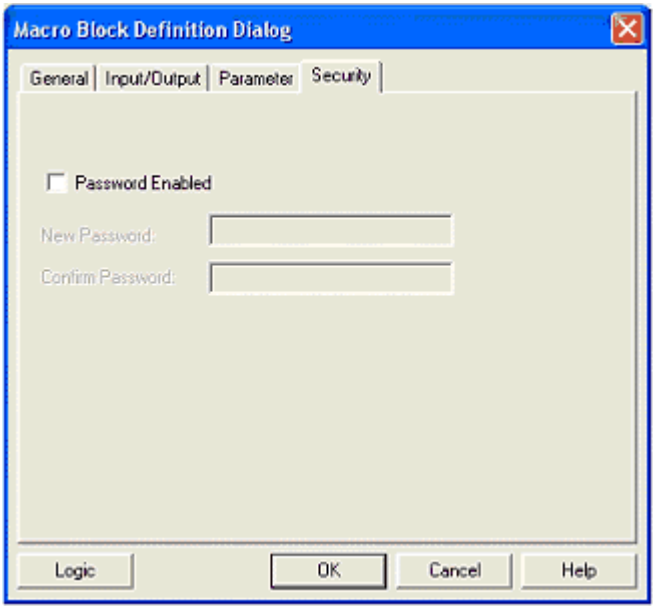


Set Macro Security

Password protection provides the capability for Macro Block designers to protect definitions and parameters. You may enable Macro security when you create the Macro block; by default, password protection is disabled on a new Macro block. The security scope extends to each instance of the Macro block that you add to the ladder logic.

To enable security on a Macro block, complete these steps.

1. Choose **Tools > Macro > Open Macro Block Definition**.
2. Click the Security tab.



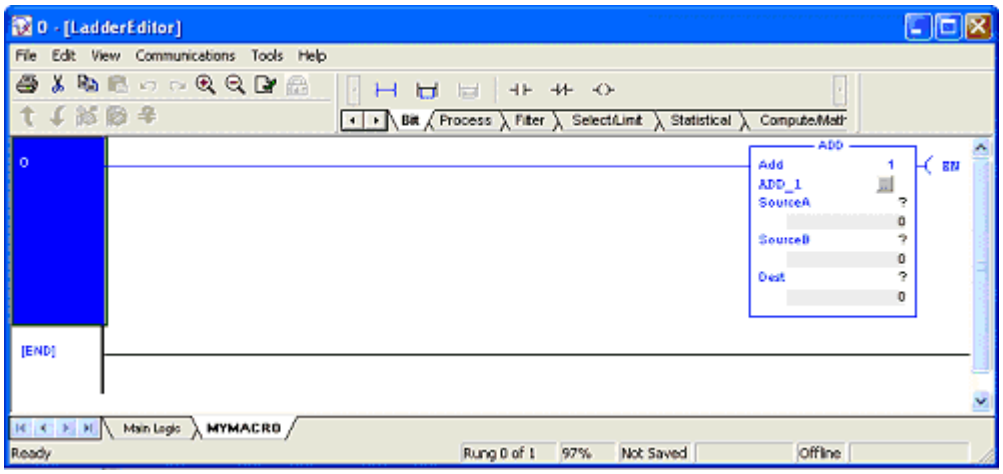
3. Select the Password Enabled box.
4. Type the New Password.
5. Confirm the password.
6. Click **OK**.

When editing security-enabled Macro definitions or parameters, you will be prompted to enter the password.

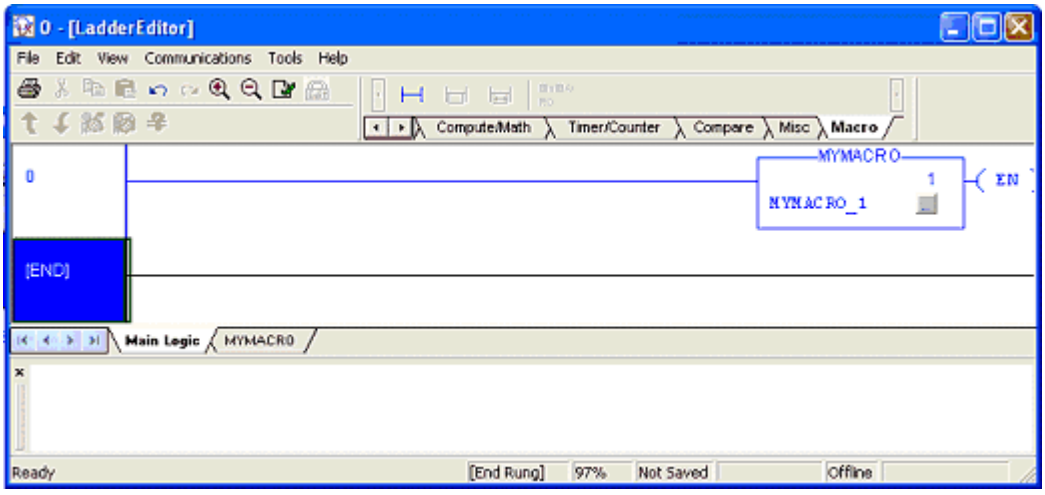
IMPORTANT If you forget the password for the Macro block, you will not be able to access it. In this situation, you must create a new Macro block to overwrite the locked one.

Apply the Macro Definition

When you click **OK** on any tab in the Macro Block Definition dialog box, your edits will be applied if the logic is valid. When you click **OK** or **Logic** on this dialog box, a tabbed editing window for the new Macro block displays. To close this tabbed window, right-click the tab, and choose Close.



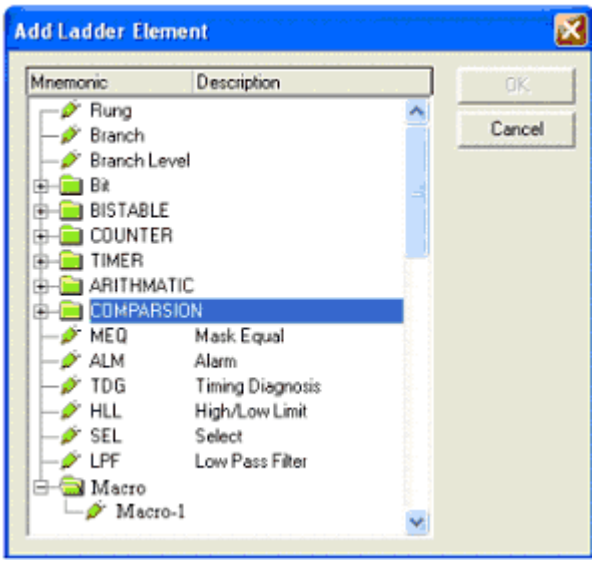
Your newly-defined Macro instruction will appear in the new Macro tab in the Instruction toolbar, along with other defined Macro instructions. If you have not created and defined a new Macro instruction, then this tab will be empty. If a Macro instruction is invalid or has not been verified, then it will be grayed-out in the Macro tab.



Add Macro Instruction to Ladder Logic

Add the Macro instruction into your ladder logic configuration just as you would add a standard instruction.

- Choose **Edit > Add Ladder Element**, select the Macro instruction element in the Add Ladder Element dialog box, and click **OK**.

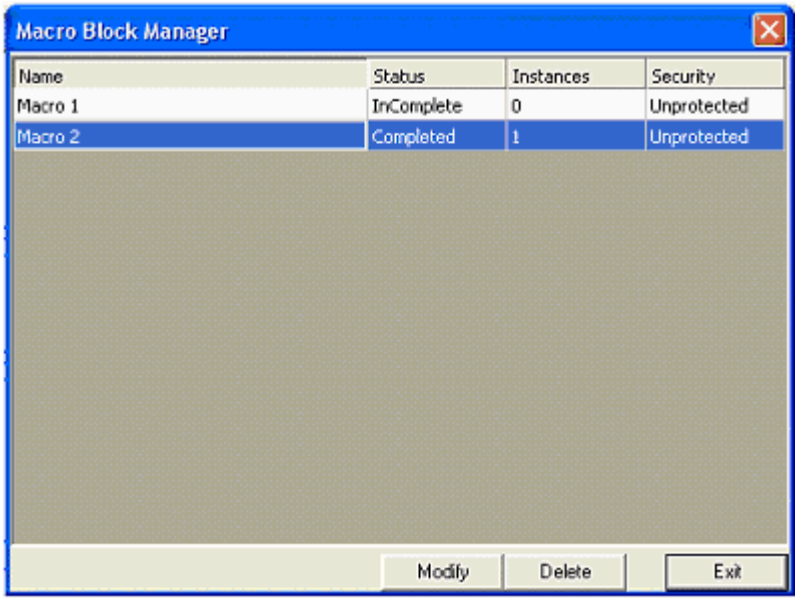


- Click the instruction icon in the Macro tab of the Instruction toolbar.
- Drag the instruction from the Macro tab of the Instruction toolbar and drop it into the ladder logic.

Only valid Macro definitions will appear in the Instruction toolbar and in the Add Ladder Element dialog box.

Manage the Macro

Choose **Tools > Macro Block > Macro Block Manager** to open the Macro Block Manager dialog box, where you can select Macro blocks to modify or delete.



You can delete a Macro block only if there are no instances of the Macro in the ladder logic. You cannot undo a Macro block deletion.

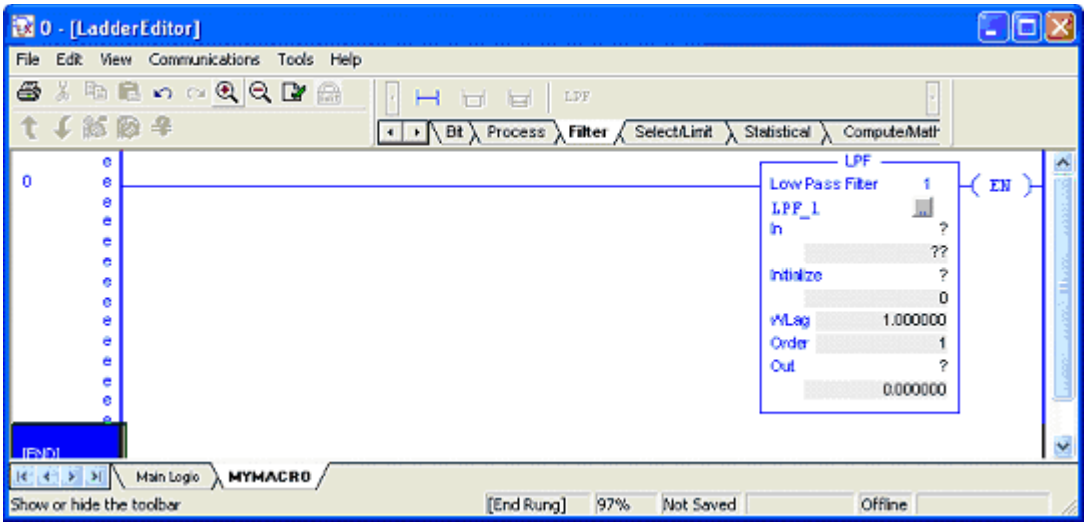
Edit the Macro Definition and Parameters

Once you have created a new Macro definition, you can choose **Tools > Macro > Open Macro Definition** to edit the Macro definition and parameters. In addition, with the Macro tabbed window open, you can edit logic for this Macro instruction just as you would a standard instruction.

When you are editing a Macro instruction, the Tag Database will list information specific to this Macro only. Only the inputs and outputs of this Macro scope and any instructions used in this Macro will appear in the Tag Database.

To edit a Macro's parameters, complete the following steps.

1. In the ladder logic, click the Properties button for the target instructions.

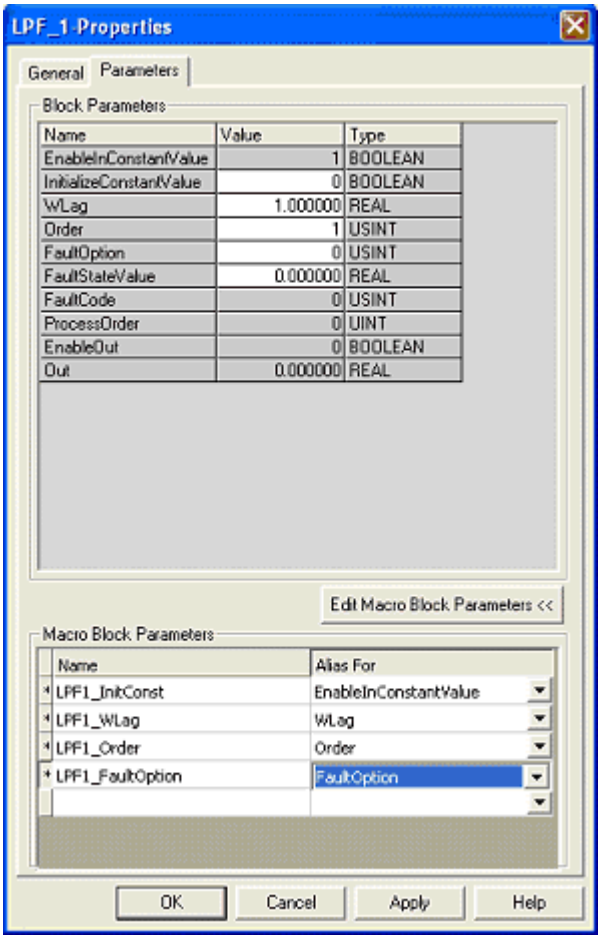


The Properties dialog box displays.

2. Click the Parameters tab.

3. Click the Edit Macro Block Parameters button.

The dialog box expands to show the list of externally-viewable Macro Block Parameters.

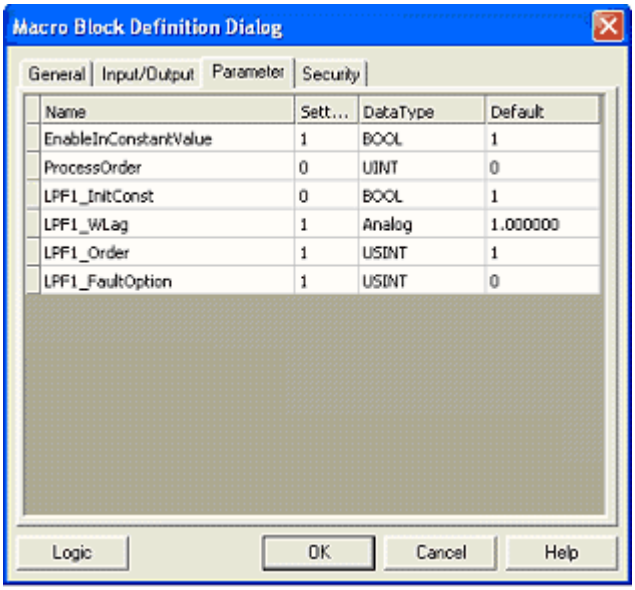


4. Assign a unique Name for the Macro block parameter.

5. Select the block parameter that this Macro block parameter references from the Alias For drop-down list.

6. Click **OK**.

To view the Macro block parameters, choose **Tools > Macro > Open Macro Definition**, and click the Parameter tab.



Consider these guidelines when editing the Macro definition.

- Nested Macro block configuration is not supported. While you configure Macro block logic, all Macro block instructions in the Instruction toolbar are grayed-out so that you cannot add a Macro within a Macro definition.
- You may copy and paste logic from the main logic window to the Macro tabbed window, assuming that the copied logic doesn't contain any Macro instructions. When you copy and paste from the main logic window, all I/O bindings will be removed.
- You may copy and paste logic from the Macro tabbed window to the main logic window. When you copy and paste from the Macro tabbed window, all I/O bindings will be removed.
- Copying and pasting logic will set all contact, coil, or analog I/O to unbound.

Interface Changes Related to Macro Instruction

Three areas of the new Ladder Editor interface behave differently with Macro instructions than with standard instructions:

- Instruction toolbar
- Tag Database
- Edit menu

Instruction Toolbar and Tag Database

- When you create or edit a Macro definition, product scope I/O and other Macro instructions are not available for use in the definition.
- The I/O defined in the Input/Output tab are called Macro scope I/O and are available for use in the Macro.
- Some standard instructions have individual limits for usage within the ladder logic. For example, you can use only three instances of a MAVE instruction, even if the device supports 100 instances. These limits also apply when you use these instruction types in Macro definitions. Once you reach the usage limit for a given instruction type, the instruction will be grayed-out.

Edit Menu

When you add and edit Macro instructions, the information displayed on the Add Ladder Element dialog box depends on what you are editing.

- If you are editing a Macro definition, only Macro scope I/O are listed. No product scope I/O are available for use in Macro definitions.
- If you are editing the main ladder logic, the Macro instruction types and their instances are listed, but Macro scope I/O are not.

Online Operations

Edit Parameter Values When Online

When the device is online, you cannot edit the logic. However, you can modify certain parameter values and apply them while the logic is running, including the Timer's and Counter's Preset (PRE) and Accumulator (ACC).

To modify a parameter value while the device is online, complete these steps.

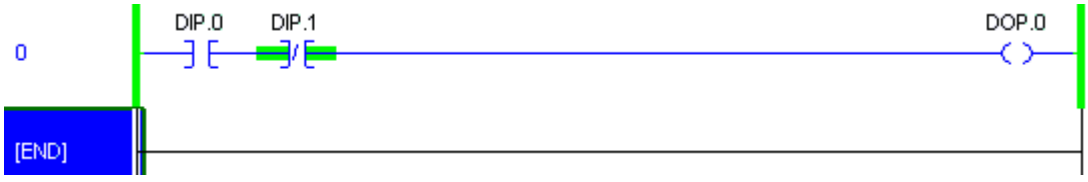
1. Double-click the instruction that you want to change.
2. Select the Parameters tab.
3. Locate the value that you want to change (fields that can be edited are white). Enter the new value in the box.
4. Click **OK**.

The new value takes immediate effect. If you change a preset, remember to save the logic configuration the next time you exit the new Ladder Editor.

Online Animation

When the logic in the new Ladder Editor is synched with the logic in the device, the device is online, and the logic is enabled, the device enters online animation mode.

When the device is in online animation mode, all input, output, and parameters will update periodically. You can monitor the data in the new Ladder Editor during online animation.



Enable and Disable Logic

Use the Logic Enable On function to notify the device to start processing the stored logic diagram. This turns on the Logic Enable Bit in the produced data. Once connections are complete, you can verify the logic. To set Logic Enable On, choose **Communications > Logic Enable On**.

Use the Logic Enable Off option to notify the device to stop executing the logic diagram. When you select this option, the device enters an idle state, turning off the Logic Enable Bit in the produced data. The Logic Enable Bit in the Produced I/O assembly of the device reflects a 0 or 1 for Logic Enable Off or Logic Enable On. To set Logic Enable Off, choose **Communications > Logic Enable Off**.

For some host software, the logic enable is determined by the connection. For more information, refer to Chapter 6, Navigate the New Ladder Editor Interface.

Verify Logic

Once you have created your logic, you must verify your configuration before downloading it to the firmware in the device. Click **Verify** in the toolbar, or choose **Tools > Logic Verify** to verify the logic syntax.

- Each rung has minimum input and output instruction.
- Each branch has minimum instruction.
- Each instruction has the minimum number of pins bound.
- Each parameter has the appropriate data type associated with it.
- Both ends of a binding are of the same type.
- Device has enough memory to store the logic and screen format data.

If any of the logic syntax fails, the Message Pane displays corresponding error messages. You cannot download the logic to the device until the logic passes the verification without error.

If the logic syntax and the screen format data pass the verification without error, the Message Pane displays the actual memory usage for logic and screen format.

If the logic syntax passes verification, and if both the logic layout and Tag Database can be downloaded to the device, but the other screen format data cannot be downloaded to the device, the Message Pane displays corresponding error information. If you want to change which items to include in the download, choose **Tools > Screen Format**.

Upload and Download Logic

Upload Logic

To upload logic, select **Communications > Upload**. The logic configuration in the device is read and displayed in the new Ladder Editor. Any unsaved changes will be discarded, and Online Animation begins.

Note that some devices are capable of storing screen format information. If a particular DeviceLogix-enabled device supports this feature (dependent on the static memory size), the function block logic and layout position information is stored and will be returned with an upload. Further, the text comments are also saved, but the quality of the content cannot be guaranteed.

When you upload, the uploaded information is not automatically stored into the RSNetWorx for DeviceNet software .dnt file. You must make sure you save any changes after you exit the new Ladder Editor to ensure that any changes made to the schematic are saved, including changes made as a result of an upload.

Download Logic

To download logic, select **Communications > Download**. The logic configuration in the new Ladder Editor is duplicated and transferred to the device. Before a download can be performed, the logic must pass the verification process.

You can download when logic is enabled or disabled. If you download when logic is enabled in the device, the download process occurs as follows:

1. The new Ladder Editor checks to see if the logic is enabled in the device.

2. If the logic is enabled, you are prompted to disable the logic.
   - If you click **NO**, the logic is not downloaded to the device, and an error message displays.
   - If you click **YES**, the logic is downloaded to the device.

3. If the logic was enabled before the download, when the download completes, you are prompted to re-enable the logic. If logic was not enabled before the download, you are not prompted to enable or disable the logic.

4. When the download is complete, you are notified that the download was successful.

IMPORTANT If you receive no response from the new Ladder Editor, cycle power.

You can click **Cancel** anytime during the download process to abort downloading the logic to the device. If you click **Cancel**, an error message displays, and neither the old logic nor the new logic exists in the device. You must perform a complete download in order for the device to contain and execute logic.

Once the download is successful, the new Ladder Editor is able to animate the logic. During online animation, the wires connecting the elements in the configuration display real-time values. In addition, parameters and other instruction output values are updated with their real-time values in the device. During online animation, the new Ladder Editor displays updated information approximately every 500 milliseconds.

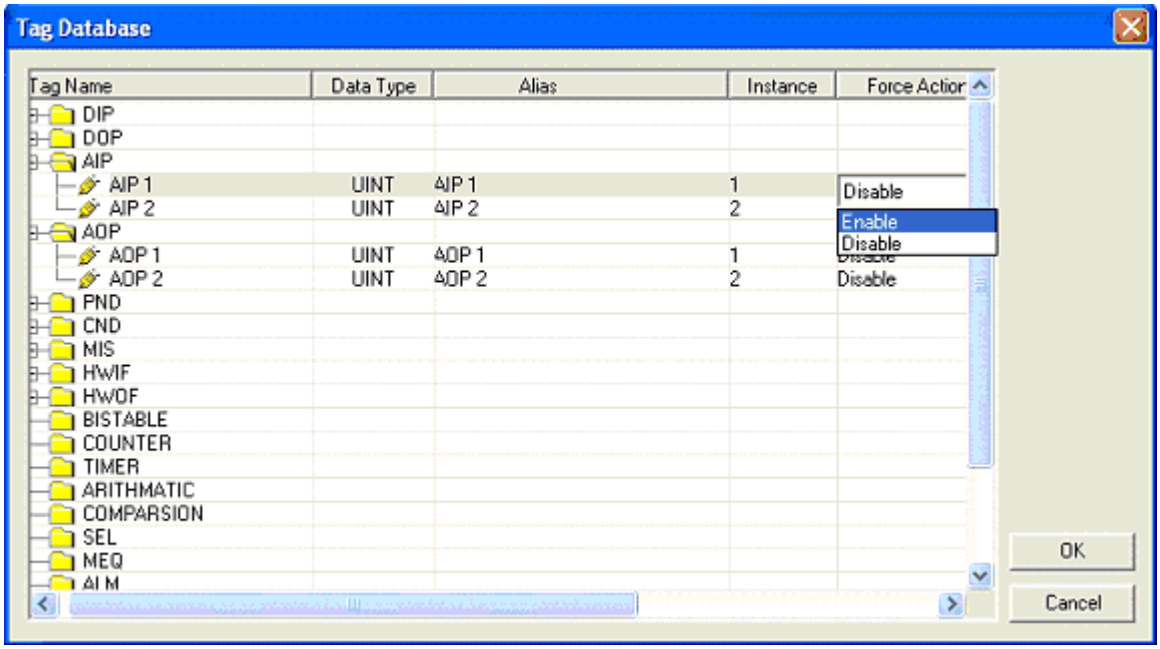
If the logic was disabled after the download, then there is no activity after the download completes. The current state of connections is displayed, and you can change inputs.

Force Inputs and Outputs

To aid in troubleshooting and debugging of your schematic, you can force hardware inputs and hardware outputs. No other inputs or outputs can be forced within the new Ladder Editor. If you exit the new Ladder Editor with a force enabled, that force will stay in effect until you remove it. You can force I/O using the Tag Database or the new Ladder Editor.

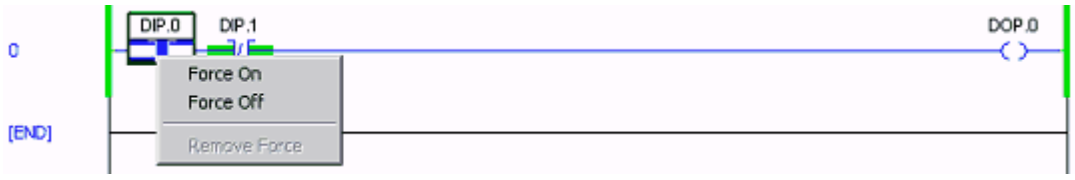
Force I/O in the Tag Database

Select Enable or Disable in the Force Action field, and edit the Force Value as needed.



Force I/O in the new Ladder Editor

- For discrete I/O, right-click the input or output element you wish to force, choose the appropriate force option, and edit the Force Value.
- For analog I/O, right-click the binding, choose the the appropriate force option, and edit the Force Value.



- When you force a tag, the background turns yellow.

Notes:

Register EDS Files and Add Devices Offline/Online

What This Appendix Contains

Read this appendix to learn about EDS files and how to unregister and register them. Additionally, learn how to add devices to the network (online or offline) using RSNetWorx for DeviceNet software.

EDS Files

Before you access the DeviceLogix capabilities of a device, you must have an EDS file registered that supports DeviceLogix functionality. Use the EDS Wizard in RSNetWorx for DeviceNet software to register the EDS file. The EDS file that supports DeviceLogix functionality must be newer than an existing EDS file for the device, if there is one. For best results, you should unregister the existing EDS file before registering the new EDS file.

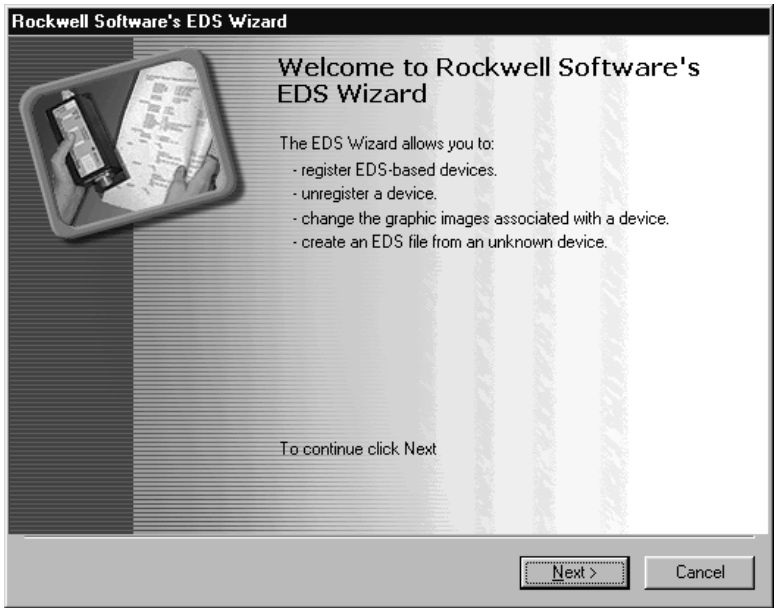
If your version of RSNetWorx for DeviceNet software is earlier than version 3.0, you must close RSNetWorx software and reopen it for the changes to take effect.

Unregister EDS files

To unregister an existing EDS file:

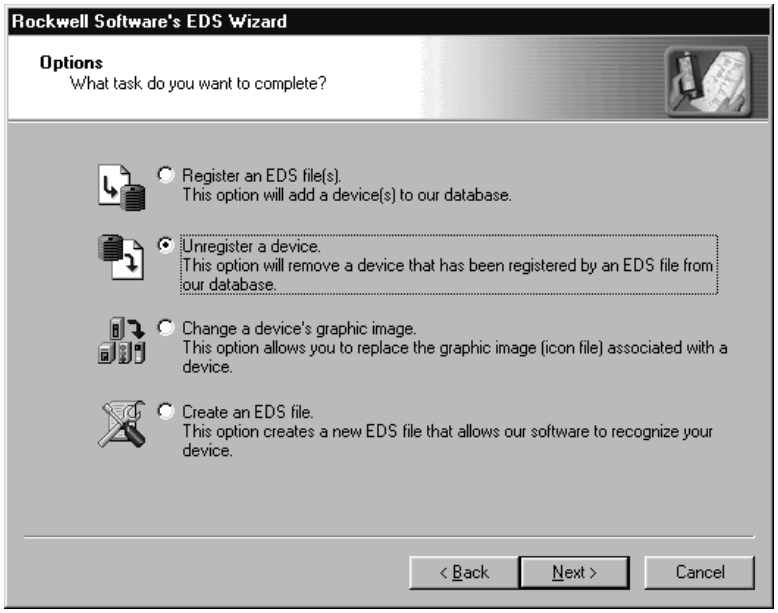
1. Open RSNetWorx for DeviceNet software.
2. Select **Tools > EDS Wizard**.

You see the welcome window for the EDS Wizard.

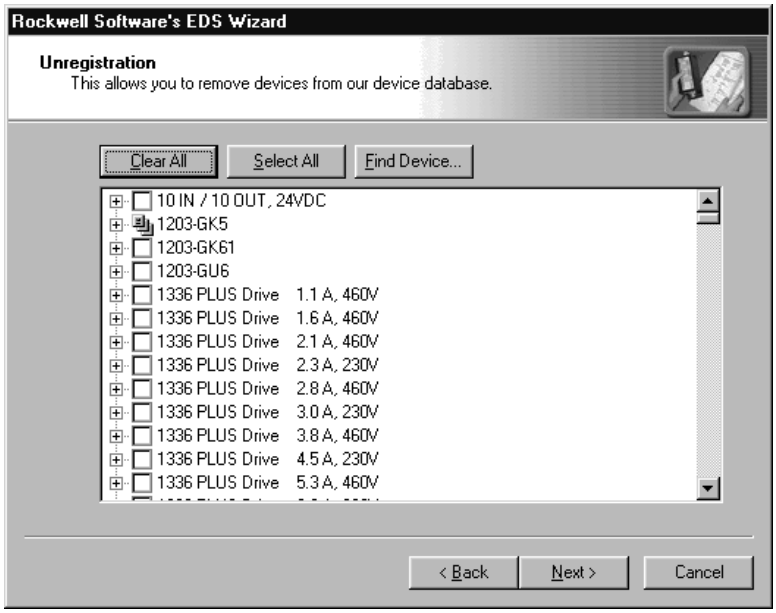


3. Click **Next** to start.

4. Select **Unregister a device**.



5. Click **Next**.



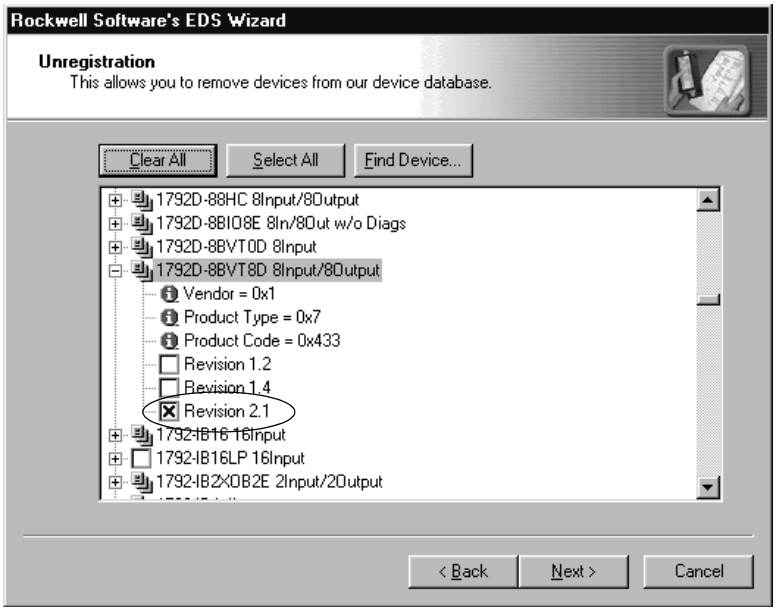
6. Click **Find Device**.

7. Type the name of the device you want to unregister in the **Find what device** box.

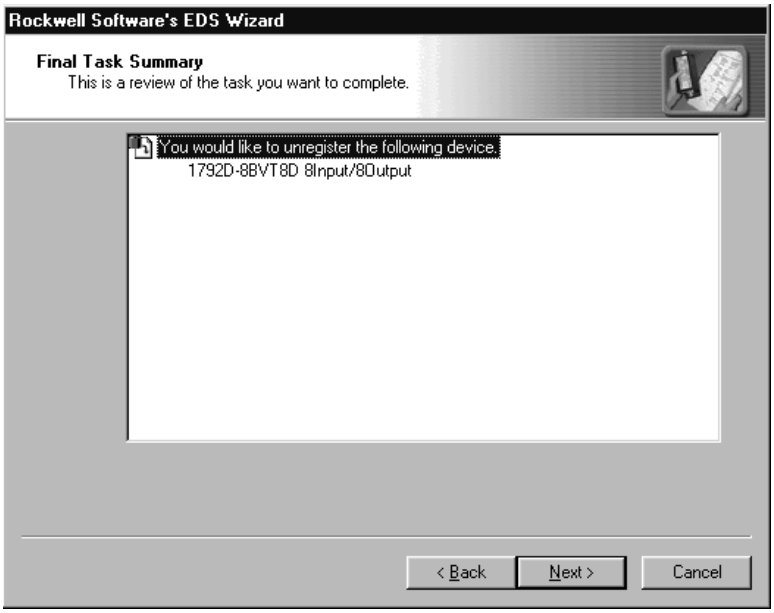
8. Click **Find Now**.

9. When the device appears in the EDS Wizard window, click **Cancel**.

10. Click the box in front of the device or revision so an **x** appears.



11. Click **Next**.



12. Click **Next** to unregister the EDS file.

You have successfully unregistered the existing EDS file.

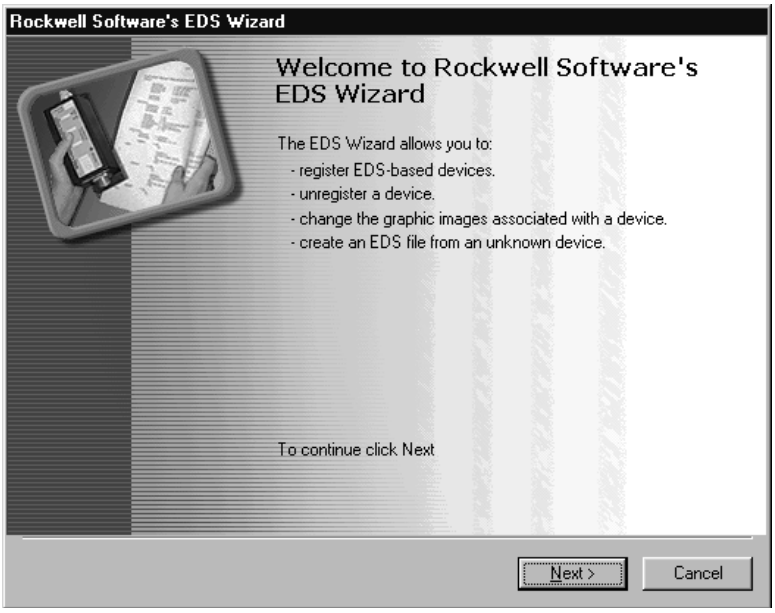
13. Click **Finish** to close the EDS Wizard.

Register EDS Files

To register EDS files:

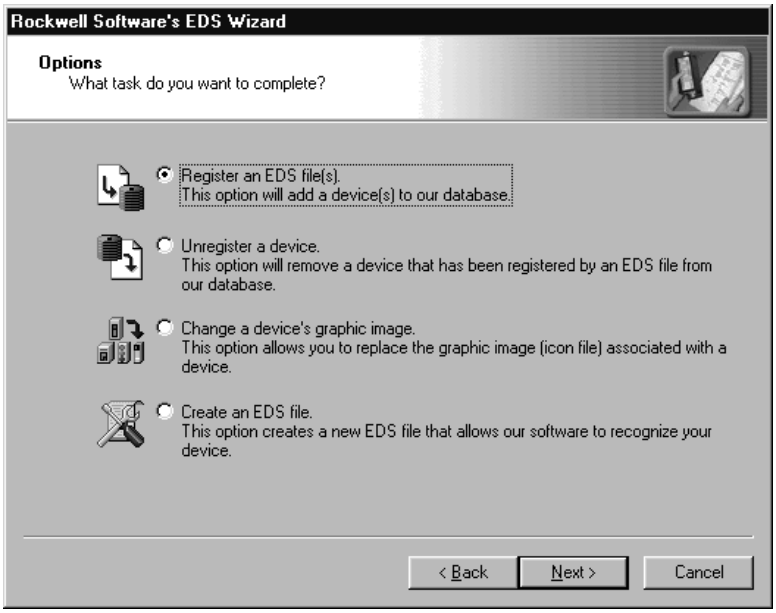
1. Open RSNetWorx for DeviceNet software, if it is not already open.
2. Select **Tools > EDS Wizard**.

   You see the welcome window for the EDS Wizard.



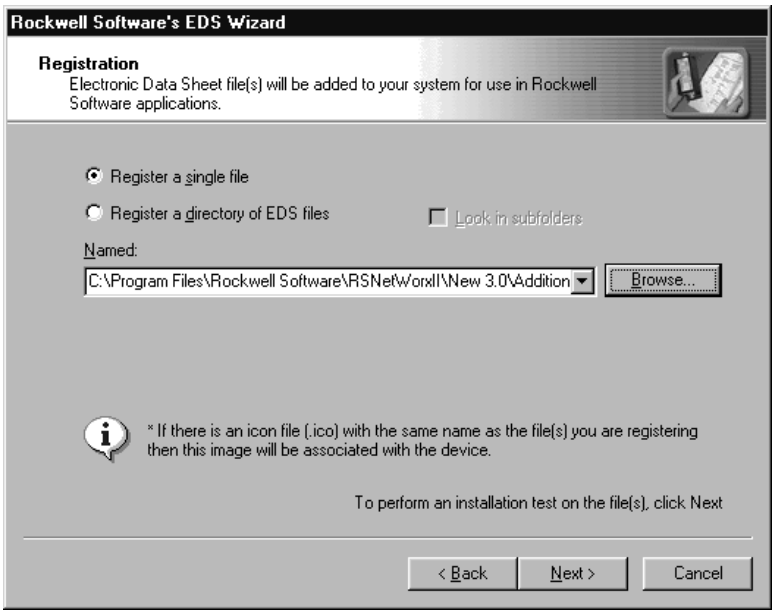
3. Click **Next** to start.

4. Verify that **Register an EDS file(s)** is selected.



5. Click **Next**.

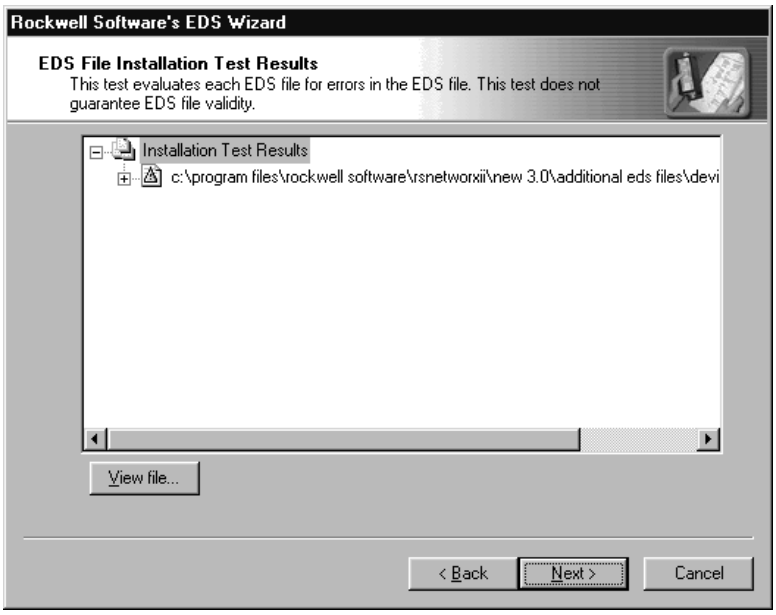
6. Choose to register a single file or a directory of files.



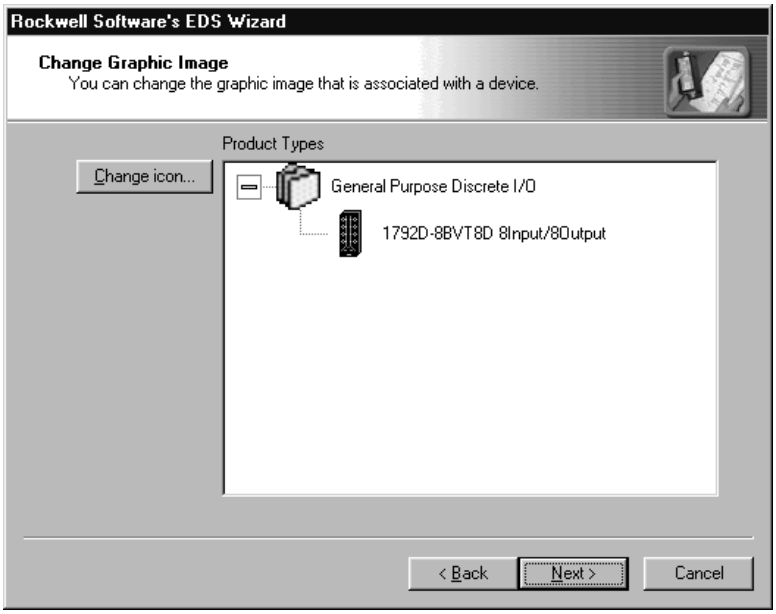
7. Click **Browse** if you need to find the correct file.

8. Click **Next**.

9. Click **Next** to evaluate the EDS files for errors.

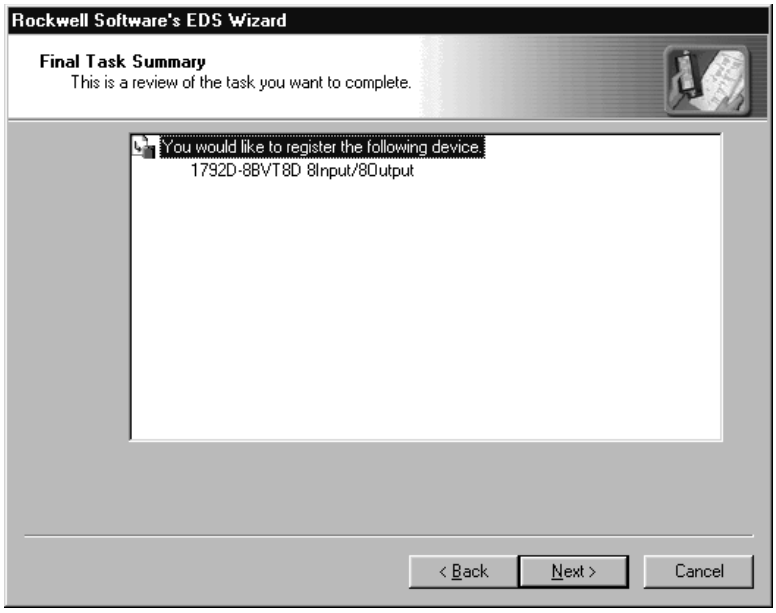


10. (optional) Click **Change icon** to choose a different icon.



11. Click **Next**.

12. Verify that you are registering EDS files.



13. Click **Next**.

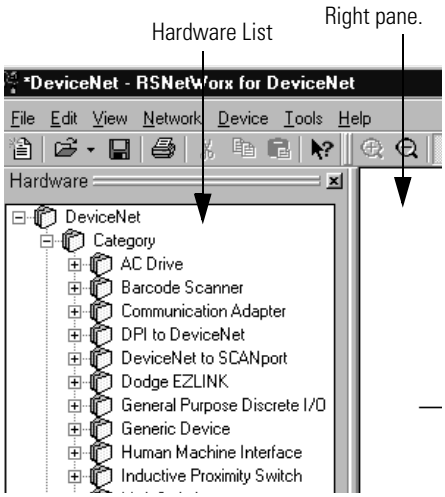
You have successfully registered the EDS file(s).

14. Click **Finish**.

15. If your version of RSNetWorx for DeviceNet software is lower than version 3.0, you must close RSNetWorx software and reopen it for the changes to take effect. If you continue to have problems registering EDS files, open RSNetWorx for DeviceNet software and follow this path for additional help: **Help > Release Notes > Known anomalies > Devices unrecognized after registering EDS files with the EDS Wizard**.

Add Devices offline

Once the EDS files are registered, you can add and configure devices in RSNetWorx for DeviceNet software. If you are not connected to a DeviceNet network, you can add devices offline. Follow the steps below to add devices offline.



1. Start RSNetWorx for DeviceNet software, if it is not already open.
2. Once RSNetWorx for DeviceNet software is open, search the hardware list (see figure on the left) for the desired category.
3. Click the **+** sign in front of the desired category to expand the list.
4. Search the list of products for the device you want to add.
5. Click once on the device you want to add and hold down the mouse key.
6. Drag and drop the device onto the graph (right pane - see figure on the left).
7. Add other devices as needed by following steps 3 through 6 above.

Add Devices online

If you are connected to a DeviceNet network, you can add devices online in RSNetWorx for DeviceNet software. Before you attempt to go online, make sure all the devices on the network:

- have a unique DeviceNet address
- are powered
- are at the correct baud rate
  NOTE: Autobaud devices only check for baud rate at power-up.

Once you have accomplished the above tasks, access RSNetWorx for DeviceNet software and select **Network > Online**.

RSNetWorx for DeviceNet software scans the DeviceNet network and adds all devices it finds. It also finds the appropriate EDS file definition for the revision of the device, if registered correctly.

You are now online.

If the device is DeviceLogix compliant and the EDS file has been properly imported into RSNetWorx for DeviceNet software, the device is ready to configure.

If a problem exists, a symbol appears above the device icon as displayed in RSNetWorx for DeviceNet software. See the appropriate documentation for RSNetWorx for DeviceNet software for details on how to resolve discrepancies that occur.

Notes:

RSNetWorx for DeviceNet Software and the DeviceLogix Editors

What This Appendix Contains

In this appendix you will access the RSNetWorx for DeviceNet software device properties dialog box to set up general properties, determine parameters and launch the DeviceLogix Editor tools.

Access Device Properties

Access device properties and DeviceLogix features from the device properties dialog box in RSNetWorx for DeviceNet software. The device properties dialog box is the same for both DeviceLogix devices and non-DeviceLogix devices, except when you are working with a DeviceLogix-enabled device, you see an additional tab called DeviceLogix in the properties dialog box.

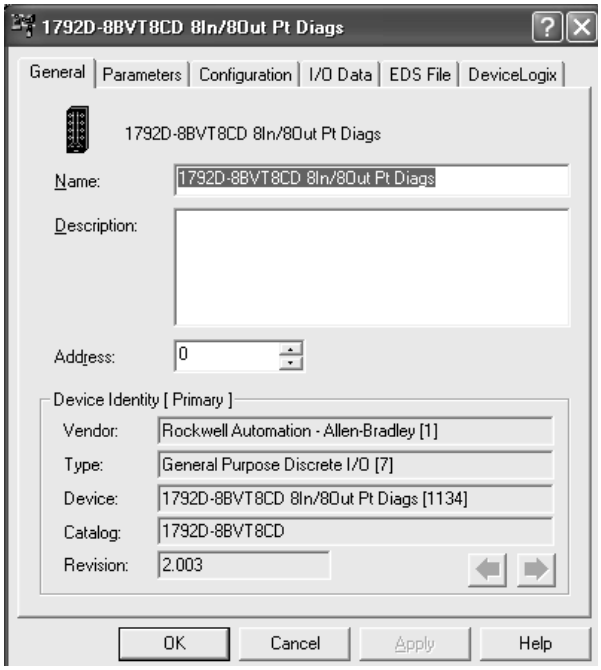
To access the device properties dialog box:

- Double-click the DeviceLogix-enabled device you added on the RSNetWorx for DeviceNet software graph (right pane).

The following sections describe the tasks you perform in the device properties dialog box that affect DeviceLogix features. For a complete discussion of the general features of the RSNetWorx for DeviceNet software device properties dialog box, see the RSNetWorx for DeviceNet software documentation.

Check General Information

The device properties dialog box opens to the General properties dialog. If you are in another dialog in the device properties dialog box, click on the General tab to return to the General dialog.



TIP The DeviceLogix tab is available on the General properties window when a DeviceLogix EDS file is registered for the current device.

The General window is common to both DeviceLogix-enabled devices and non-DeviceLogix-enabled devices. For DeviceLogix devices, you should make special note of the Address and Revision fields.

Set the Node Address

The address in this window must match the physical address on the module. If you added your devices off line, you most likely need to adjust the addresses of the devices. If you need to adjust the address:

1. Click once in the Address box.
2. Change the number of the address so that it matches the physical address on the module.
3. Click **OK**.

If you added the devices on line by letting RSNetWorx for DeviceNet software scan the network for devices, then the addresses in the General window should already match the physical addresses of the devices.

Check Revision Field

The revision field shows the current revision of the firmware in the I/O device. The revision field can help you identify if you are using the correct EDS file for DeviceLogix-enabled devices. If you are not using the correct EDS file for DeviceLogix-enabled devices, DeviceLogix features will not be enabled.

Using Window Buttons

The buttons at the bottom of the General window are common to all of the windows in the device properties dialog box. These buttons are:

Common Buttons

Function	Description
OK	Saves changes and closes the device properties dialog box.
Cancel	Discards changes made after the last time the Apply or OK button was used.
Apply	Saves changes without closing the device properties dialog box.
Help	Displays information that assists you with the dialog.

IMPORTANT Even though the OK and Apply buttons save changes made in the device properties dialog box, they do not actually save the information to the .dnt file or make changes to the configuration stored in the I/O device. You save information to the .dnt file by using the **File > Save** option in RSNetWorx for DeviceNet software.

If you make changes while on line, you are prompted to download or upload the device information whenever you try to go to a different window in the device properties dialog box. Click **OK** or **Apply** at the prompt. If you make changes while off line, you are not prompted to upload or download the device until you go on line and try to go to a different window in the device properties dialog box.

Once you have entered the desired information in the General window, click on other tabs on the device properties dialog box to access other device information.

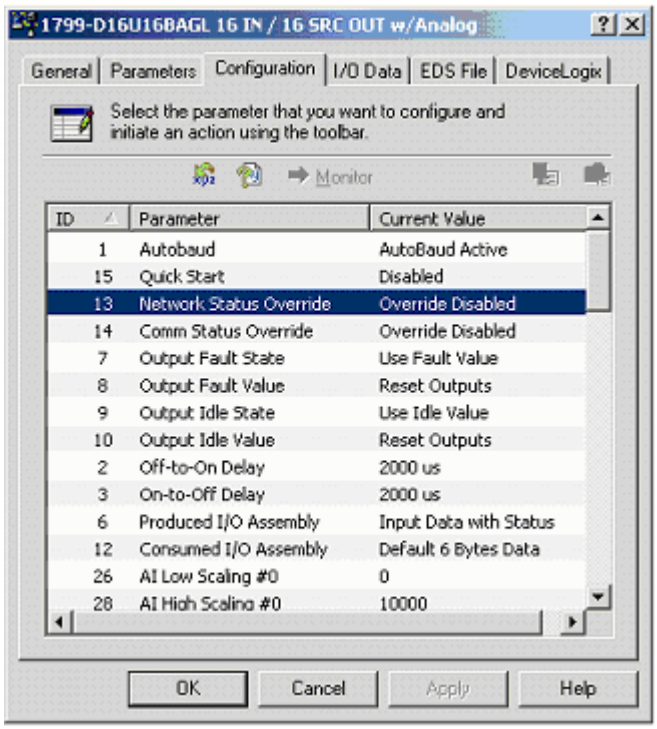
IMPORTANT If you are on line and you click on either the Device Parameters or the DeviceLogix tab, you may be prompted to upload or download the device. When you are on line, the dialog checks the configuration in the device and compares it to the current configuration. If the configurations are not the same, you must upload or download the device to make the configurations the same before you can make changes. If you need to make changes without uploading or downloading, exit the dialog box, go off line and re-enter the dialog box to make the desired changes and save to a .dnt file.

Enter Device Parameters

In this section, we describe the general parameters of the Device Parameters dialog box and help you become familiar with the parameters specific to DeviceLogix functionality. We do not describe parameters that are device specific. The only parameters discussed in detail are those that affect DeviceLogix features. For more information on other parameters presented in this dialog box, refer to documentation that comes with your specific device.

To access the Parameters dialog box, click on the Parameters tab.

IMPORTANT For some products, the DeviceLogix parameters may appear on the Configuration tab, as seen in the following illustration. You can more easily access parameters that you use for device configuration.



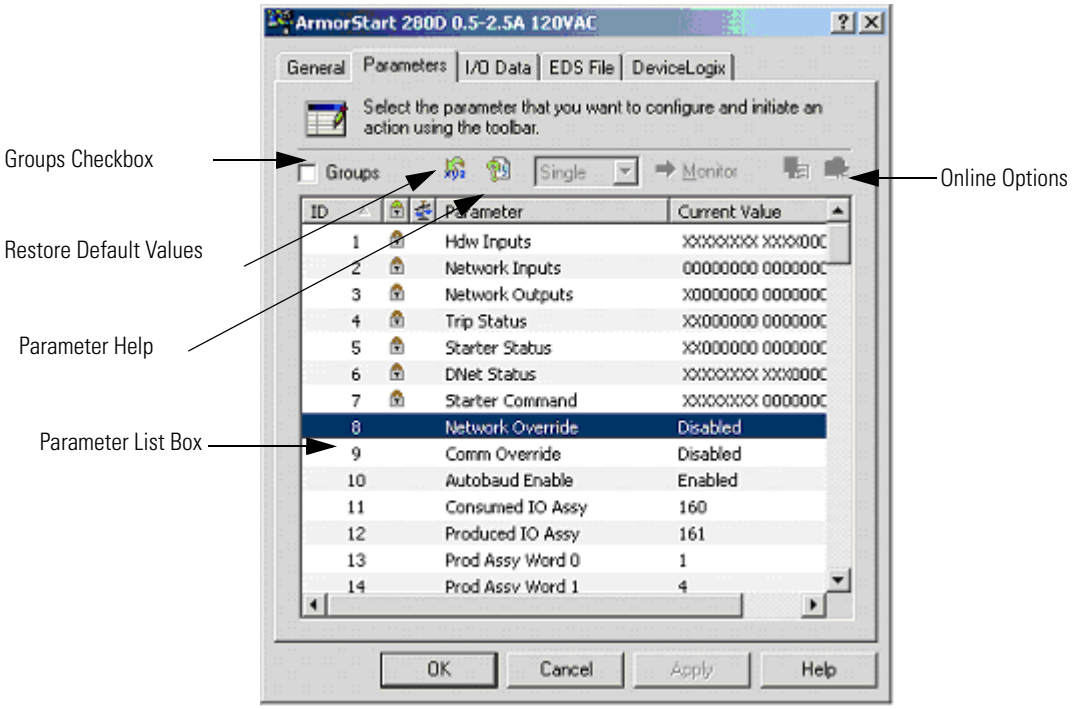
Device Parameter Controls

IMPORTANT Legacy DeviceLogix-enabled devices use a different style of the Parameter property page than the current DeviceLogix-enabled devices. Although the property pages may look different, they contain the same corresponding functions and options. The graphics and discussion in this section will only show the current Parameter property page.

There are a number of controls presented in the Parameters dialog box. Most of the controls are available both on line and off line, while others are available only on line. The on line only features include Groups, Upload From Device, Download To Device, and Start Monitor buttons as described in the Online Functions table.

Online Functions

Function	Description
Groups	Select which group of parameters to view.
Upload From Device	Reads the configuration data from the device.
Download To Device	Writes the current configuration to the device.
Start Monitor	Continually monitors either a single parameter in the parameter list or all the parameters in the parameter list. You tell the monitor routine whether you want to monitor a single parameter or all parameters using the Single or All buttons located to the left of the on-line buttons. Note that when you click the Start Monitor button, the button name changes to Stop Monitor. You can toggle the monitor mode on or off by clicking this button.



Features available for both on line and off line use include controls that let you:

- Select which parameters to view
- Restore parameter defaults
- Access help for specific parameters, and
- Access a list of parameters and their current values

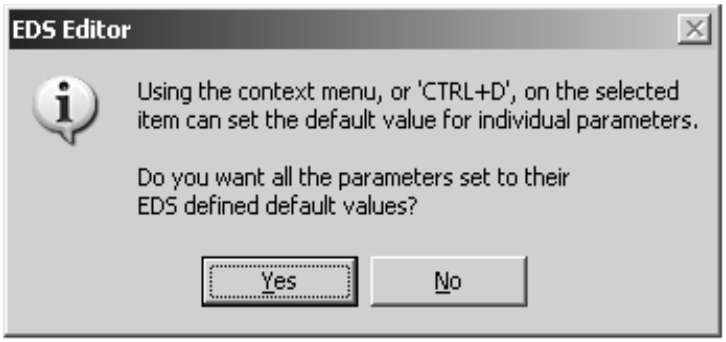
Select Group Parameters

For Current DeviceLogix-enabled devices, the Groups checkbox lets you select which parameters you wish to view in the parameter list. The default is all parameters. Once you check the Groups check box, the parameters will be automatically grouped.

Restore Default Values

The Restore Default Values button resets all the parameters to their default values as defined in the EDS file. It is possible that the default values are different from the values last saved in the .dnt file or from the values stored in the device. The default values are in the device when it is shipped from the factory. To restore default values:

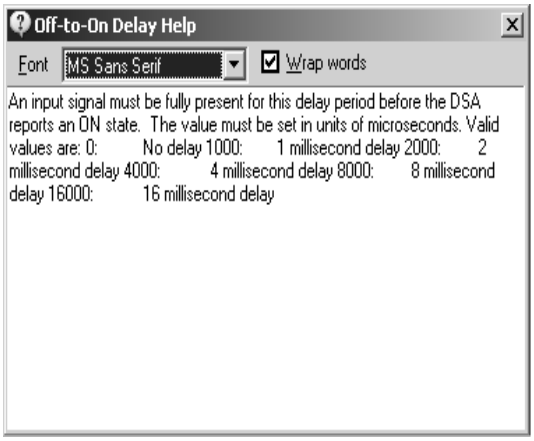
1. Click the Restore Default Values button.
2. Click **Yes** to restore default values.



Access Parameter Information

Use the Parameter Help button to display a short description of a parameter.

1. Click the parameter in the list box at the bottom of the window.
2. Click the Parameter Help button to see the short description.



Understand Parameters

The bottom portion of the Parameters dialog box contains the parameter list box. All the parameters of the I/O device that can be modified are listed here. Most of these parameters are specific to the I/O device and are not discussed in this publication. However, the following four parameters are specific to DeviceLogix functionality:

- Comm Status Override
- Network Status Override
- Consumed I/O Assembly
- Produced I/O Assembly

Before we help you determine values for DeviceLogix parameters, we will discuss network I/O.

Understand Network I/O

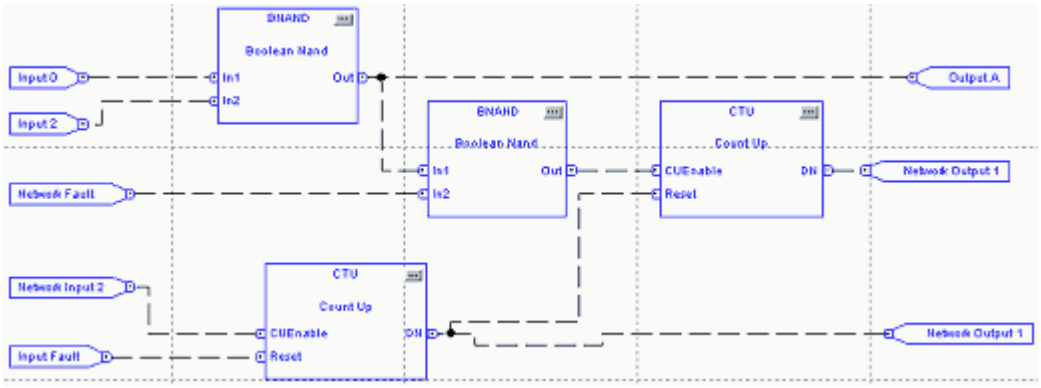
Network I/O is data that is consumed or produced by a DeviceLogix-enabled device that is not directly associated with any hardware on the device. The produced and consumed cases are discussed separately.

Understand Produced Network I/O (also known as Network Outputs)

Under normal conditions, an I/O device produces the state of its inputs and the status of any fault information on the device. However with local logic running on a device, a master controller sometimes needs to know the results of some intermediate state or value of logic. Using a special I/O assembly containing network outputs, it is possible for the device to report the state of any portion of the logic.

Each network output has a space reserved for it in the module's produced I/O assembly. When you connect this network output to something in the logic, those results are reported in the produced data.

In the following example, you can see when hardware inputs 0 and 1 are on at the same time. This is reported in Network Output 0. Network Output 1 reports when Faults have occurred. Both of these pieces of information are reported in the produced I/O data shown below the example.



Produced I/O Data

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	In 7	In 6	In 5	In 4	In 3	In 2	In 1	In 0
Byte 1		OPWR	Logic Ena					
Byte 2	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0
Byte 3	Network Output 7	Network Output 6	Network Output 5	Network Output 4	Network Output 3	Network Output 2	Network Output 1	Network Output 0

All data in the assembly including Network Outputs are capable of causing a Change of State production.
OPWR = Output Power; Logic Enabled = DeviceLogix Enabled

Understand Consumed Network I/O (also known as Network Inputs)

Under normal conditions, an I/O device consumes data to apply to its hardware outputs. DeviceLogix-enabled devices may consume additional information to use in local logic.

An assembly is a collection of parameters from one or more objects. The consumed I/O assembly for a device with digital outputs contains the value parameters of the Discrete Output Point Objects representing the hardware outputs present on the device. In other words, the assembly is sent with the data to control the hardware outputs on the device. Such an assembly for a module with 8 digital outputs is shown in the Consumed I/O Assembly table.

Consumed I/O Assembly

	Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Byte 0	Output Value 7	Output Value 6	Output Value 5	Output Value 4	Output Value 3	Output Value 2	Output Value 1	Output Value 0

When local logic is being used, DeviceLogix-enabled devices treat all consumed I/O data as network inputs. That is to say that the device may consume any amount of data, regardless of hardware, and treat it as generic input from the network. By default, the first *N* Network Inputs provide the data for the *N* hardware outputs found on the device. However, if you bind the hardware output to some other entity in the logic, the corresponding network input does not provide the control for that output.

Each Network Input has a space reserved for it in the consumed I/O assembly. The number of network inputs that a device consumes is not directly associated with the number of hardware resources present on the device. Therefore extra data can be sent to the module for use in the local logic. One such assembly is shown below for the same 8-output device considered above.

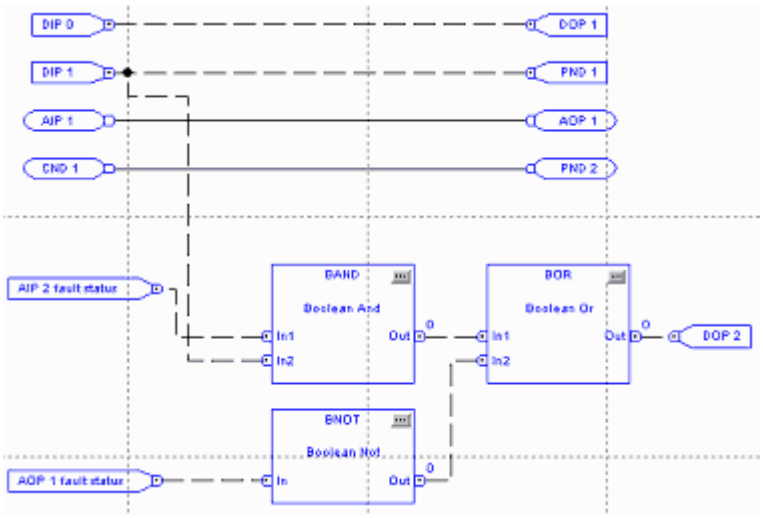
Reserved Consumed I/O Assembly Bit

	Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Byte 0	Network Input 7	Network Input 6	Network Input 5	Network Input 4	Network Input 3	Network Input 2	Network Input 1	Network Input 0
Byte 1	Network Input 15	Network Input 14	Network Input 13	Network Input 12	Network Input 11	Network Input 10	Network Input 9	Network Input 8

Note that even though the device only contains 8 outputs, it is consuming 16 bits of information. This gives you the option to map additional information into the data that is sent to this device by the master controller.

The following list contains several important details about network inputs.

- If an output is not used in the local logic, there is an assumed connection to a corresponding bit in the consumed data. In the illustration below, this connection is shown explicitly for outputs 1, 3, 4, 5, 6, 7. Note this connection does not need to be made graphically with the configuration tool. The DeviceLogix-enabled module assumes this connection until it is broken by making a connection to some other piece of information. If no connection is made to this output, it continues to take its control from the master controller via its network input.



- Network inputs can be used as input to more than one entity.
- When an output is bound to logic, its corresponding network input can be used somewhere else or not used at all.
- As mentioned earlier, the number of network inputs is not directly associated with the number of hardware outputs present on the device.
- For devices that support analog functions, the assembly may also include analog data types (DINT, REAL, etc.).
- Network I/O extends the capabilities of I/O modules by allowing them to produce and consume the data needed in their application. Network I/O is critical to the effective use of DeviceLogix-enabled devices in a networked control system because of its ability to exchange information between the local control and external events being monitored by the master controller.

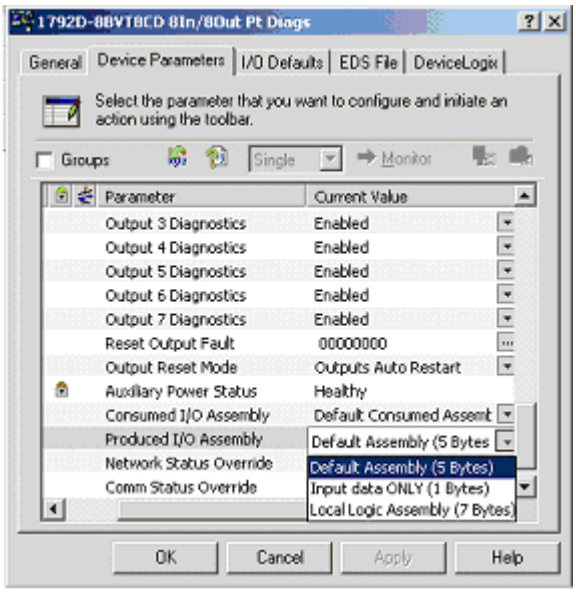
TIP In legacy products, network I/O is referred to as CNB (Consumed Network Bits) and PNB (Produced Network Bits) respectively. In the latest set of products, network I/O is called Network Input and Network Output. The data types could be Boolean, DINT, REAL, etc.

Determine Parameters

To configure your system, there is a possibility that you have to determine produced and consumed I/O assemblies. Refer to product specific publications to help you determine which parameters are applicable. The following sections help you determine values for common DeviceLogix parameters.

Determine Produced I/O Assembly

The produced I/O assembly parameter lets you select how much data gets produced across the network. In order to have DeviceLogix specific data produced by the device, select Local Logic Assembly. You must make sure that the settings in the scanner's scanlist match your selection here. The following example illustrates this concept for an 8in/8out MaXum block.



Input 7	Input 6	Input 5	Input 4	Input 3	Input 2	Input 1	Input 0
OW-D	OW-C	OW-B	OW-A	ISC-D	ISC-C	ISC-B	ISC-A
OFLT 7	OFLT 6	OFLT 5	OFLT 4	OFLT 3	OFLT 2	OFLT 1	OFLT 0
	OPWR	Logic Ena					
OUT 7	OUT 6	OUT 5	OUT 4	OUT 3	OUT 2	OUT 1	OUT 0
PNB 7	PNB 6	PNB 5	PNB 4	PNB 3	PNB 2	PNB 1	PNB 0

Standard MaXum Status and Diagnostic Bits

OW = Off-Wire; ISC = Input Short Circuit; OFLT = Output Fault; OPWR = Output Power

DeviceLogix Bits;

Logic Ena = DeviceLogix is Enabled. This can be used in the PLC to recognize that the remote device is running a control program.

Out X = Status of Local Output Bit if it is under control of DeviceLogix

PNB X = Network Output Bits

(Where X = the number of the Output Bit or Network Output Bit.)

To change a produced I/O assembly:

1. Click the current value of the Produced I/O Assembly.

   The field turns into a drop-down list box.

2. Click the arrow to display the values.

3. Click the desired value.

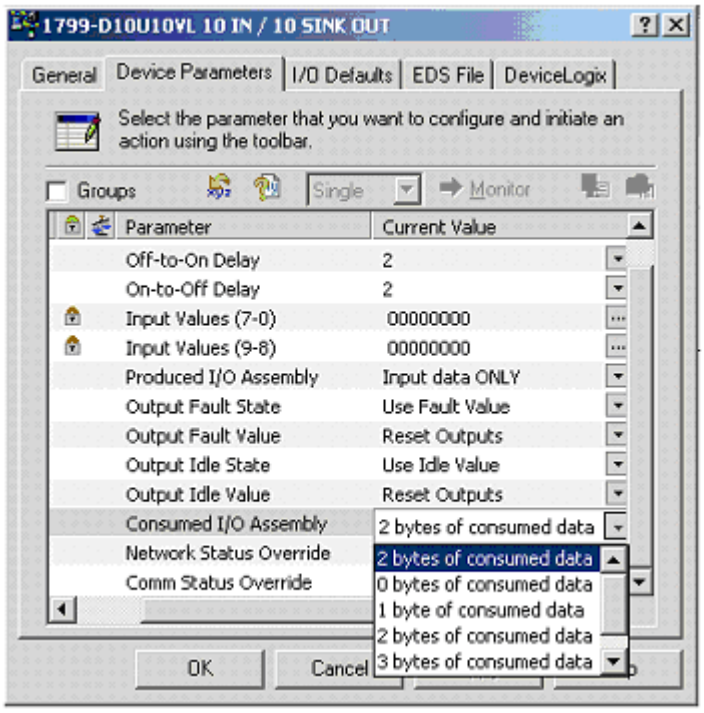
4. Click **Apply** to save the new value without closing the window. Or click **OK** to save the new value and to close the window.

IMPORTANT Apply and download will be unsuccessful unless local logic is disabled and the device is removed from the scanlist as discussed in the Download To a Device section.

For the layout of the Produced I/O Assembly for your chosen device, refer to the Technical Data publication for that device.

Determine Consumed I/O Assembly

Devices, when enabled with DeviceLogix functionality, can receive additional data across DeviceNet. This data is called network inputs. The consumed I/O parameter lets you select how much data you want the device to consume from a master. After setting this parameter, the same number of bytes must be selected in the scanner's scan list. For example, if you choose 3 bytes of data to be consumed, you must indicate that 3 bytes of data are to be transmitted in the scanner's scanlist. If the number of consumed bytes and the number of transmitted bytes do not match, the connection cannot be made.



Note that this parameter is not in all DeviceLogix-enabled devices.

To change a consumed I/O assembly:

1. Click the current value of the Consumed I/O Assembly.

   The field turns into a drop-down list box.

2. Click the arrow to display the values.

3. Click the desired value.

4. Click **Apply** to save the new value without closing the window. Or, click **OK** to save the new value and to close the window.

To see the layout of the consumed I/O assembly for your chosen device, refer to the Technical Data publication for that device.

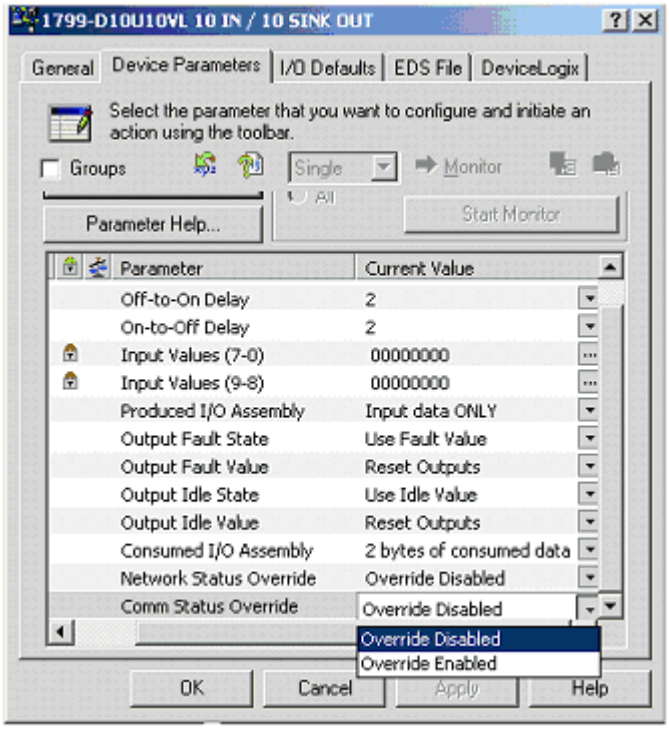
Determine Comm Status Override

The Comm Status Override parameter controls whether local logic should control outputs when no active I/O connection exists with the device. You determine if the communication status override is enabled or disabled.

The Comm Status Override parameter is used to override normal behavior during the following events.

Comm Status Override Overview

Event	Behavior with Comm Status Override Parameter Disabled	Behavior with Comm Status Override Parameter Enabled
Communications not established (module not on line). - OR - The module is on line but has no connections.	Output remains in the Available state until an I/O connection is established.	Local logic updates output values.
An I/O connection transitions to timed out state.	Output value is updated based on the output's Fault Action and Fault Value parameters.	Local logic continues to update output values.
An I/O connection is deleted.	Output enters the Available state until a new I/O connection is established.	Local logic continues to update output values.
An Idle is received.	Output value is updated based on the output's Idle Action and Idle Value parameters.	Local logic continues to update output values.



IMPORTANT The above information is relative only to bound outputs (i.e., outputs that are participating in the local logic).

To change the comm status override:

1. Click the current value of the Comm Status Override parameter.

   The field turns into a drop-down list box.

2. Click the arrow to display the values.

3. Click the desired value.

4. Click **Apply** to save the new value without closing the window. Or click **OK** to save the new value and to close the window.

Determine Network Status Override

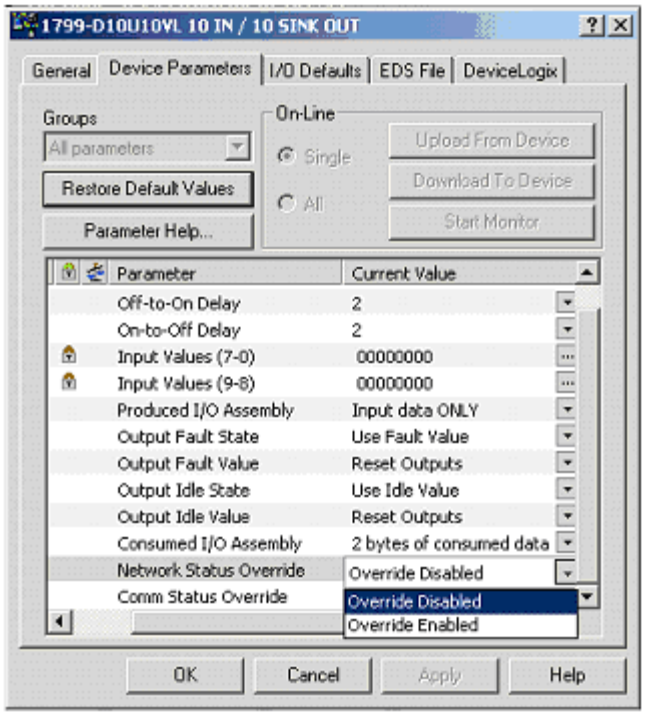
The network status override parameter controls whether local logic should control outputs when it detects a network error condition. You determine if the network status override is enabled or disabled.

When the override is disabled (default), the device turns outputs off under any of the events. If the override is enabled, the device ignores any network fault. If DeviceLogix functionality is enabled, the control of outputs is maintained even if there is a network fault.

The Network Status Override parameter is used to override normal behavior during the following events.

Network Status Override Overview

Exemplary Network Error Conditions for DeviceNet	Behavior with Network Status Override Parameter Disabled	Behavior with Network Status Override Parameter Enabled
Duplicate MAC ID Failure	Module is put into an inoperable state and all outputs remain off.	Local logic continues to update output values.
Entering the bus off state at power up	Module is put into an inoperable state and all outputs remain off.	Local logic continues to update output values.
Entering the bus off state during run time	Module is put into an inoperable state and all outputs assume a safe state.	Local logic continues to update output values.



To the change the network status override:

1. Click the current value of the Network Status Override parameter.

   The field turns into a drop-down list box.

2. Click the arrow to display the values.

3. Click the desired value.

4. Click **Apply** to save the new value without closing the window. Or click **OK** to save the new value and to close the window.

Download to a Device That Is Scanning

When you change the size of a connection, you must download to the device for the change to take effect. When downloading to a device that a master is currently scanning, you must:

1. Access the DeviceNet scanner's scanlist to remove the device from the scanlist.

2. Put the master in Idle mode. (This may mean turning a processor's keyswitch to Program.)

3. Access the device's parameters to change the size of the connection.

4. Download to the device.

5. Add the device to the scanlist and click **Apply**.

6. Click the Edit I/O Parameters button and verify that the Rx and Tx size corresponds to the new size.

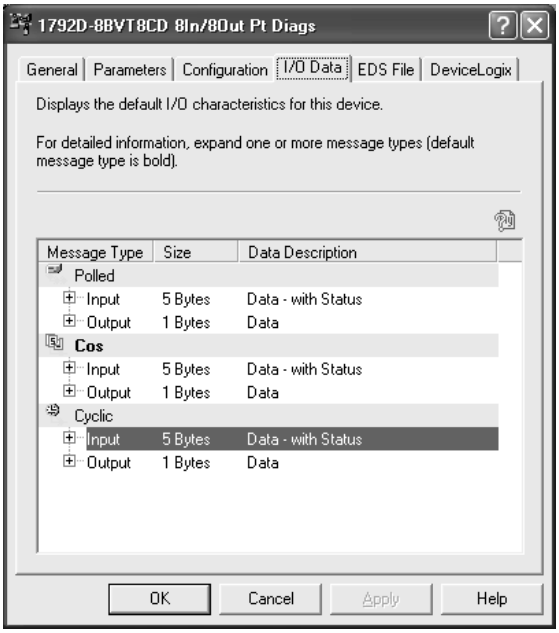
7. Put the master in Run mode for normal operation.

Access I/O Data Information

The I/O Data window provides information about default I/O characteristics for the device. The amount of data that the device reports is described here. Note however that the help only covers the data returned by the EDS file default settings. This window does not report data enabled by the DeviceLogix-specific settings.

To access the I/O Data window, click on the I/O Data tab in the device properties dialog box.

IMPORTANT Legacy DeviceLogix-enabled devices use a different style of the I/O Data (called I/O Defaults) property page than the Current DeviceLogix-enabled devices. Although the property pages may look different, they contain the same corresponding functions and options. The graphics and discussion in this section will only show the current I/O Data property page.



To access I/O data information:

1. For the default message type (in bold), click Input to see all of the input I/O default characteristics for the connection. Click Output to see all of the output I/O default characteristics for the connection.

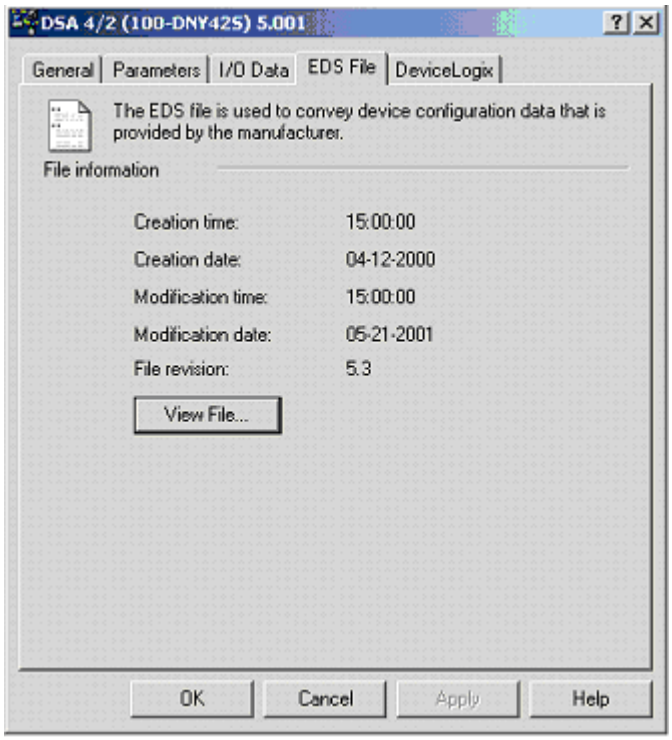
   Only those message types that are supported for the device connection are displayed.

2. Repeat for the remaining message types (if necessary).

Access EDS Information

The EDS File window lists information about the selected device. Use this information to determine if the EDS file is appropriate for the device. Compare information in this window against EDS files available on the web to determine if you have the most current version of the file.

To access the EDS File window, click on the EDS File tab in the device properties dialog box.



Click **View File** to view the contents of the selected EDS file. The information stored in the EDS file is very detailed and is beyond the scope of this document. However, if you experience difficulty enabling DeviceLogix features, check for the following lines in the [Device] section of the EDS file:

- "1_RSNetWorx_DNetTool = "LeoApp.LeoDevice";

  Indicates that LCT should be used on this device. If only this keyword appears, then the new Ladder Editor will not be available, the old Ladder Editor will be available, and the Parameter tab and I/O Default tab will reflect the old interface.

- "1_RSNetWorx_DNetToolEx = "1.00.00", "LeoApp.LeoDevice", "7.00","RSI.DeviceNet.EDSApplet.1";

  Indicates LCT should be used on this device. The Parameter tab and I/O Default tab will reflect new interface, but the DeviceLogix editor will not available.

- "1_DNetEDSAppletPlugIn = "7.00.00", "DeviceLogixPlugIn.EDSAppletPlugIn.1";

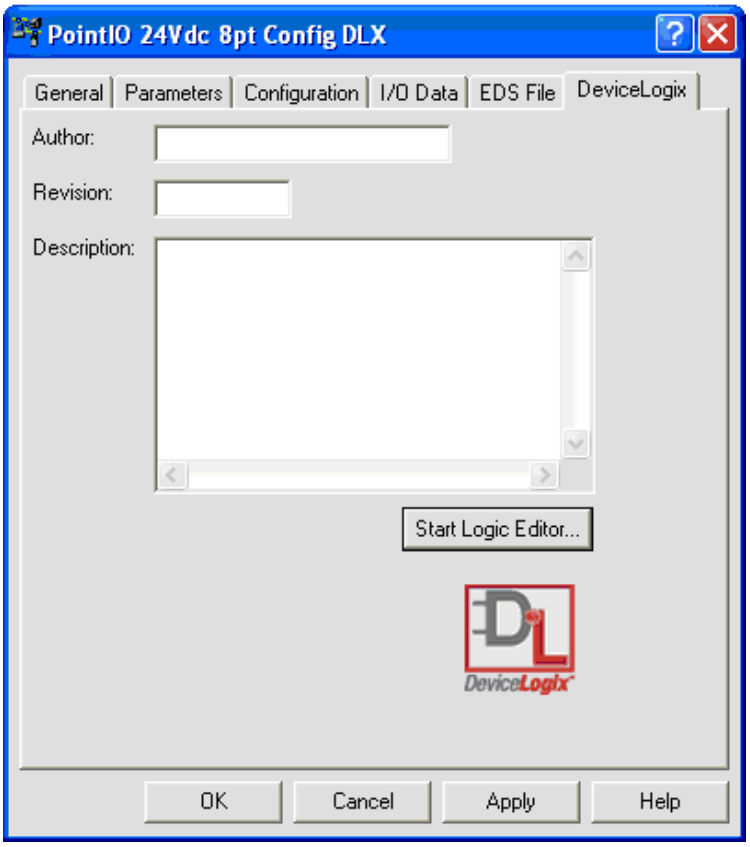
  Indicates LCT should be used on this device. The Parameter tab and I/O Default tab will reflect new interface, and the new Ladder Editor will be available.

If these three lines are not part of the [Device] section of the EDS file for the device, the device will not be DeviceLogix-enabled.

DeviceLogix EDS files will not work in a version prior to 3.0 of RSNetWorx for DeviceNet software. DeviceLogix EDS files do not ship with some devices or as part of the normal RSNetWorx software release. They are found in a separate folder on the CD that contains RSNetWorx for DeviceNet software. Therefore, to enable DeviceLogix features for a DeviceLogix-capable device, you must register the EDS file for your particular device using the EDS wizard.

Launch the DeviceLogix Editor

After you configure the properties for your DeviceLogix-enabled device, you can launch the DeviceLogix Editor. You see an additional tab in the device properties dialog box for all DeviceLogix-enabled devices. This tab is labeled DeviceLogix. This tab provides access to the start-up window for the DeviceLogix Editor. You have the option to fill in your name, a revision number, and a description of your configuration (all optional fields). The Last Saved box updates every time you save within RSNetWorx for DeviceNet software.



IMPORTANT If you are on line and you click on either the Parameters or the DeviceLogix tab, you may be prompted to upload or download the device. When you are on line, the dialog checks the configuration in the device and compares it to the current configuration. If the configurations are not the same, you must upload from or download to the device to make the configurations the same before you can make changes. If you need to make changes without uploading or downloading, you can exit the dialog box, go off line and re-enter the dialog box to make the desired changes.

To start the DeviceLogix Editor for a DeviceLogix-enabled device, click **Start Logic Editor**. On the DeviceLogix Editor Style Selection dialog, you are prompted to select the editor type that you want to launch. After selecting an editor type, click **OK**.:

If the current device does not support one of the editor types, no selection will be presented.

IMPORTANT If you select an editor type for a particular device and that type is committed to the .dnt file (clicking **OK** or **Apply**), that editor style is registered. Therefore, you cannot switch to another editor style in that same .dnt file (the next time you launch the DeviceLogix Editor Style Selection dialog, the other editor style is grayed out). If you want to change the editor type (and a device supports both editor types), you must create a new project file, delete this device and add it again, and then select the other editor type.

For more information on the Function Block Editor, refer to Chapters 2 and 3. For more information on the old Ladder Editor, refer to Chapters 4 and 5. For more information on the new Ladder Editor, refer to Chapters 6 and 7.

Notes:

M

N

O

V

W

Notes:

Rockwell Automation Support

Rockwell Automation provides technical information on the Web to assist you in using its products. At http://www.rockwellautomation.com/support/, you can find technical manuals, a knowledge base of FAQs, technical and application notes, sample code and links to software service packs, and a MySupport feature that you can customize to make the best use of these tools.

For an additional level of technical phone support for installation, configuration, and troubleshooting, we offer TechConnect support programs. For more information, contact your local distributor or Rockwell Automation representative, or visit http://www.rockwellautomation.com/support/.

Installation Assistance

If you experience an anomoly within the first 24 hours of installation, review the information that is contained in this manual. You can contact Customer Support for initial help in getting your product up and running.

United States or Canada	1.440.646.3434
Outside United States or Canada	Use the Worldwide Locator at http://www.rockwellautomation.com/support/americas/phone_en.html, or contact your local Rockwell Automation representative.

New Product Satisfaction Return

Rockwell Automation tests all of its products to ensure that they are fully operational when shipped from the manufacturing facility. However, if your product is not functioning and needs to be returned, follow these procedures.

United States	Contact your distributor. You must provide a Customer Support case number (call the phone number above to obtain one) to your distributor to complete the return process.
Outside United States	Please contact your local Rockwell Automation representative for the return procedure.

Documentation Feedback

Your comments will help us serve your documentation needs better. If you have any suggestions on how to improve this document, complete this form, publication RA-DU002, available at http://www.rockwellautomation.com/literature/.

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europe/Middle East/Africa: Rockwell Automation, Vorstlaan/Boulevard du Souverain 36, 1170 Brussels, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Publication RA-UM003B-EN-P - February 2010
Supersedes Publication RA-UM003A-EN-P - October 2008

Allen-Bradley

DeviceLogix System

User Manual