LonPoint™ Application and Plug-in Guide

Version 2.0

ECHELON
Corporation

078-0166-02A

Echelon Corporation
4015 Miranda Avenue
Palo Alto, CA 94304, USA

Contents

1

LonPoint System Overview

This chapter introduces the LonPoint System, applications, plug-in, and utilities.

Introduction to the LonPoint System

The LonPoint System is a family of LONMARK® products used to integrate new and legacy devices, as well as other LONMARK devices, to create interoperable control systems. The LonPoint system product family includes the following:

- *LonPoint Interface, Scheduler, Data Logger, and Router Modules*. LONMARK devices that provide I/O processing, application resources, scheduling, sequencing, data logging, and routing for a LonPoint system. The LonPoint interface, scheduler, and data logger modules are certified to meet the LONMARK Interoperability Guidelines.
- *LonMaker for Windows Integration Tool*. An LNS installation tool with a Visio™ user interface that supports LonPoint devices, other LONMARK devices, and other LONWORKS devices. You can use the LonMaker tool to design, configure, commission, and maintain a distributed control network. The LonPoint Plug-in is included with the LonMaker tool to simplify LonPoint device configuration.
- *LonPoint Plug-In*. An LNS application that provides an easy-to-use interface for configuring LonPoint devices. The plug-in can be called from any LNS tool that follows the LNS plug-in standard, including the LonMaker tool.
- *LNS Server*. The service provider for the LNS network operating system. Provides a central database that can be used by multiple LonMaker tools, LonPoint Plug-ins, and other LNS applications simultaneously. The LNS Server may run on the same PC with the LonMaker tool and LonPoint Plug-in, or may run on a different PC. A remote LNS Server may be accessed over a LonWorks network, a local area network, or the Internet.
- *LNS Network Interface Hardware*. Hardware that allows you to connect a PC running the LNS Server, the LonMaker tool, and the LonPoint Plug-in to a LONWORKS network. This hardware is not necessary for network design, but must be installed to commission, test, or manage devices. The LNS network interfaces include the PCLTA-10 ISA card, the PCC-10 PC Card (PCMCIA compatible), the SLTA-10 Serial LonTalk Adapter, and the PCNSI ISA card.

Each LonPoint device comes pre-loaded with application software that implements a number of functional blocks. These functional blocks are listed in Table 1.1. They are installed and interconnected using the LonMaker tool as described in the *LonMaker for Windows User's Guide*. The functional blocks are then configured as described in this user's guide.

Table 1.1 LonPoint Functional Blocks

Chapter	***Name***	***Description***	***Examples***
2	Digital Input (DI)	Senses digital inputs	Contact closure, push button input, logic input
3	Digital Input/Counter (DC)	Senses, counts, and times digital inputs	Turnstyle counting, frequency measurement, flow meter interface.
4	Digital Output (DO)	Drives digital outputs	Control relays, logic outputs
5	Digital Encoder (DE)	Performs logic functions on up to four digital inputs	Relay logic, interlocks, Boolean logic, device enables

6	Analog Input (AI)	Senses current, voltage, or resistance inputs	Measure temperature, pressure, humidity, velocity, level
7	Analog Output (AO)	Drives current, voltage, or resistance analog outputs	Variable speed drive control; valve control; damper control
8	Analog Function Block (AFB)	Performs functions on two analog values and a digital value such as add, subtract, multiply, divide, greater than, less than, enthalpy, and comparison	Deadband thermostat, heating/cooling optimization
9	PID Controller (PID)	Performs closed-loop control	Pressure, temperature, or position loops
10/11	Real Time Clock (RTC)	Maintains date, day of week, and time of day	Non-volatile system time source
10/11	Event Scheduler (ES)	Schedules system events	Scheduler with occupied and unoccupied periods, daily, weekly, holiday schedules
10/11	State Machine (SM)	Controls system state	Sequence of operations controller
12	Type Translator	Converts network variable data types	Interface between incompatible devices
13	Data Logger	Logs periodic data based on exceptional data values or large changes in data value.	Logging periods of extreme temperature/rapid fluctuations.
14	Node Object	Management commands for a device	Putting devices into Override. Getting device status.

The LonPoint functional blocks are loaded into the LonPoint devices as listed below.

Table 1.2 Assignment of LonPoint Functional Blocks to LonPoint Devices

LonPoint Device	Functional Block	Quantity per Device
DI-10	Node Object	1
	Digital Input	4
	Digital Encoder	2
	Analog Function Block	4
	Type Translator	6
DO-10	Node Object	1
	Digital Output	4
	Digital Encoder	2
	Analog Function Block	2
	Type Translator	6
DIO-10	Node Object	1
	Digital Input/Counter	2
	Digital Output	2
	Digital Encoder	2
	Analog Function Block	2

	Type Translator	6
AI-10	Node Object	1
	Analog Input	2
	Digital Encoder	2
	Analog Function Block	4
	Type Translator	4
AO-10	Node Object	1
	Analog Output	2
	PID Controller	2
	Digital Encoder	1
	Analog Function Block	2
	Type Translator	2
SCH-10 (with SCH-10 application)	Node Object	1
	Real Time Clock	1
	Event Generator	1
	State Machine	1
SCH-10 (with DL-10 application)	Node Object	1
	Data Logger	1
	Real Time Clock	1

Each functional block has a number of network variables, through which data is passed to or from the functional block, and configuration properties, which are used to configure the functional block and define its behavior. The functional blocks are implemented as LONMARK objects on the LonPoint devices, and are configured from the LonMaker tool using the LonPoint Plug-in and utilities.

Note:

The SCH-10 device ships with the SCH-10 application pre-loaded. The DL-10 application can be downloaded into the SCH-10 device with a network integration tool such as the LonMaker tool for Windows.

Getting Started

Install the LonPoint Plug-in software as described in the *LonMaker for Windows User's Guide*. Create a network design that includes LonPoint devices and functional blocks.

To start the LonPoint Plug-in, right-click a LonPoint functional block and select *Configure* from the shortcut menu. The LonPoint Plug-in window associated with the selected functional block opens.

LonPoint Applications

Each LonPoint device has a *LonPoint Application*. This is the application firmware in the device. All LonPoint devices are shipped with applications already loaded. The version of the application which shipped in the device is printed on the LonPoint device label. Newer versions of applications may become available and may be loaded into the LonPoint device at any time using an LNS tool such as the LonMaker tool (see *Loading a New Application into a Device* in Chapter 5 of the *LonMaker for Windows User's Guide*).

The LonPoint AI-10, AO-10, DI-10, and DO-10 applications have been upgraded to version 3, but many of the devices currently available still have the version 2 applications loaded into them. The version 3 applications have been certified to meet the LONMARK Interoperability Guidelines. The program ID for all four applications has been changed to indicate a LONMARK. The Digital Output, AFB, and PID functional blocks have been enhanced in the version 3 applications as described in the LonPoint Read Me First document in the LonPoint program folder. Each functional block has an independent version number that is displayed on its `Status` tab.

When you first install a LonPoint device in a network, you should upgrade it to version 3 if it has a version 2 application. You do not have to upgrade devices that have already been installed.

Each application contains a program ID that identifies the class, subclass, model number, and application version. If an attempt is made to load an application into a device that currently contains an application with a different class or subclass, the LonMaker tool will issue an application load warning. The following table summarizes the program ID information for all version 2 and version 3 LonPoint applications:

Device Application	**Version 2**	**Version 3**
DI-10	Device Class: 0532 Device Subclass: 8A04 Model Number: 02 LONMARK Certified: No	Device Class: 052A Device Subclass: 8A04 Model Number: 03 LONMARK Certified: Yes
DO-10	Device Class: 0533 Device Subclass: 8A04 Model Number: 02 LONMARK Certified: No	Device Class: 0533 Device Subclass: 8A04 Model Number: 03 LONMARK Certified: Yes
AI-10	Device Class: 0518 Device Subclass: 8A04 Model Number: 02 LONMARK Certified: No	Device Class: 0518 Device Subclass: 8A04 Model Number: 03 LONMARK Certified: Yes
AO-10	Device Class: 0519 Device Subclass: 8A04 Model Number: 02 LONMARK Certified: No	Device Class: 0519 Device Subclass: 8A04 Model Number: 03 LONMARK Certified: Yes
SCH-10	Device Class: 0114 Device Subclass: 8A04 Model Number: 02 LONMARK Certified: No	Device Class: 0114 Device Subclass: 8A04 Model Number: 03 LONMARK Certified: Yes
DIO-10	No version 2 DIO-10 application available.	Device Class: 0528 Device Subclass: 8A04 Model Number: 03 LONMARK Certified: Yes
DL-10	No version 2 DL-10 application available.	Device Class: 0104 Device Subclass: 8A04 Model Number: 03

		LONMARK Certified: Yes

The DI-10 device's version 2 and version 3 applications have different Device Class values, so the LonMaker tool will issue a warning when an attempt is made to load a version 3 DI-10 application into a device which currently contains a version 2 DI-10 application. This warning may be safely ignored.

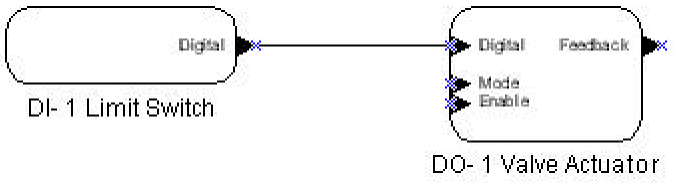
The SCH-10 device supports two applications, the SCH-10 application and the DL-10 application. Attempting to load the DL-10 application into a SCH-10 device which currently contains the SCH-10 application or vice versa will cause a warning in the LonMaker tool that an attempt is being made to load an application with a different Device Class or Device Subclass into a device already containing an application. In this case, this warning may be safely ignored.

Terminology Used in this Manual

The following sections contain definitions for several concepts presented in this manual in conjunction with the LonPoint functional blocks.

Upstream and Downstream

The terms *upstream* and *downstream* represent the flow of information. If functional block A sends information to functional block B, B is said to be *downstream* of A, and A is said to be *upstream* of B. For example, in the following figure, the DO- 1 Valve Actuator is downstream of the DI- 1 Limit Switch.



Heartbeats

A *heartbeat* is a network variable update that is automatically sent if the network variable has not otherwise been updated for a configurable length of time.

Most LonPoint output network variables can be configured to send heartbeat updates. Most LonPoint input network variables can monitor heartbeats from upstream functional blocks to detect device failures using heartbeat receive checking. If a heartbeat is not received within the specified amount of time, the functional block will go into heartbeat failure and cease propagating output network variables, and if the functional block is associated with a hardware output it will cause that output to go to its configured default value. The functional block will return to normal operation once the heartbeat is restored.

The heartbeat send time should be approximately 4 times as frequent as the corresponding expected heartbeat receive time to allow for lost messages.

Heartbeat failure propagates downstream in a LonPoint system. Once a LonPoint functional block receives a heartbeat failure, it will stop sending heartbeats, causing any downstream functional blocks that check heartbeats to go into heartbeat failure as well. Functional blocks which go into heartbeat failure will output their default values (see *Defaults*, below)

You can disable heartbeat receive checking on downstream functional blocks. This may be desirable in situations where it makes sense for the application to continue to use the last known valid data rather than change to the configured default network variable or hardware output.

Throttle

The *throttle* option limits the rate of updates on an output network variable. Use throttling to reduce network traffic. In order to minimize network traffic, set the throttle value to the longest interval between updates on the output network variable that is compatible with correct system operation. Turn off throttling by setting the value to 0. The AI and PID functional blocks produce data at their configured scan interval, and do not have a throttle configuration property.

Default Values

Default values are values that apply to hardware outputs and both input and output network variables.

A functional block will use its default hardware or network variable output value when any of the following conditions occur:

- The functional block has just come out of reset and has not yet received any network variable updates.
- The functional block has an enable input network variable which is turned off.
- The functional block senses a heartbeat failure.

Default values for input network variables are used when any of the following conditions occur:

- The network variable has not yet received any updates since its last reset.
- The network variable is not connected. There are often situations where one or more inputs on a functional block will not be connected. The network variable should be configured with an appropriate default input value. This also allows you to manually set the value of an input network variable for debugging and testing.

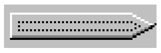
Override Values

Override values are values that apply to output network variables and hardware outputs. LonPoint functional blocks can be forced into an override state with the LonMaker tool or the LonPoint Plug-in. When a functional block is in the override state, the output network variables and hardware outputs, if any, will be set to their override values. Override values and the override state are preserved across power cycles and resets.

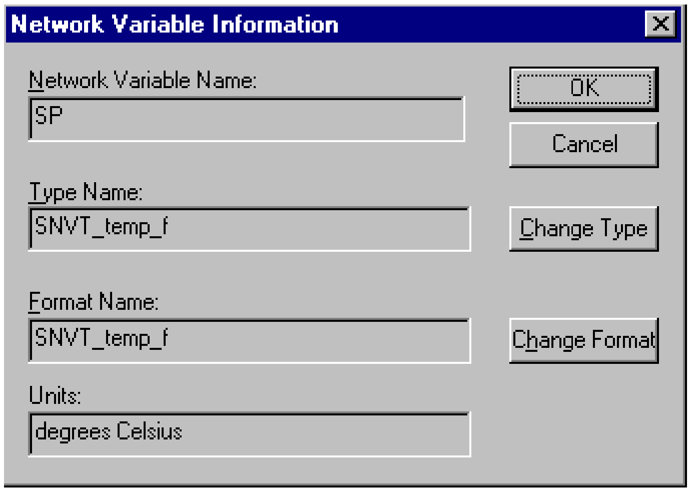
While override values are stored in both the device and in the LNS database, the override state of a device is stored only in the device, not in the LNS database. If you replace a device with the LonMaker tool, the old device's override values are transferred to the new device. **The new device will not necessarily be in the same override state as the old device.**

Network Variables

On the first tab of the LonPoint Plug-in window for each LonPoint functional block (except for the Node Object) there is a network variable shape for each network variable which looks like this:



To get information about a network variable, click on the network variable shape. The following `Network Variable Information` dialog appears:



Network Variable Name: SP

Type Name: SNVT_temp_f

Format Name: SNVT_temp_f

Units: degrees Celsius

This dialog contains the following information:

Network Variable Name	The name of the network variable. You can change this name using the LonMaker tool.
Type Name	The network variable type. If the type is a standard network variable type, the network variable may only be connected to network variables with the same type. If the network variable has a changeable type, the `Change Type` button will be enabled.
Format Name	The network variable format. The format determines how data from the network variable will be formatted when it is displayed or input by an LNS tool such as the LonMaker tool. If the format is changeable, the `Change Format` button will be enabled.
Units	The type of units the network variable uses. If the network variable does not use a specific type of unit, this field will be empty.

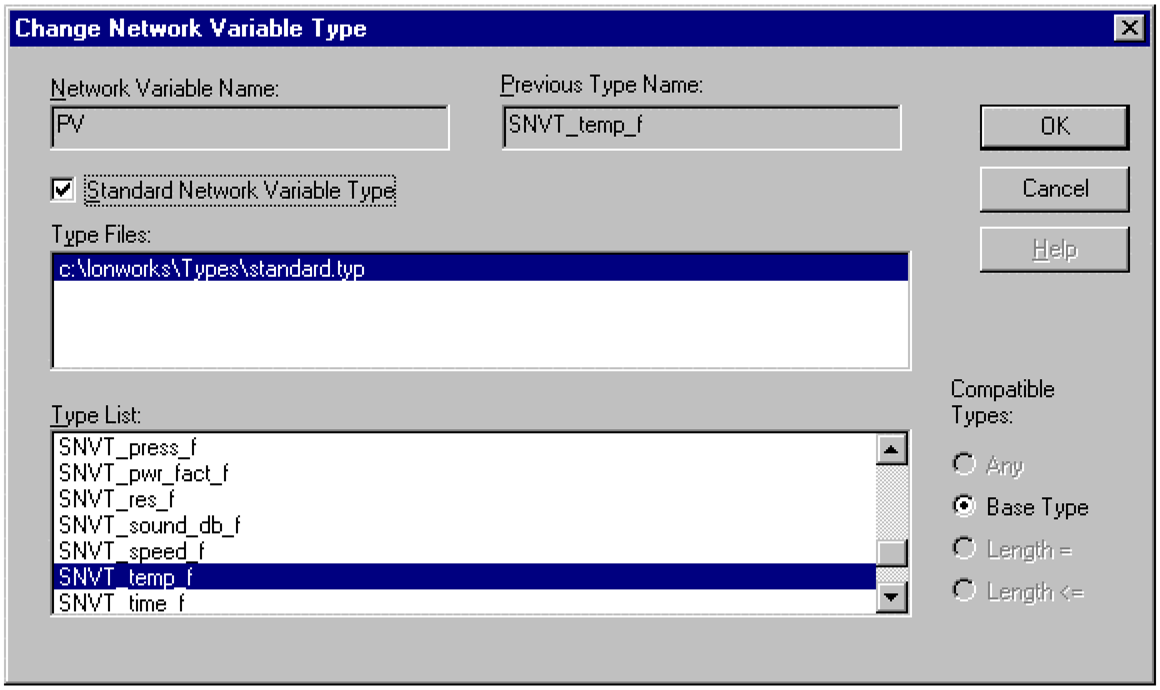
To change a network variable type, click the `Change Type` button. To change a network variable format, click the `Change Format` button.

Changing Network Variable Types

You can change the types of some network variables in LonPoint functional blocks. Most changeable types are either floating-point or enumerated network variables, as well as the network variables in the Type Translator functional block.

Changing the type of a network variable will reset its format to the default for that type. This affects the display of related fields in the LonPoint Plug-in. See Network Variable Formats for more information.

To change the type of a network variable, click the network variable shape in the LonPoint Plug-in window, then click the `Change Type` button in the `Network Variable Information` dialog. The following `Change Network Variable Type` dialog appears:



Network Variable Name: PV

Previous Type Name: SNVT_temp_f

[✓] Standard Network Variable Type

Type Files: c:\lonworks\Types\standard.typ

Type List: SNVT_press_f, SNVT_pwr_fact_f, SNVT_res_f, SNVT_sound_db_f, SNVT_speed_f, SNVT_temp_f (selected), SNVT_time_f

Compatible Types: [✓ Base Type]

This dialog contains the following fields:

Network Variable Name	The name of the network variable. This field is read-only. You can change the name of a network variable using the LonMaker tool.
Previous Type Name	The current network variable type. This field is read-only.
Standard Network Variable Type	Specifies whether the new network variable type is a standard type or a user defined type. Enable this option if you want to use a SNVT. (See the *SNVT Master* list in the *LNS Utilities and LONMARK Reference* help file.) Disable this option if you want to select a user-defined type.
Type Files	Lists all available type files from the device resource file catalog. If the Standard Network Variable Type option is selected, only the standard type file is listed. Select the

	type file containing the new network variable type from this list.
Type List	Lists all network variable types in the selected file that are compatible with the selected network variable. Select the new network variable type from this list.
Compatible Types	Indicates how compatibility is decided. For all functional blocks except the Type Translator, this is by `Base Type` (e.g., floating-point or enumeration). The Type Translator uses the `Length <=` option, listing all types less than or equal to four bytes long. You cannot change this option.

Network Variable Formats

Many floating-point network variable types have at least two standard formats: one for SI units (i.e. the metric system, which are the default units of most SNVTs), and one for comparable U.S. units. There may also be more than one format for either of these two basic systems (e.g. SNVT_flow has U.S. formats for both gallons per second and cubic feet per minute). If there is more than one format, one of them is considered the default. The LonMaker and LonPoint Plug-in installation programs allow installing format files that have either all SI units as the default, or all U.S. units as the default.

Selecting a format does not affect the actual network variable data on the network or configuration property data in the device. This data is always in the native units of the network variable or configuration property. The format only affects how the data is displayed (or interpreted when doing data entry). Any format that does not use the native configuration property or network variable units will convert the data to or from the native units using conversion values associated with that format. For example, if U.S. units are being used, data being entered is converted internally to SI units, and converted back to U.S. units for display. Due to floating-point rounding, values displayed may not always be exactly what was entered.

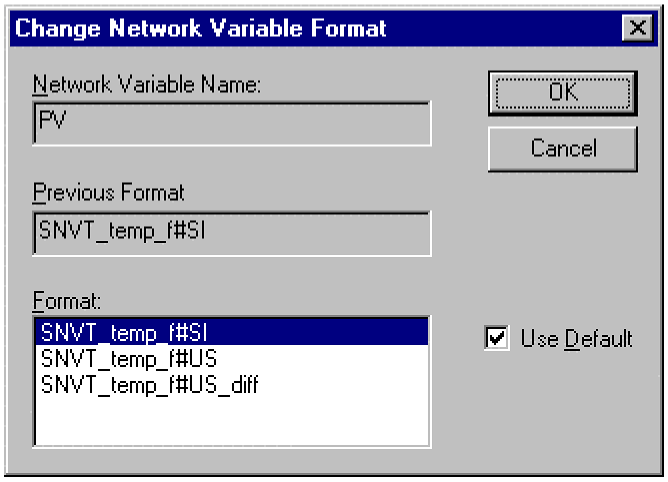
Changing Network Variable Formats

You can change the format of most LonPoint floating-point network variables, as well as the network variables in the Type Translator functional block.

Note:

> Changing the format of a network variable affects the display of related fields in the LonPoint Plug-in, the LonMaker tool, and other third-party LNS applications. The format change will be immediately visible in the LonPoint plug-in. However, other applications may not reflect the format change until some action causes the format to be refreshed; this can be accomplished by restarting the application or by some explicit command, such as the *LonMaker Browser's* `Refresh All` command.

To change a network variable format, click the network variable shape in the LonPoint Plug-in window, then click the `Change Format` button in the `Network Variable Information` dialog. The following `Change Network Variable Format` dialog appears:


Change Network Variable Format

Network Variable Name: PV

Previous Format: SNVT_temp_f#SI

Format:
SNVT_temp_f#SI
SNVT_temp_f#US
SNVT_temp_f#US_diff

This dialog contains the following fields:

Network Variable Name	The name of the network variable. This field is read-only. You can change the name of a network variable using the LonMaker tool.
Previous Format	The current format of the selected network variable. This field is read-only.
Format	Lists the available formats for the selected network variable type. Select a format from this list to explicitly determine the type of units used in the selected network variable and its associated configuration properties.
Use Default	Automatically selects the default format from the Format field. Use this option to reset a network variable to use its default format.

When you select a new format for a network variable, the units of that format will be indicated in the LonPoint Plug-in anywhere there is a field which contains data that matches the network variable type (e.g. an override value). Data in these fields is entered and displayed in those units.

2

The Digital Input Functional Block: Application and Plug-in

This chapter describes how to configure a Digital Input functional block using the LonPoint Plug-in.

The Digital Input Functional Block

The Digital Input functional block reads the state of a digital signal. This value is then processed, and the resulting digital value is sent to the output network variable. The following figure and table summarize the inputs and outputs of the Digital Input functional block:

DI- 1

Output Network Variables

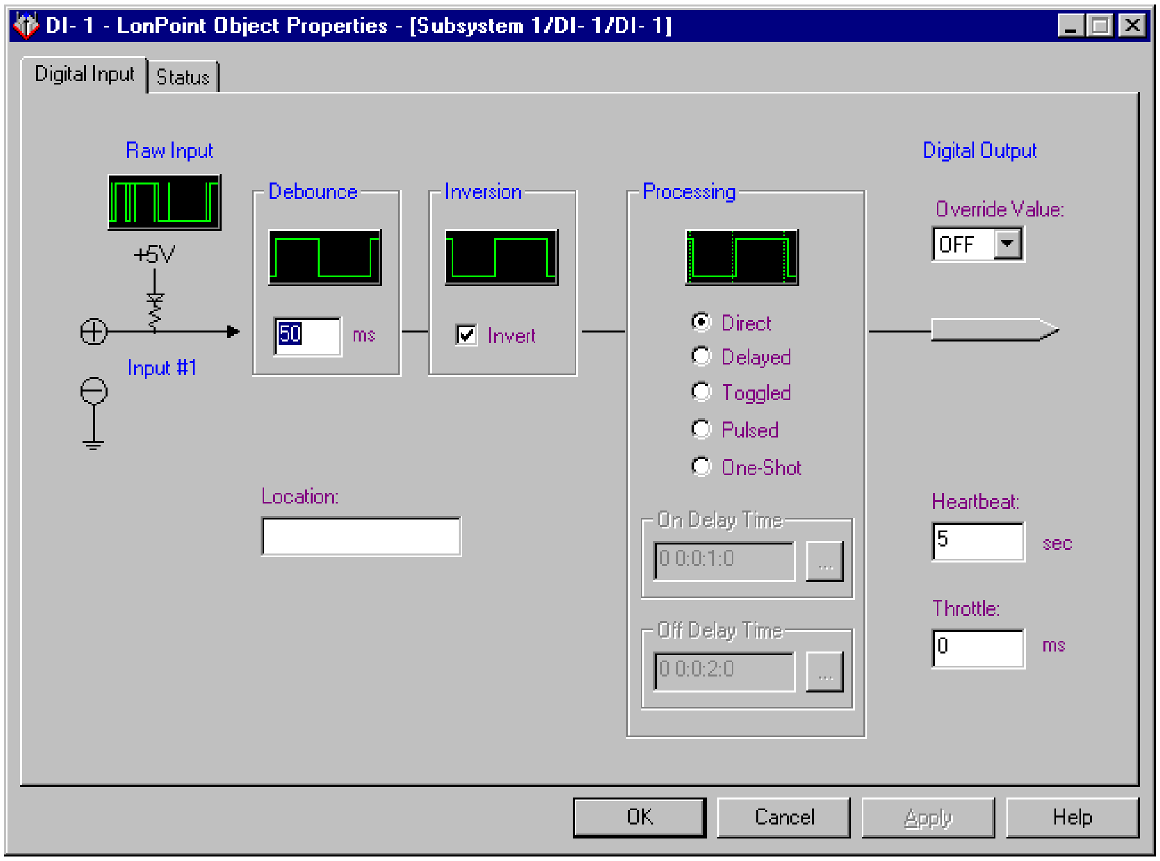
Default name	Default type	Description
Digital	SNVT_switch	The Digital output network variable driven by the sensor.

Configuring the Digital Input with the LonPoint Plug-in

Right-click a Digital Input functional block and select *Configure* from the shortcut menu to open the Digital Input window of the LonPoint Plug-in. You can also choose *Plug-ins* from the shortcut menu and then select `Configure LonPoint Object` from the dialog box. The Digital Input functional block window has two tabs, `Digital Input` and `Status`.

Digital Input

The `Digital Input` tab, pictured below, provides a graphical interface to the LonPoint Digital Input (DI) functional block. It allows you to determine how data from a physical digital input is interpreted and what value is sent over the network on the DI functional block's output network variable.



DI- 1 - LonPoint Object Properties - [Subsystem 1/DI- 1/DI- 1]

Digital Input | Status

Debounce: 50 ms

Invert: ✓

Processing: (● Direct) ○ Delayed ○ Toggled ○ Pulsed ○ One-Shot

On Delay Time: 0 0:0:1:0

Off Delay Time: 0 0:0:2:0

Location:

Override Value: OFF

Heartbeat: 5 sec

Throttle: 0 ms

The data flow in this tab is left to right. The raw digital input signal can be modified by the debounce configuration property, then passed to the configurable inversion function, then that data is passed to the remaining processing steps.

Use the plug-in to set the following configuration properties:

Debounce	The debounce time for the digital input. This is the amount of time, in milliseconds, that the input must remain constant for the value to be passed on to the rest of the functional block. Set this value to 0 to turn off the debounce function.
Invert	Specifies whether or not the data from the digital input is inverted before further processing is done.
Location	The location string for this digital input. This property can be used to document the associated sensor's location within the plant so it can be easily

found. This field may contain up to 30 characters. This value is separate from the device's location property.

Processing

Affects the translation of incoming data to the value passed to the output network variable. There are five processing options: `Direct`, `Delayed`, `Toggled`, `Pulsed`, and `One-Shot`.

The `Direct` option causes data to be output directly after the debounce and invert functions have executed.

The `Delayed` option specifies a delay from a change on the input to an update on the output network variable. A change from Off to On is delayed by the time indicated in the `On Delay Time` field. A change from On to Off is delayed by the time indicated in the `Off Delay Time` field.

The `Toggled` option causes the output data to toggle, or change state, every time the input data changes from Off to On. For example, if a Digital Input functional block is attached to a push button and the functional block is configured with the `Toggled` option, the network variable output value will change every time the button is pushed.

The `Pulsed` option generates a pulse on the output network variable every time the input data changes from Off to On. The pulse is generated after a delay specified on the `On Delay Time` field. The duration of the pulse is specified on the `Pulse Time` field. It does not matter how long the input data remains on, the output will always send a pulse of the specified length after waiting the configured delay. If a delayed pulse is re-triggered during a pulse, the delay will be ignored. If a pulse is re-triggered during a delay, the trigger will be ignored.

The `One-Shot` option generates a pulse on the output network variable every time the input data changes from Off to On. The pulse is generated after a delay specified on the `On Delay Time` field. The duration of the pulse is specified in the `Pulse Time` field. If the input data changes from Off to On while the pulse is being sent, the pulse timer will be reset (i.e., if a two second pulse was retriggered after one second, the output would be on for two more seconds, or three seconds total).

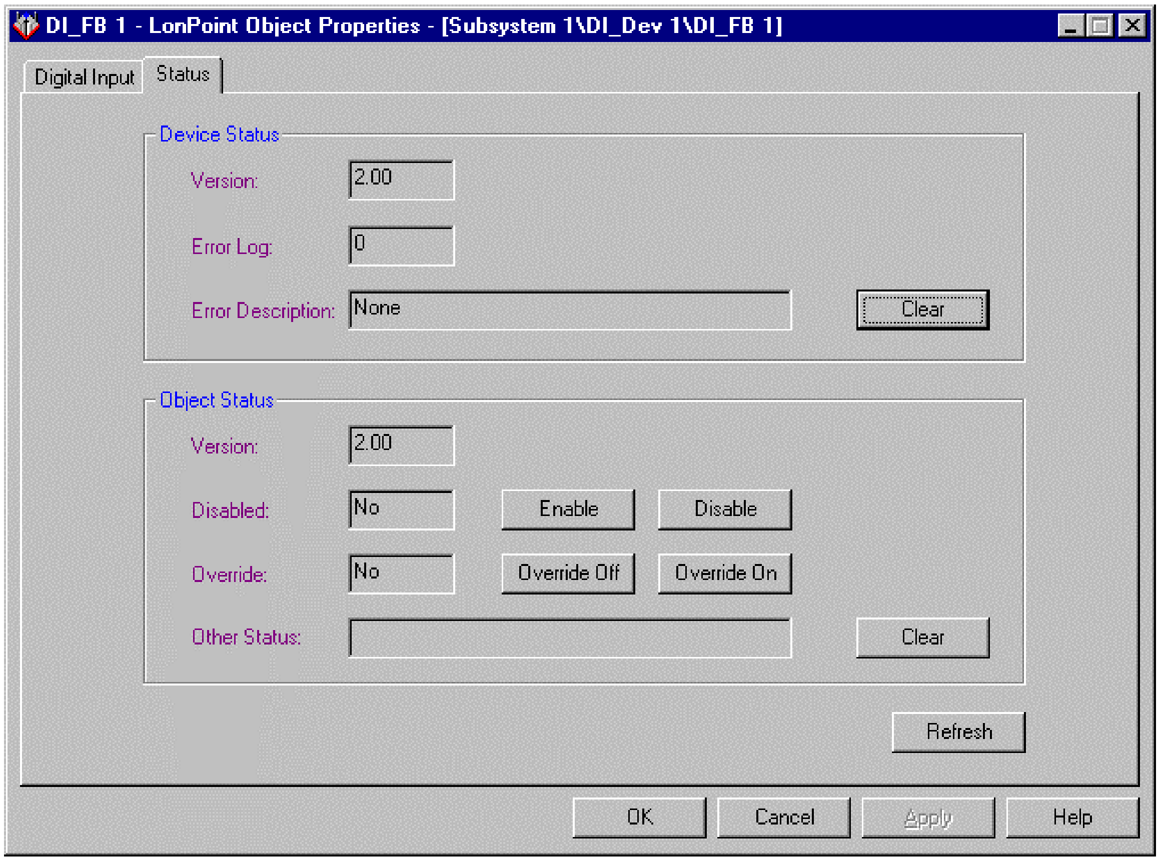
On Delay Time

Use this field with the `Delayed`, `Pulsed`, and `One-Shot` Processing options. This value determines the length of the delay in the change from the off state to the on state. To change the value, click the … button to the right of the time value to be changed and enter

	the new values to be used for the delay. The valid range for this value is from zero to 48 days, 23 hours, 59 minutes, 59 seconds, and 999 milliseconds.
Off Delay Time	Use this field with the `Delayed Processing` option. This value determines the length of the delay in the change from the on state to the off state. To change the value, click the ... button to the right of the time value to be changed and enter the new values to be used for the delay. The valid range for this value is from zero to 48 days, 23 hours, 59 minutes, 59 seconds, and 999 milliseconds.
Pulse Time	If you select either the `Pulsed` or the `One-Shot Processing` options, the `Pulse Time` field appears in place of `Off Delay Time`. This value determines the length of time the on state remains active. To change the value, click the ... button to the right of the time value to be changed and enter the new values to be used for the pulse. The valid range for this value is from zero to 48 days, 23 hours, 59 minutes, 59 seconds, and 999 milliseconds.
Override Value	Determines the value sent to the network via the output network variable if the functional block is put into override mode as described in the next section.
Heartbeat	Determines how often the functional block sends a heartbeat over the network. The behavior of the system in case of a missed heartbeat is determined by the functional blocks which fail to receive the heartbeat. Setting this property to 0 disables the heartbeat for this functional block. Disabling the heartbeat causes the output network variable to only be transmitted in response to a changed input value.
Throttle	Limits how often data is sent over the network. Setting this property to 0 disables throttling.

Status

This tab allows you to view and change the status of a Digital Input functional block. This tab appears as follows:



Version:	2.00
Error Log:	0
Error Description:	None
Version:	2.00
Disabled:	No
Override:	No
Other Status:	

This tab contains the following fields and buttons:

Device Version	The version number of the application in this device. The minor version number (after the decimal point) is always read from the device itself. If the network is unattached or Offnet, the minor version number will read XX (e.g. 2.XX).
Error Log	The most recently logged error on the device. This error may not apply to the functional block you are configuring.
Error Description	A description of the most recently logged error.
Clear (Device Status)	Clears the status of the device, including the Error Log. This also clears other device communication statistics information that is not displayed here (e.g., Lost Messages). If you wish to examine the other information before clearing it, use the Test command described under *Managing Devices, Functional*

	Blocks, and Routers in the *LonMaker for Windows User's Guide*.
Object Version	The version number of this functional block. The minor version number (after the decimal point) is always read from the device itself. If the network is unattached or Offnet, the minor version number will read `XX` (e.g. `2.XX`).
Disabled	Indicates `Yes` if this functional block is disabled, `No` if it is enabled, and `?` if the plug-in is not in communication with the device. If you can communicate with the device, you can change the state of the functional block by clicking the `Enable` and `Disable` buttons. See *Managing Devices, Functional Blocks, and Routers* in the *LonMaker for Windows User's Guide* for more information.
Override	Indicates `Yes` if this functional block is in override, `No` if it is not in override, and `?` if the plug-in is not in communication with the device. If you can communicate with the device, you can change the mode of the functional block by clicking the `Override Off` and `Override On` buttons. See *Managing Devices, Functional Blocks, and Routers* in the *LonMaker for Windows User's Guide* for more information.
Other Status	Displays other information relating to the functional block status (e.g. communication error, range error).
Clear (Object Status)	Clears the status of the functional block.
Refresh	Refreshes the information in this tab. Any changes to the data displayed in this tab since the plug-in was started (by using the LonMaker tool's Manage command, for example) will not be updated until this button is pressed. Some LonPoint plug-ins can experience a temporary lock-out. This can be cleared by clicking this button.

3

The Digital Input/Counter Functional Block: Application and Plug-in

This chapter describes how to configure a Digital Input/Counter functional block using the LonPoint Plug-in.

The Digital Input/Counter Functional Block

The Digital Input/Counter functional block reads the state of a digital signal. This value is then processed, and the resulting value is sent to the digital output network variable. The Digital Input/Counter is a superset of the Digital Input; it contains all the functionality of the Digital Input plus additional capabilities. The following figure and tables summarize the inputs and outputs of the Digital Input/Counter functional block:

Control
Digital
Analog

DC-1

Input Network Variables

Default name	Default type	Description
Control	`UNVT_ count_control` (changeable)	The Control network variable is an enumerated type that controls counting functions.

Output Network Variables

Default name	Default type	Description
Digital	`SNVT_switch`	The Digital output network variable. The meaning varies according to the processing option. See the *Processing Options*, later in this chapter, for more information.
Analog	`SNVT_temp_f` (changeable)	The Analog output network variable. The meaning varies according to the processing option. See the *Processing Options*, later in this chapter, for more information.

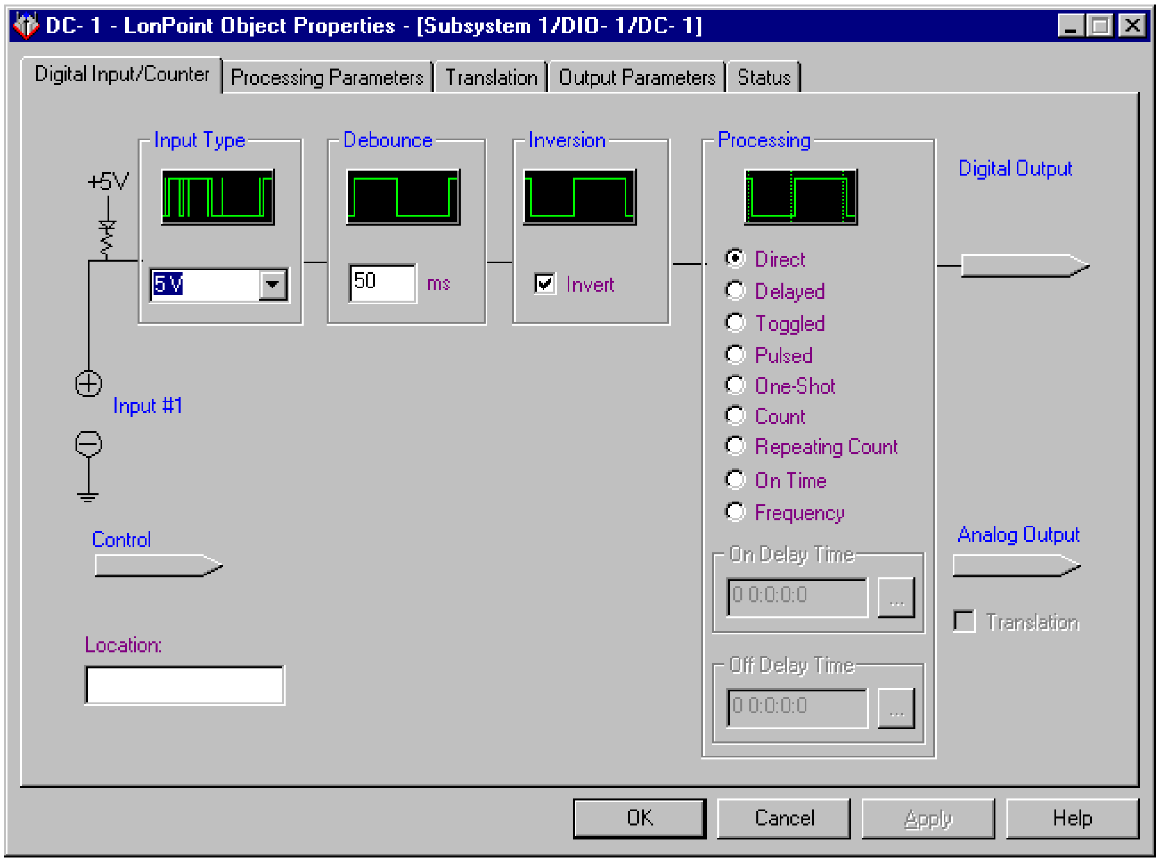
This functional block contains an analog output network variable which is used for several different purposes depending on the processing option, and a control input network variable which may be used to clear, hold, pause, or preset the analog output. The digital output network variable type is always `SNVT_switch`. The analog output network variable type is changeable. See Chapter 1 for a discussion of network variables.

Configuring the Digital Input/Counter with the LonPoint Plug-in

Right-click a Digital Input/Counter functional block and select *Configure* from the shortcut menu to open the Digital Input/Counter window of the LonPoint Plug-in. The Digital Input/Counter functional block window contains the following tabs, `Digital Input/Counter`, `Processing Parameters`, `Translation`, `Output Parameters`, and `Status`.

Digital Input/Counter

The `Digital Input/Counter` tab, pictured below, provides a graphical interface to the LonPoint Digital Input/Counter (DC) functional block. It allows the user to determine how data from a hardware digital input is interpreted and what value is sent over the network on the Digital Input/Counter functional block's output network variables.



Input Type: 5 V
Debounce: 50 ms
Inversion: [✓ Invert]
Processing: [✓ Direct] Delayed Toggled Pulsed One-Shot Count Repeating Count On Time Frequency
On Delay Time: 0 0:0:0:0
Off Delay Time: 0 0:0:0:0

The data flow in this tab is left to right. The raw digital input signal is checked against the input voltage threshold in order to determine the logical state of the input (on or off) and then can be modified by the debounce configuration property, then passed to the configurable inversion function, then passed to the remaining processing steps. Click the `Control` and `Analog Output` network variable buttons to change the types of these network variables.

Use the plug-in to set the following configuration properties:

Input Type	Determines the voltage threshold levels which trigger a state change. The value may of Dry Contact, 5V, 12V, 24V, or 31V. Use Dry Contact to determine if a relay or switch has been closed.
Debounce	The debounce time for the digital input. This is the amount of time, in milliseconds, that the input must

	remain constant for the value to be passed on to the rest of the functional block. Set this value to 0 to turn off the debounce function.
Inversion	Specifies whether or not the data from the digital input is inverted before further processing is done. Determines whether a high voltage level is interpreted as an On state (not inverted) or an Off state (inverted).
Processing	Affects the translation of incoming data to the value passed to one of the output network variables. There are nine processing options: `Direct`, `Delayed`, `Toggled`, `Pulsed`, `One-Shot`, `Count`, `Repeating Count`, `On Time`, and `Frequency`. The `Direct`, `Delayed`, `Toggled`, `Pulsed`, and `One-Shot` options cause the Digital input/Counter functional block to behave identically to the Digital Input functional block, and the Control and Analog output network variables will not be used. See *Processing Options* in the next section for a description of the processing options.
Location	The location string for this digital input/counter. This property can be used to document the associated hardware's location within an installation so it can be easily found. This field may contain up to 30 characters. This value is separate from the device's location property.

Processing Options

The following sections describe the processing options selected in the Digital Input/Counter tab.

Direct

Causes data to be output directly to the digital output network variable after the debounce and invert functions have executed.

Delayed

Specifies a delay from a change on the input to an update on the digital output network variable. A change from Off to On is delayed by the time indicated in the `On Delay Time` field. A change from On to Off is delayed by the time indicated in the `Off Delay Time` field.

Toggled

Causes the digital output network variable value to toggle, or change state, every time the input data changes from Off to On. For example, if a Digital Input/Counter functional block is attached to a push button and the functional block is configured with the `Toggled` option, the digital network variable output value will change every time the button is pushed.

Pulsed

Generates a pulse on the digital output network variable every time the input data changes from Off to On. The pulse is generated after a delay specified on the `On Delay Time` field. The duration of the pulse is specified on the `Pulse Time` field. It does not matter how long the input data remains on, or if the input transitions from Off to On again during the pulse, the output will always send a pulse of the specified length after waiting the configured delay.

One-Shot

Generates a pulse on the output network variable every time the input data changes from Off to On. The pulse is generated after a delay specified on the `On Delay Time` field. The duration of the pulse is specified in the `Pulse Time` field. If the input data changes from Off to On while the pulse is being sent, the pulse timer will be reset (i.e., if a two second pulse was retriggered after one second, the output would be on for two more seconds, or three seconds total). If a delayed pulse is re-triggered during a pulse, the delay will be ignored. If a pulse is re-triggered during a delay, the trigger will be ignored.

Count

Counts Off to On transitions. When the hardware detects an input change from off-to-on, an internal counter is incremented. The count value is processed and sent to the Analog network variable. An appropriate network variable type for the Analog output network variable in this mode is `SNVT_count_f`. The Control network variable governs processing of the count as described in *Using The Control Network Variable*, later in this chapter. The input is sampled and the Digital network variable updated at 500ms intervals. The maximum number of events that can be counted is 16,777,215.

The count values are kept in RAM. The count is copied to non-volatile EEPROM once per day just past midnight as determined by the DIO-10 node object, whenever the node is sent offline, and whenever the functional block is commanded to go to the disabled state.

Power cycles, hard-resets, soft-resets, or watchdog timeouts will lose the RAM count values. After recovery from any of these events the last count stored in EEPROM will be loaded into the RAM counter.

If the hardware input has been configured for Dry Contact, the input may be configured to be debounced with a configurable time period. If the hardware input has been configured for voltage input (5V, 12V, 24V, or 32V) debounce is not supported.

The maximum input frequency when configured for count processing is dependent on the input hardware configuration as follows:

DRY_CONTACT:

minimum period = min(500ms, 2 * debounce interval)
maximum frequency = 1/minimum period

5V, 12V, 24V, or 32V :

maximum frequency = 20kHz

Repeating Count

Counts the off to on transitions from zero up to a configurable value called the *terminal count*. When the Digital Input/Counter reaches the terminal count the count is reset to zero. For example, a terminal count of 4 would produce values of zero, one, two, three, zero, one, two, three, … on the `Analog` output network variable (an appropriate network variable type for the `Analog` output network variable in this mode is `SNVT_count_f`).

The Digital Input/Counter signals that it has reached the terminal count by asserting the `Digital` network variable. The `Digital` network variable stays on until the `Control` network variable is set to CLEAR.

If the hardware input has been configured for Dry Contact, the input may be configured to be debounced with a configurable time period. If the hardware input has been configured for voltage input (5V, 12V, 24V, or 31V), debounce is not supported.

If the `Control` network variable is not connected, the Repeating Count processing will free-run and count repeatedly from zero to terminal count –1 on the `Analog` network variable. In this case the maximum input frequency is dependent on input hardware configuration as follows:

DRY_CONTACT:

minimum period = min(500ms, 2 * debounce interval)
maximum frequency = 1/minimum period

5V, 12V, 24V, or 32V :

maximum frequency = 20kHz

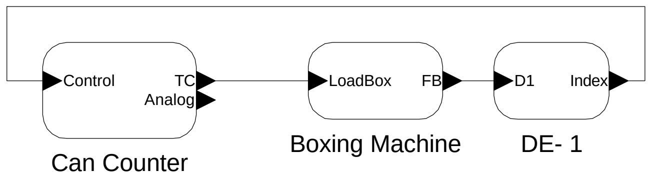
If the `Control` network variable is connected, the Repeating Count maximum input frequency is dependent on the time between the `Digital` network variable going to On and downstream logic setting the `Control` network variable to CLEAR for a short enough time period such that the count function does not reach the terminal count a second time.

Repeating count example:

A canning application needs to determine when four cans have passed down a chute to tell a downstream boxing machine to package the cans. The boxing machine indicates when it has completed its task. On the Digital Input/Counter, select Repeating Count mode, set the terminal count to four, and the Default Input for the `Control` network variable to RUN via the LonPoint Plug-in. After reset the count is set to 0, as cans go by, the count reported by the `Analog` network variable goes; 0, 1, 2, 3, then, on the arrival of the next can, the `Digital` output goes On and the count as reported by the Analog network variable is set to 0. Cans continue: 1, 2…, then, some downstream device controlling the boxing process sees the Digital network variable to be on, boxes the cans, and sends CLEAR to the Control network variable. The digital output goes Off. Cans continue three, zero and the process repeats.

There are no race conditions, however there is a performance requirement – the Control network variable must be cleared before the terminal count is reached a second time in order to get a new signal. If the Control network variable is not cleared before the terminal count is reached, counting will continue between zero and the terminal count, the Digital network variable will remain on, and the overrange bit of the object status for the Digital Input/Counter function block will be set. This status bit can be used as an indicator that the control loop is not being closed. If the Control network variable is not connected, the overrange status bit will not be set.

It is good design practice for the output network variable that updates the Control input network variable to have a send heartbeat so that the loop works even if some messages are lost.



On-Time

Measures the accumulated on-time of the hardware input. A typical application for this mode is monitoring equipment usage time in order to determine when maintenance is due. Each half-second, an internal counter is incremented if the input is on. The count value is processed and sent to the Analog network variable (an appropriate network variable type for the Analog output network variable in this mode is SNVT_time_f; the units are seconds). The Control network variable governs processing. The input is sampled and the Digital network variable is updated at 500ms intervals.

The maximum on time that can be measured in this mode is 20 years.

The on-time value is kept in RAM. The value is copied to non-volatile EEPROM once per day just past midnight as determined by the node object, whenever the node is sent offline, and whenever the functional block is commanded to go to the disabled state. Power cycles or watchdog timeouts will lose the RAM on-time values. After recovery from any of these events the last count stored in EEPROM will be loaded into the RAM counter

If the hardware input has been configured for Dry Contact the input may be configured to be debounced with a configurable time period. If the hardware input has been configured for voltage input (5V, 12V, 24V, or 31V) debounce is not supported

Frequency

Measures the frequency of the input signal in the range of 1Hz to 20kHz. Minimum resolution of the frequency measurement is <0.5% over the full measurement range. The Analog network variable reports the instantaneous measured frequency or a translated representation of it. The Digital network variable is not used with this processing option.

On startup the Analog output network variable is not immediately available as the input is auto-ranged. This process takes up to four seconds.

The frequency measurement may be modified using a translation table. The translation table operates in the same manner as the translation table in the Analog input functional block found in the AI-10 LonPoint Module. The translation table is in turn followed by a filtering function. The filter averages over N samples, where N is the configured filter length. The supported range of N is 1 to 255. The equivalent filter time constant for this range is approximately 0.5 seconds to 50 seconds. No filtering is done if N equals 1.

If no translation is applied, an appropriate network variable type for the Analog output network variable in this mode is `SNVT_freq_f`. If translation is applied then the appropriate floating point SNVT for the process variable being measured should be used (e.g. `SNVT_flow_f`).

The input cannot be configured for dry contact when the frequency processing option is selected. Debounce is not supported. The hardware input may be configured for voltage input (5V, 12V, 24V, or 31V only).

Using The Control Network Variable

The `Control` network variable can be driven from another device on the network or an HMI application. It is sequenced in a way that allows count values to be sampled while continuing to count without interruption. It applies to the Count, Repeating Count, and On Time processing options. The `Control` network variable type is `UNVT_count_control`.

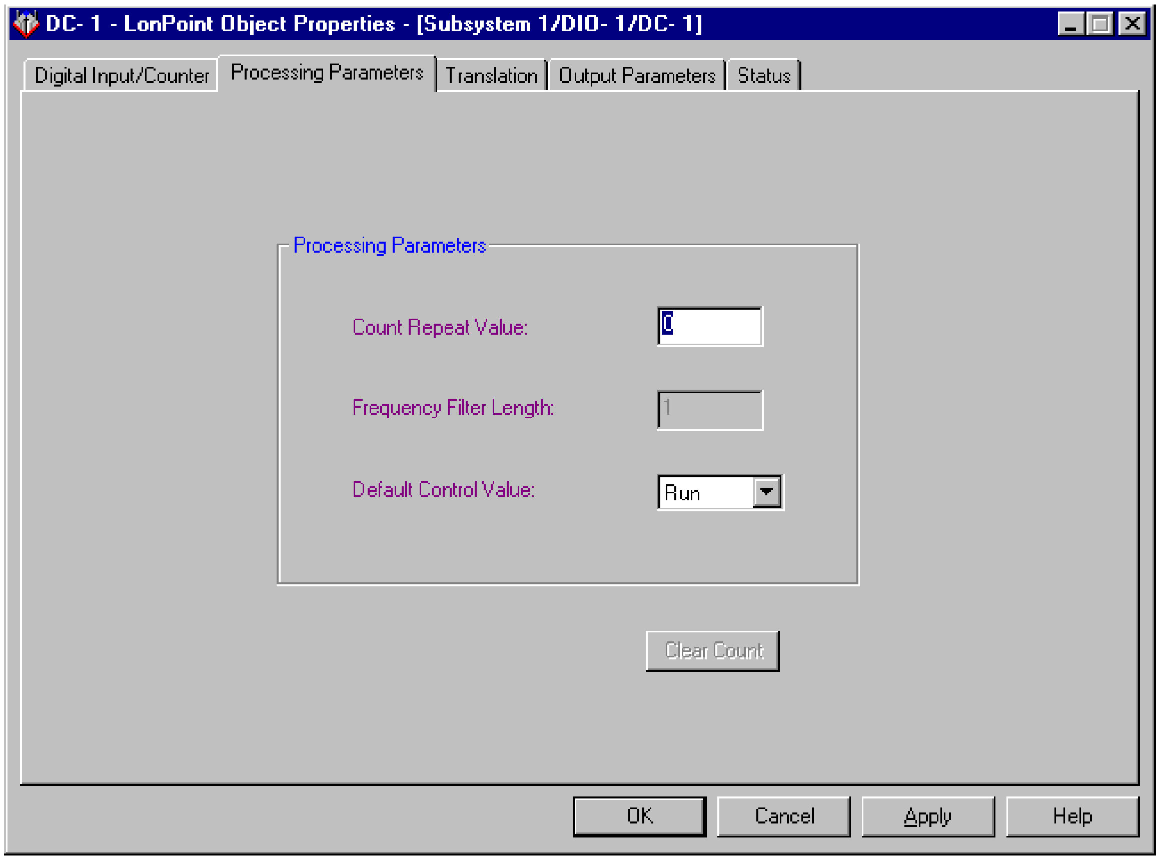
The following summarizes the input values to the `Control` network variable:

`RUN`	Enables event counting. The count value is output to the `Analog` network variable. The send-on-delta value is used to throttle the network variable, so the network variable is not necessarily propagated with every count.
`CLEAR`	Copies the internal count to the `Analog` network variable and then clears the internal count. Events continue to be counted, but the `Analog` network variable is not updated. This allows an HMI application to read the `Analog` network variable while the `Control` network variable is set to `CLEAR`. The HMI application can set the `Control` network variable to `RUN` after reading the `Analog` network variable.

HOLD	Holds the current value of the Analog network variable. Events are accumulated but the network variable is not updated. The HOLD state does not apply to the Repeating Count processing option. The Analog network variable will continue to update if Repeating Count is selected.
PAUSE	Disables event counting. The internal count value and Analog network variable are not updated.
PRESET	Presets the counter to the value specified in the preset value configuration property. The new value updates the stored count in EEPROM.

Processing Parameters

This tab appears as follows:



Count Repeat Value: 0

Frequency Filter Length: 1

Default Control Value: Run

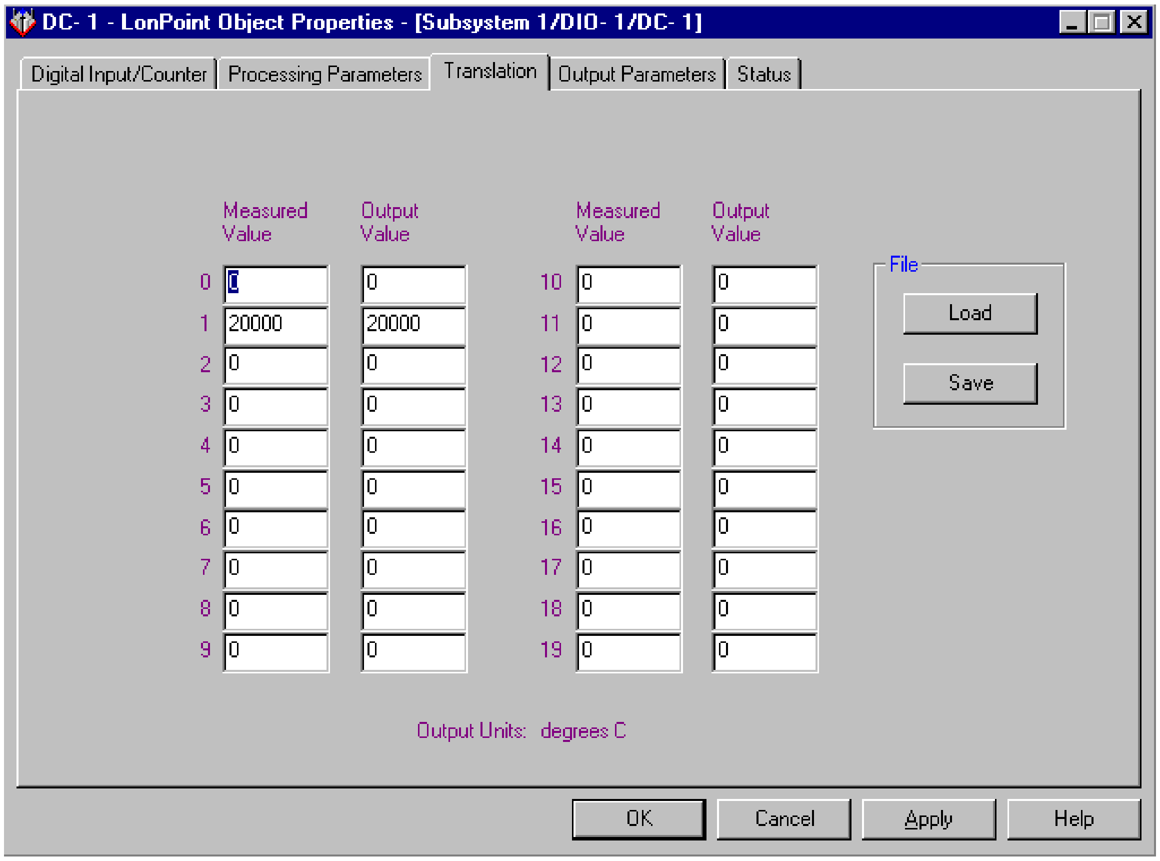
This tab allows you to set several parameters relating to the processing option selected in the Digital Input/Counter tab. The following parameters may be set using this tab:

Count Repeat Value	This parameter is only used if the processing option is set to `Repeating Count`. This integer value determines the terminal count used for this processing option (e.g. a value of 5 would result in a count of 0, 1, 2, 3, 4, 0, 1, 2, …).

Frequency Filter Length	This parameter is only used if the processing option is set to `Frequency`. This value may be set from 1 to 255 and determines the number of readings that are sampled and averaged before a value is passed to the Analog network variable.
Default Control Value	This parameter is only used if the processing option is set to `Count`, `Repeating Count`, or `On Time`. This value determines the control value if the Control input network variable is not connected or has not received an update since the devices last reset.

Translation

This tab appears as follows:



	Measured Value	Output Value		Measured Value	Output Value
0	0	0	10	0	0
1	20000	20000	11	0	0
2	0	0	12	0	0
3	0	0	13	0	0
4	0	0	14	0	0
5	0	0	15	0	0
6	0	0	16	0	0
7	0	0	17	0	0
8	0	0	18	0	0
9	0	0	19	0	0

Output Units: degrees C

This tab is only used if the `Frequency` processing option is selected and the `Translation` option is selected in the `Digital Input/Counter` tab.

This tab allows you to set up a translation table which translates the raw measured data into the values to be sent by the `Analog` output network variable after filtering (see the `Frequency` processing option, earlier in this chapter).

The translation table provides a means for mapping any input set of values into a different set of output values via pairs of input/output values. The Digital

Input/Counter software automatically linearly interpolates to determine translation values that lie between defined input/output pairs.

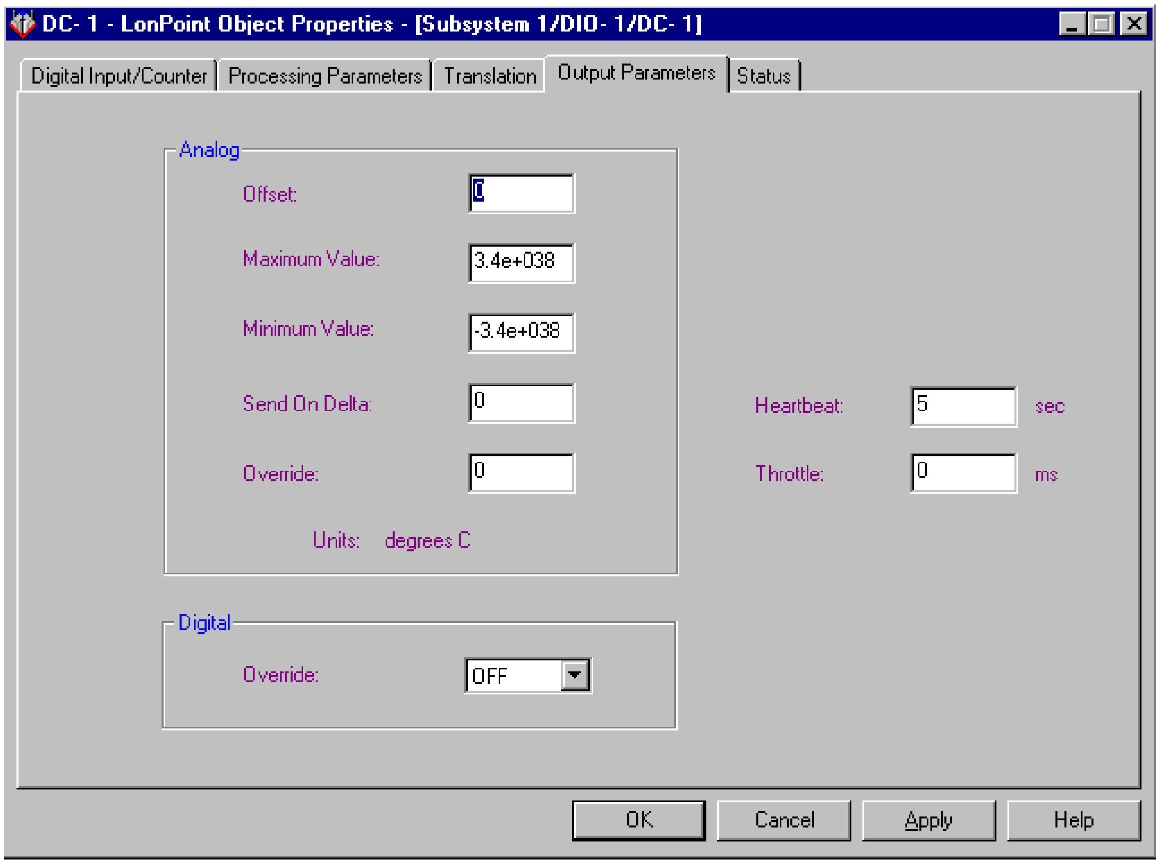
The table must contain at least two input/output pairs and may contain as many as twenty.

The only constraint on the Measured Value column is that the entries increase in value. To mark the end of the table, enter a measured value less than the preceding one.

Click the Save button to save translation table data into a tab-delimited text file. Click the Load button to load data from a tab-delimited text file into a translation table. Use these functions to save and reuse commonly used translation tables for your hardware inputs. You can also use custom LonMaker shapes to save frequently used configurations.

Output Parameters

This tab appears as follows:


DC- 1 - LonPoint Object Properties - [Subsystem 1/DIO- 1/DC- 1]

Analog

Offset: 0

Maximum Value: 3.4e+038

Minimum Value: -3.4e+038

Send On Delta: 0

Override: 0

Units: degrees C

Heartbeat: 5 sec

Throttle: 0 ms

Digital

Override: OFF

This tab allows you to determine parameters for the output network variables on the Digital Input/Counter functional block. Not all of the parameters are configurable for every processing option.

Offset	This parameter is only used if the `Frequency` processing option is being used. This value is added to the sensed reading prior to sending the reading on

	the Analog network variable. The offset may be positive or negative. The units match the units being sent on the Analog network variable. The offset is applied after the translation.
Minimum Value	This parameter is only used if the `Frequency` processing option is being used. This value determines the minimum value that will be sent by the Analog network variable. If this limit is exceeded the output will be clipped at this limit.
Maximum Value	This parameter is only used if the `Frequency` processing option is being used. This value determines the maximum value that will be sent by the Analog network variable. If this limit is exceeded the output will be clipped at this limit.
Send on Delta	This parameter is only used if the `Count`, `Repeating Count`, `On Time`, or `Frequency` processing option is being used. The sensed value must change by at least this amount before a new value is sent. A new value will still be sent if the heartbeat time expires.
Analog Override	This parameter is only used if the `Count`, `Repeating Count`, `On Time`, or `Frequency` processing option is being used. This value determines the value that will be sent by the Analog network variable if this functional block is put into override.
Digital Override	This parameter is only used if any processing option other than `Frequency` is being used. This value determines the value that will be sent by the Digital network variable if this functional block is put into override.
Heartbeat	Determines how often the functional block sends a heartbeat over the network over each of its output network variables. The behavior of the system in case of a missed heartbeat is determined by the functional blocks which fail to receive the heartbeat. Setting this property to 0 disables the heartbeat for this functional block. Disabling the heartbeat causes the output network variable to only be transmitted in response to a changed hardware input value.
Throttle	Limits how often data is sent over the network. Setting this property to 0 disables throttling.

Status

This tab allows you to view and change the status of a Digital Input/Counter functional block. See *Status* in Chapter 2 for more information.

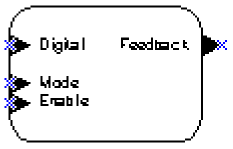
4

The Digital Output Functional Block: Application and Plug-in

This chapter describes how to configure a Digital Output functional block using the LonPoint Plug-in.

The Digital Output Functional Block

The Digital Output functional block controls the state of a digital actuator based on the value of an input network variable. The following figure and tables summarize the inputs and outputs of the Digital Output functional block:



DO- 1

Input Network Variables

Default name	Default type	Description
Digital	`SNVT_switch`	The Digital input network variable that drives the actuator.
Mode	`SNVT_hvac_mode` (changeable)	The Mode input network variable. See the `Presets` tab for more information.
Enable	`SNVT_switch`	The Enable input network variable. If this network variable is set to On, the functional block will function normally. If this network variable is set to Off, the actuator driven by this functional block will return to its default state.

Output Network Variables

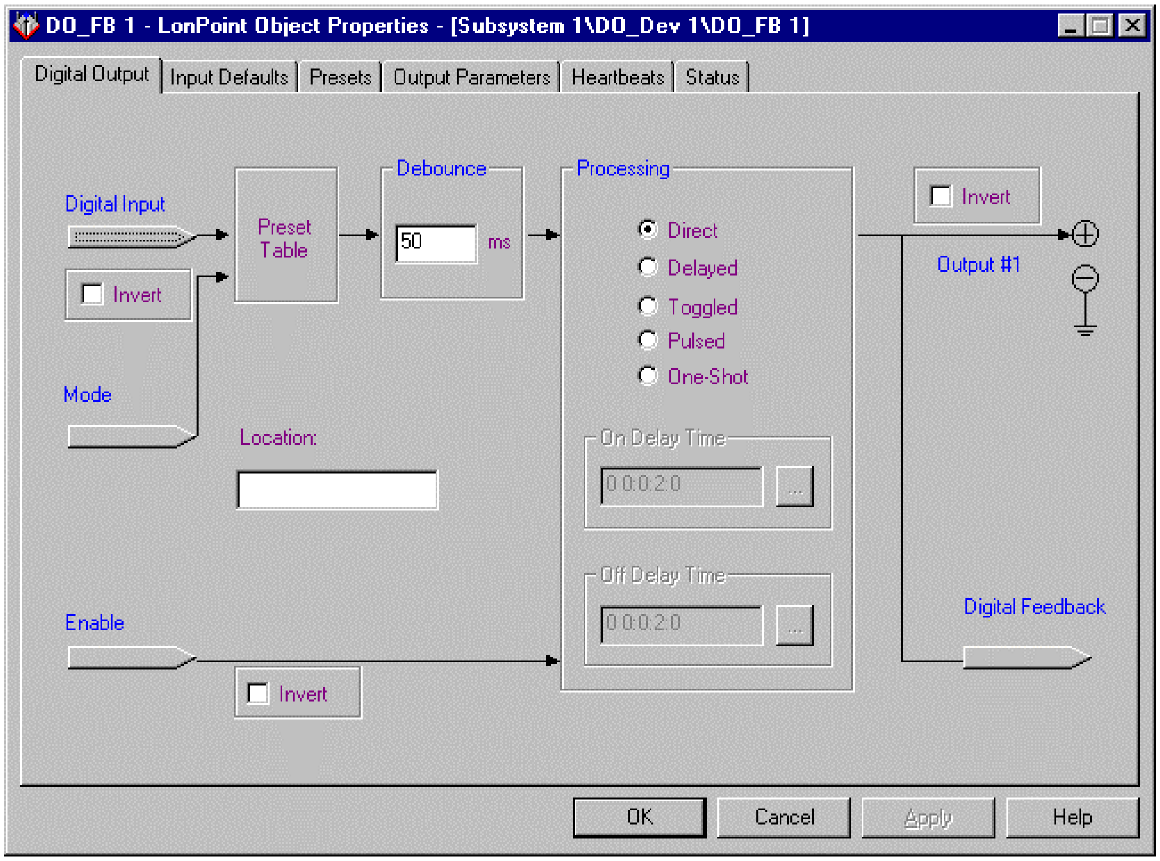
Default name	Default type	Description
Feedback	SNVT_switch	The output value controlling the actuator driven by this functional block.

Configuring a Digital Output with the LonPoint Plug-in

Right-click a Digital Output functional block and select *Configure* from the shortcut menu to open the LonPoint Plug-in. The Digital Output window of the LonPoint Plug-in contains the following tabs: `Digital Output`, `Input Defaults`, `Presets`, `Output Parameters`, `Heartbeats`, and `Status`.

Digital Output

This tab appears as follows:



This tab displays the flow of information through the Digital Output functional block. It contains the following fields:

Location	The location string for this digital output. This property can be used to document the associated actuator's location within the plant so it can be easily found. This field may contain up to 30 characters. This value is separate from the LonPoint device's location property.
Digital Input Invert	Specifies whether or not the value of the `Digital` network variable is inverted before it is processed.
Enable Invert	Specifies whether or not the value of the `Enable` network variable is inverted before it is processed. If not selected, an On value enables the output.
Output Invert	Specifies whether or not the value of the digital output value is inverted before it is sent to the attached hardware. This option does not apply to the `Feedback` output.

Debounce

Determines how long the `Digital` input network variable and the `Mode` input network variable must remain stable after a transition in order for the transition to be recognized as valid. This prevents the state from fluttering. This configuration property may be set between 0 and 999 ms. A value of 0 turns the debounce function off.

Processing

Determines how the debounced input data is translated into the value used to drive the actuator. This configuration property can be set to one of the following options: `Direct`, `Delayed`, `Toggled`, `Pulsed`, or `One-Shot`. The `Delayed`, `Pulsed` and `One-Shot` options use the time fields below. If the `Delayed` option is selected, these fields will be labeled `On Delay Time` and `Off Delay Time`. If the `Pulsed` or `One-Shot` option is selected, these fields will be labeled `On Delay Time` and `Pulse Time`.

The `Direct` option causes data to be output directly after the debounce function has been executed.

The `Delayed` option causes a change from Off to On to be delayed by the time indicated in the `On Delay Time` field and a change from On to off to be delayed by the time indicated in the `Off Delay Time` field.

The `Toggled` option causes the output data to toggle every time the input data changes from Off to On.

Use the `Pulsed` option to generate a pulse on the output network variable every time the input data changes from Off to On. The pulse is generated after a delay specified on the `On Delay Time` field. The duration of the pulse is specified on the `Pulse Time` field. It does not matter how long the input data remains on, the output will always send a pulse of the specified length after waiting the configured delay.

Use the `One-Shot` option to generate a pulse on the output network variable every time the input data changes from Off to On. The pulse is generated after a delay specified on the `On Delay Time` field. The duration of the pulse is specified on the `Pulse Time` field. If the input data is changed from Off to On while the pulse is being sent, the pulse timer will be reset (i.e., if a two second pulse was retriggered after one second, the output would be on for two more seconds, or three seconds total). If a delayed pulse is re-triggered during a pulse, the delay will be ignored. If a pulse is re-triggered during a delay, the trigger will be ignored.

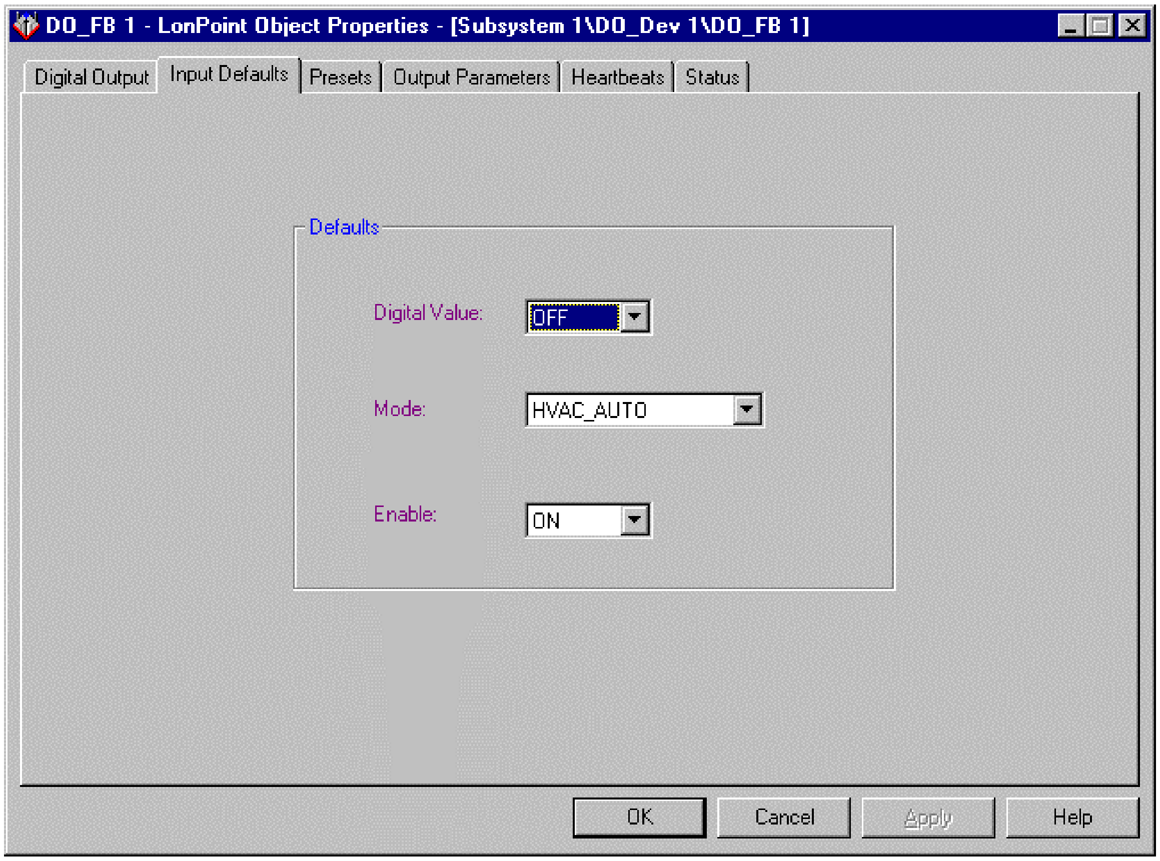
On Delay Time

Use this field with the `Delayed`, `Pulsed`, and `One-Shot` processing options. This value determines the length of the delay in the change from the Off state to the On state. To change the value, click the button to the right of the time value to be changed and enter the new values to be used for the delay. The valid range for this value is from zero to 48 days, 23 hours, 59 minutes, and 999 milliseconds.

Off Delay Time	Use this field with the `Delayed Processing` option. This value determines the length of the delay in the change from the On state to the Off state. To change the value, click the button to the right of the time value to be changed and enter the new millisecond, second, and minute values to be used for the delay. The valid range for this value is from zero to 48 days, 23 hours, 59 minutes, 59 seconds, and 999 milliseconds.
Pulse Time	If you select either the `Pulsed` or the `One-Shot` options, the `Pulse Time` field appears in place of `Off Delay Time`. This value determines the length of time the On state remains active. To change the value, click the button to the right of the time value to be changed and enter the new millisecond, second, and minute values to be used for the delay. The valid range for this value is from zero to 48 days, 23 hours, 59 minutes, 59 seconds, and 999 milliseconds.

Input Defaults

This tab appears as follows:



Defaults

Digital Value: OFF

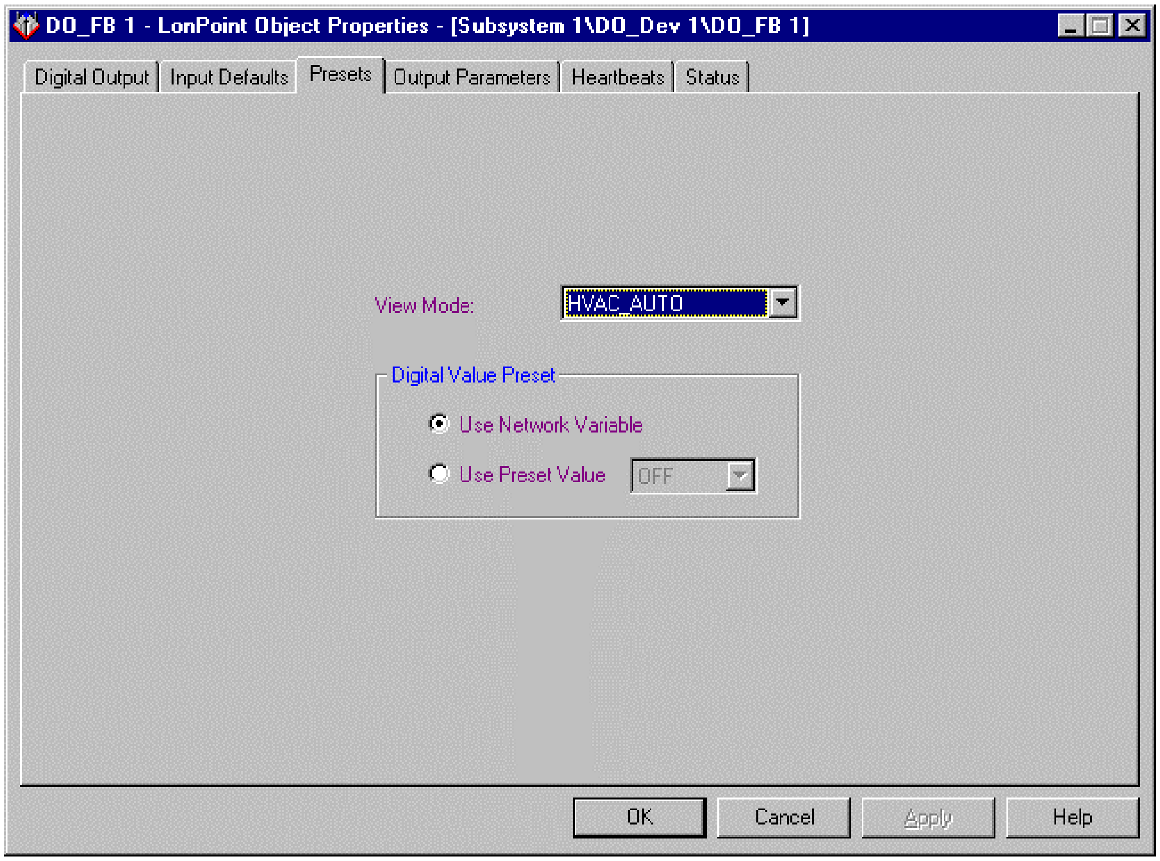
Mode: HVAC_AUTO

Enable: ON

This tab allows you to set the default inputs for the `Digital Value`, `Mode`, and `Enable` input network variables. The specified input values will be used if the associated network variable is not connected or has not received an update since the device's last reset.

Presets

This tab appears as follows:



View Mode: HVAC_AUTO

Digital Value Preset

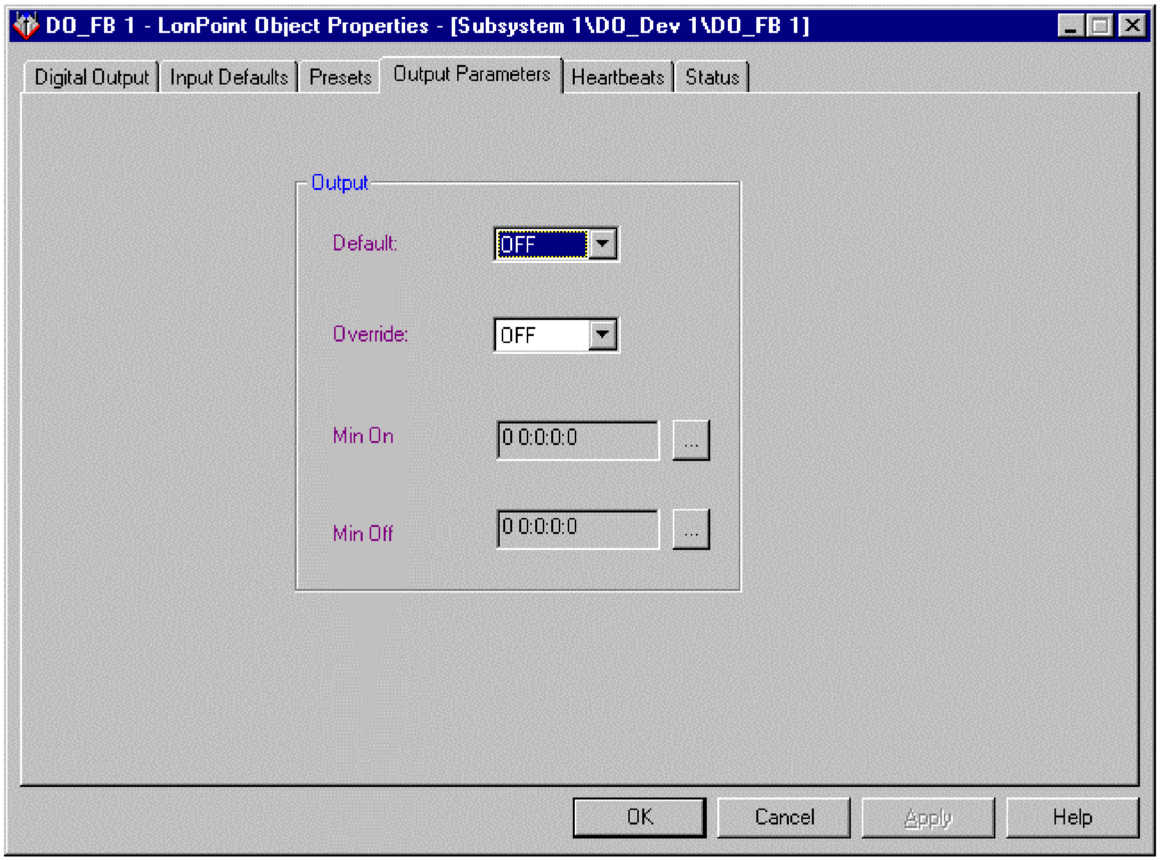
● Use Network Variable

○ Use Preset Value OFF

This tab allows you to set up the preset table which determines how the digital output behaves for each mode value. For each value that the `Mode` network variable can receive, you can configure whether the output value will be taken from the `Digital` network variable or will be set to a preset value for the selected mode. Select `Use Network Variable` to use the network variable value. Select `Use Preset Value` to use the specified preset value.

Output Parameters

This tab appears as follows:


DO_FB 1 - LonPoint Object Properties - [Subsystem 1\DO_Dev 1\DO_FB 1]

Default: OFF

Override: OFF

Min On 0 0:0:0:0

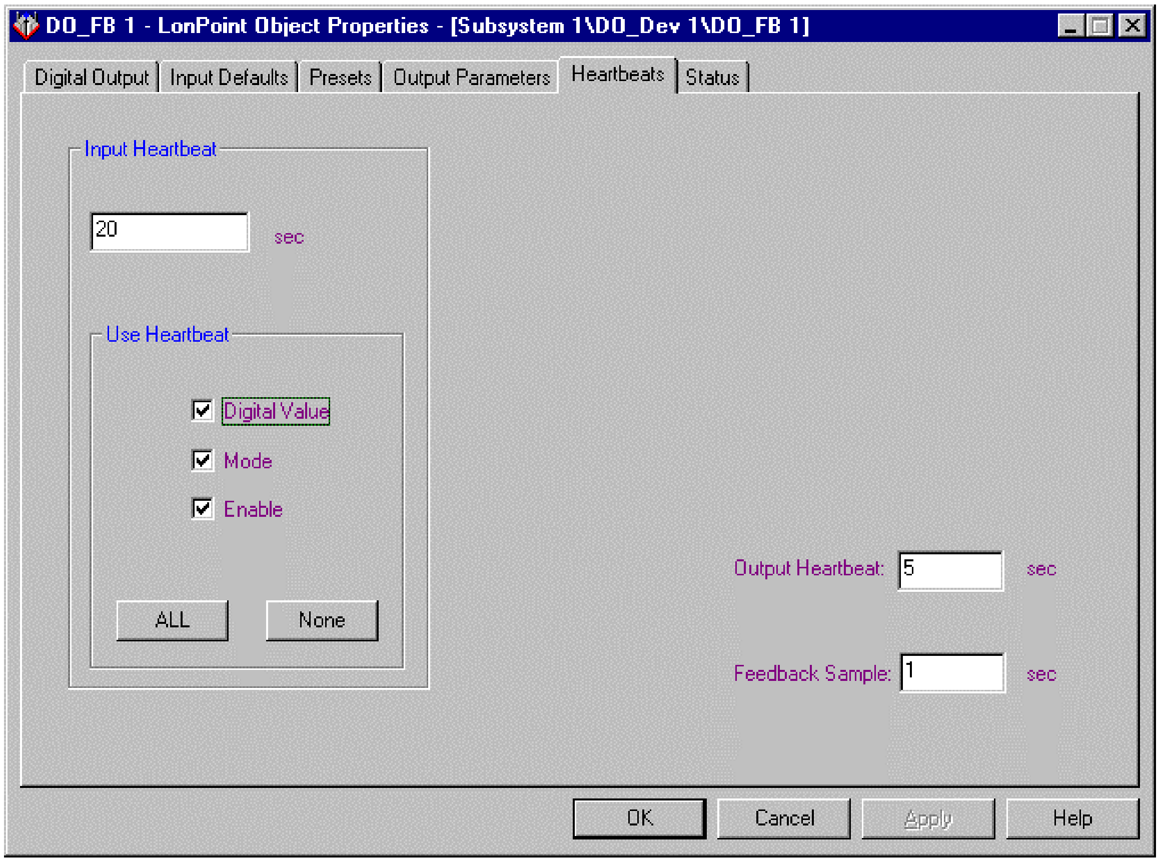
Min Off 0 0:0:0:0

This tab allows you to set options determining what information is sent to the output. This tab contains the following fields:

Default Value	Determines the value that will be sent to the hardware if the block is disabled or has detected a heartbeat failure.
Override Value	Indicates the output value that will be used if the functional block is put into override. This value will be inverted if the *Output Invert* option is selected.
Min On/Min Off	Indicates the minimum amount of time an On to Off condition will be sent to the hardware digital output and the `Feedback` network variable. If the Pulsed or One-shot processing option is being used, there will always be a delay of at least the *Min Off* time between pulses. To modify the value in either of these fields, click the associated button and enter appropriate values in the `Day`, `Hour`, `Minute`, `Second`, and `Millisecond` fields.

Heartbeats

This tab appears as follows:



Input Heartbeat
20 sec

Use Heartbeat
☑ Digital Value
☑ Mode
☑ Enable

Output Heartbeat: 5 sec

Feedback Sample: 1 sec

This tab allows you to set properties that determine the input and output heartbeat rates and feedback sampling. This tab contains the following fields:

Input Heartbeat	This property determines the interval (in seconds) the network variables that have the *Use Heartbeat* option selected will wait for a heartbeat before detecting a heartbeat failure. See *Heartbeats*, in Chapter 1, for more information.
Use Heartbeat	Specifies whether or not heartbeat checking will be used for the selected input network variable. If checked for a given network variable, it indicates that heartbeat checking will be used. The maximum time between updates for network variables with heartbeat checking enabled is set in the Input Heartbeat field. Click the ALL or None button to turn heartbeat checking On or Off for all input network variables, respectively.
Output Heartbeat	Determines the heartbeat send rate, in seconds, for the Feedback output network variable.
Feedback Sample	Determines how often the hardware digital output is sampled for the purpose of updating the Feedback

output network variable. This signal can come from a network variable or from the Hand-Off-Auto switch. The signal being sent to the hardware digital output is sampled at this rate, and the network variable is updated whenever the value changes or when determined by the heartbeat.

Status

This tab allows you to view and change the status of the Digital Output functional block. See *Status* in Chapter 2 for more information.

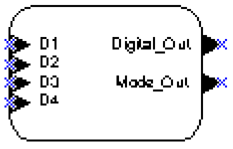
5

The Digital Encoder Functional Block: Application and Plug-in

This chapter describes how to configure a Digital Encoder functional block using the LonPoint Plug-in.

The Digital Encoder Functional Block

The Digital Encoder functional block performs logic functions on four digital input network variables to create a digital output network variable and a mode output network variable. The following figure and tables summarize the inputs and outputs of the Digital Encoder functional block:



DE- 1

Input Network Variables

Default name	Default type	Description
D1/D2/D3/D4	`SNVT_switch`	Four digital input network variables which are processed by the Digital Encoder.

Output Network Variables

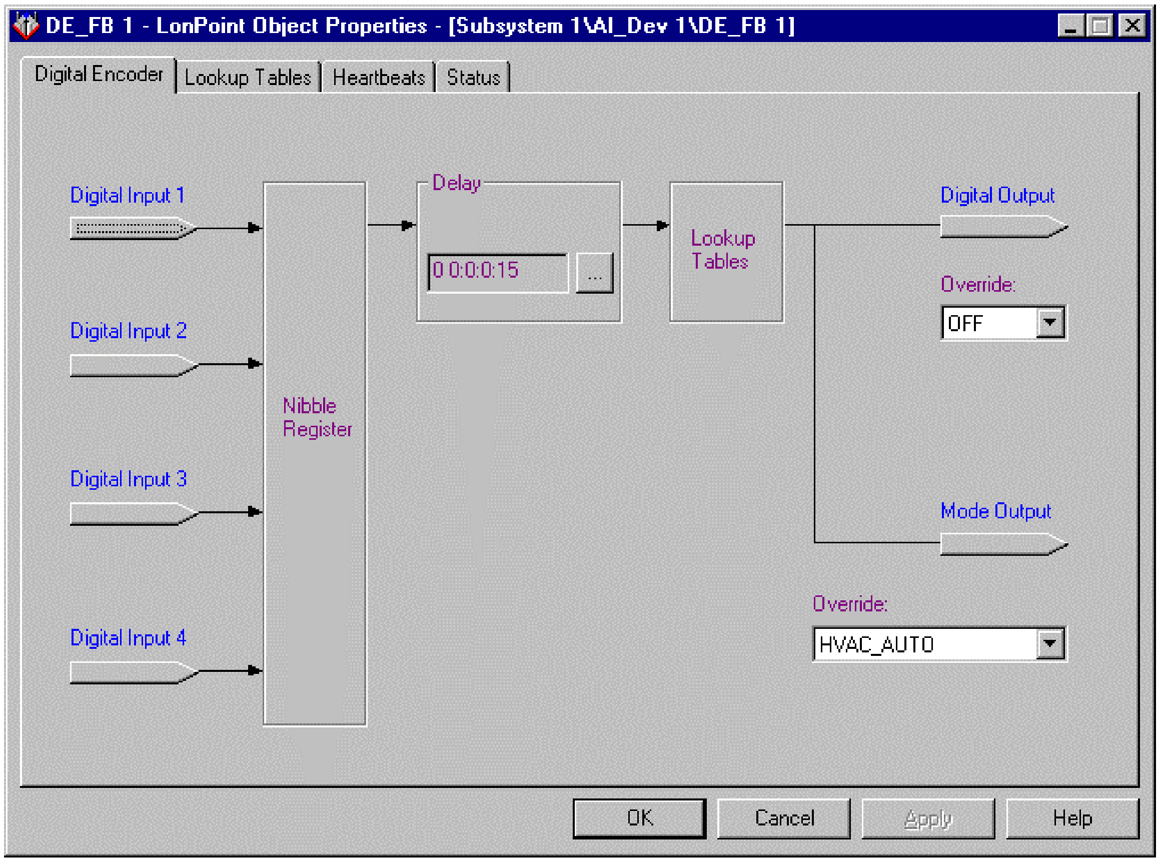
Default name	Default type	Description
Digital_Out	`SNVT_switch`	The digital output network variable determined by the Digital Encoder.
Mode_Out	`SNVT_hvac_mode` (changeable)	The mode output network variable determined by the Digital Encoder.

Configuring the Digital Encoder with the LonPoint Plug-in

Right-click a Digital Encoder functional block and select *Configure* from the shortcut menu to open the LonPoint Plug-in. The Digital Encoder window of the LonPoint Plug-in contains the following tabs: `Digital Encoder`, `Lookup Tables`, `Heartbeats`, and `Status`.

Digital Encoder

This tab appears as follows:


DE_FB 1 - LonPoint Object Properties - [Subsystem 1\AI_Dev 1\DE_FB 1]

Delay: 0 0:0:0:15

Override: OFF

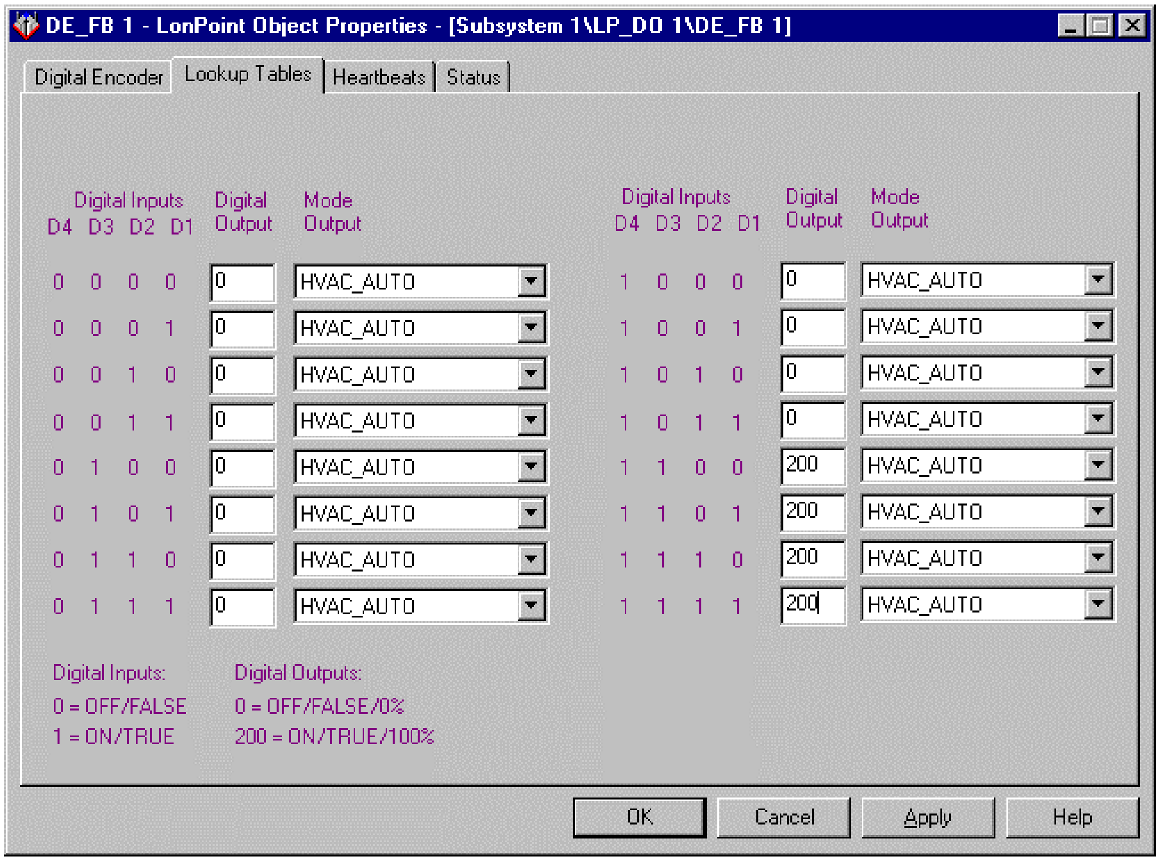
Override: HVAC_AUTO

This tab displays the flow of data through the Digital Encoder functional block. This tab contains the following fields:

Delay	Specifies the amount of time the digital inputs must stay constant before they are processed. The value must be between zero and 48 days, 23 hours, 59 minutes, 59 seconds, and 999 milliseconds. To change the value, click the button to the right of the time value to be changed and enter the new values to be used for the delay. Set this field to zero to disable the delay function.
Digital Output Override	Selects the value that will be sent on the `Digital_Out` network variable when the functional block is put into override.
Mode Output Override	Selects the value that will be sent on the `Mode_Out` network variable when the functional block is put into override.

Lookup Tables

This tab appears as follows:



DE_FB 1 - LonPoint Object Properties - [Subsystem 1\LP_DO 1\DE_FB 1]
Digital Encoder
Lookup Tables
Heartbeats
Status
OK
Cancel
Apply
Help

Digital Inputs D4	D3	D2	D1	Digital Output	Mode Output	Digital Inputs D4	D3	D2	D1	Digital Output	Mode Output
0	0	0	0	0	HVAC_AUTO	1	0	0	0	0	HVAC_AUTO
0	0	0	1	0	HVAC_AUTO	1	0	0	1	0	HVAC_AUTO
0	0	1	0	0	HVAC_AUTO	1	0	1	0	0	HVAC_AUTO
0	0	1	1	0	HVAC_AUTO	1	0	1	1	0	HVAC_AUTO
0	1	0	0	0	HVAC_AUTO	1	1	0	0	200	HVAC_AUTO
0	1	0	1	0	HVAC_AUTO	1	1	0	1	200	HVAC_AUTO
0	1	1	0	0	HVAC_AUTO	1	1	1	0	200	HVAC_AUTO
0	1	1	1	0	HVAC_AUTO	1	1	1	1	200	HVAC_AUTO

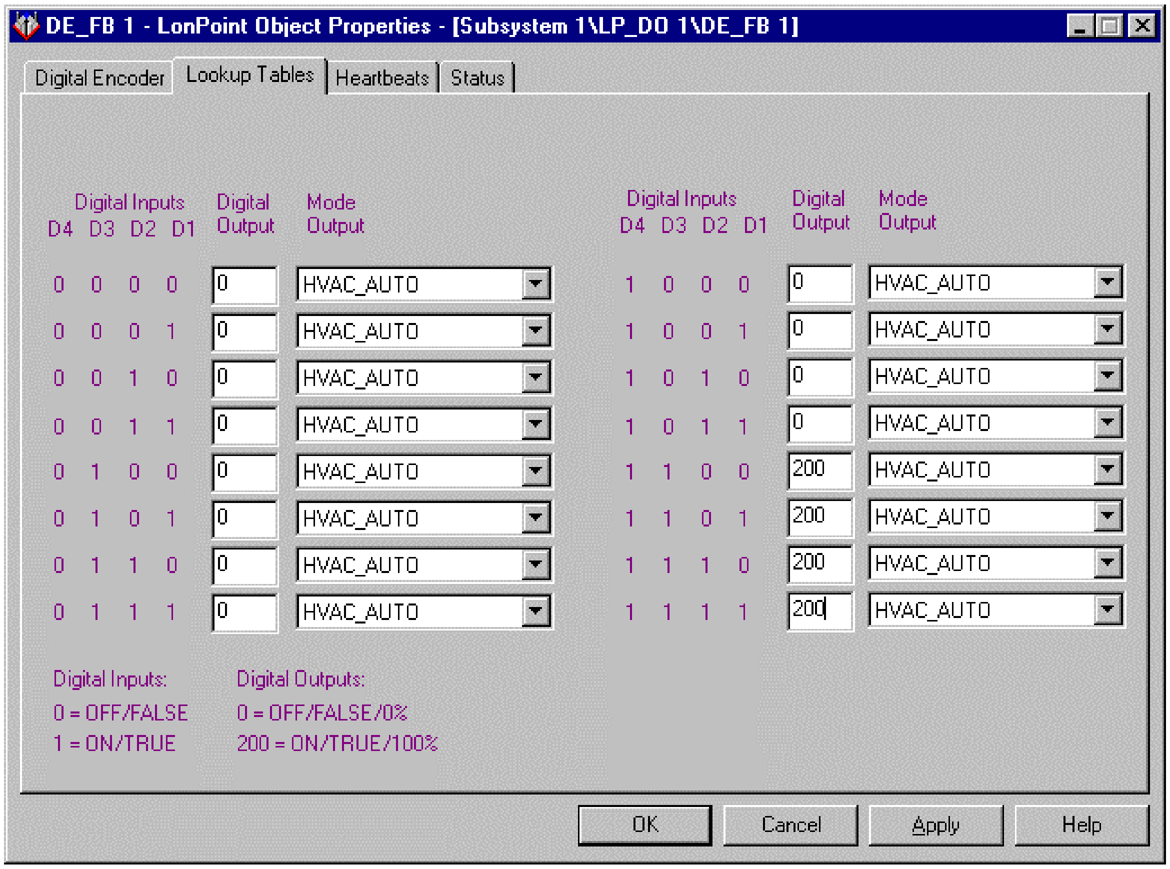
Digital Inputs:	Digital Outputs:
0 = OFF/FALSE	0 = OFF/FALSE/0%
1 = ON/TRUE	200 = ON/TRUE/100%

This tab contains the lookup table that determines the output of the `Digital_Out` and `Mode_Out` network variables based on the values of the 4 digital inputs. Each row of the table corresponds to one of the 16 possible combinations of input values from the four digital inputs.

For each combination of inputs, you can set an output value for the `Digital_Out` network variable in the `Digital Output` column and the `Mode_Out` network variable in the `Mode Output` column. The fields in the `Digital Output` column can be set to 0 to send an Off digital state, 200 to send an On digital state, or a value between 0 and 200 to send an On value with a level less than 100% (the level is one half the specified value).

The available values in the `Mode Output` column depend on the type of the `Mode_Out` network variable selected on the `Digital Encoder` tab.

For example, to create a digital encoder block that returns an On value on the Digital Output when the D3 AND D4 inputs are on and on Off value otherwise, enter a value of 0 in the `Digital Output` column in the first 12 rows (in which either or both of the D3 and D4 inputs are 0) and a value of 200 in the `Digital Output` column in the remaining rows (in which both D3 and D4 have a value of 1), as follows:

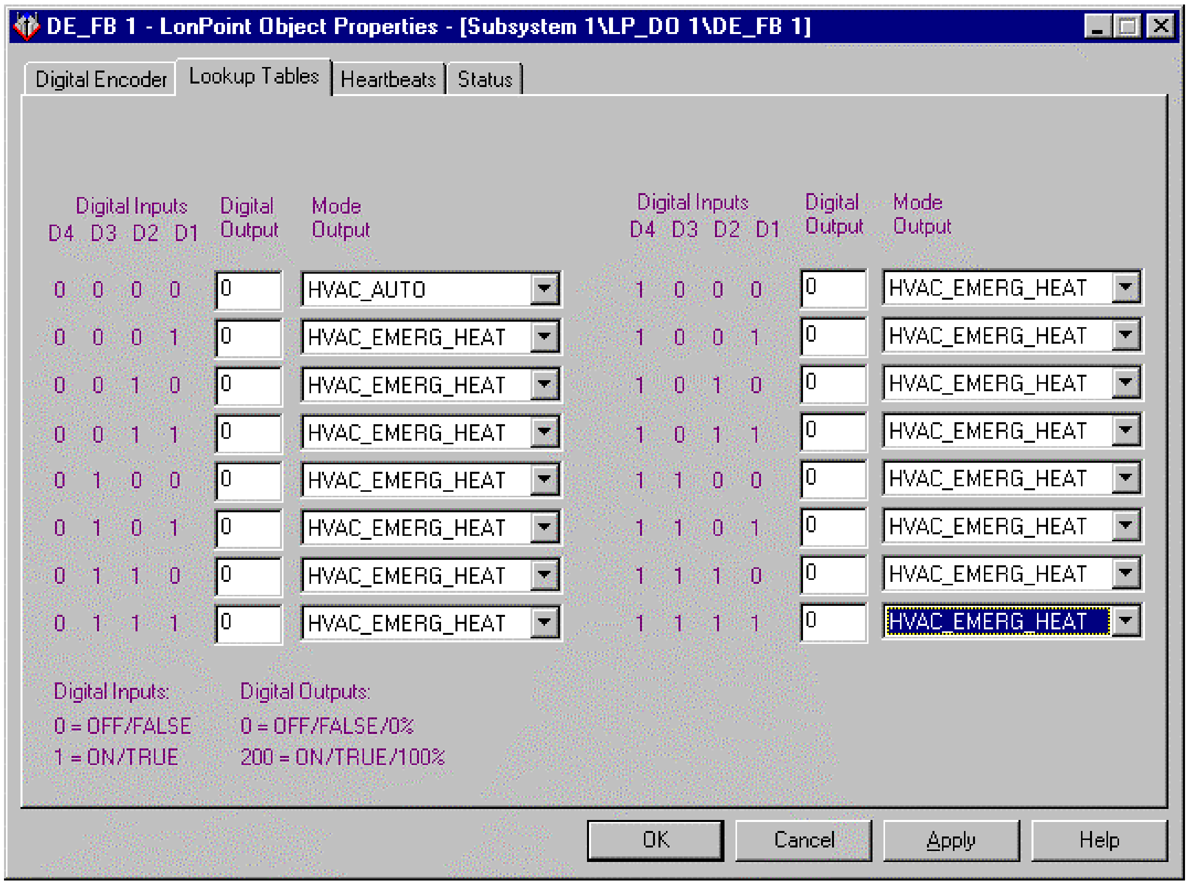

DE_FB 1 - LonPoint Object Properties - [Subsystem 1\LP_DO 1\DE_FB 1]
Digital Encoder
Lookup Tables
Heartbeats
Status
OK
Cancel
Apply
Help

Digital Inputs D4	D3	D2	D1	Digital Output	Mode Output	Digital Inputs D4	D3	D2	D1	Digital Output	Mode Output
0	0	0	0	0	HVAC_AUTO	1	0	0	0	0	HVAC_AUTO
0	0	0	1	0	HVAC_AUTO	1	0	0	1	0	HVAC_AUTO
0	0	1	0	0	HVAC_AUTO	1	0	1	0	0	HVAC_AUTO
0	0	1	1	0	HVAC_AUTO	1	0	1	1	0	HVAC_AUTO
0	1	0	0	0	HVAC_AUTO	1	1	0	0	200	HVAC_AUTO
0	1	0	1	0	HVAC_AUTO	1	1	0	1	200	HVAC_AUTO
0	1	1	0	0	HVAC_AUTO	1	1	1	0	200	HVAC_AUTO
0	1	1	1	0	HVAC_AUTO	1	1	1	1	200	HVAC_AUTO

Digital Inputs:
0 = OFF/FALSE
1 = ON/TRUE

Digital Outputs:
0 = OFF/FALSE/0%
200 = ON/TRUE/100%

To create a table that sends an HVAC_EMERG_HEAT mode on the `Mode_Out` network variable if any of the four inputs are On and HVAC_AUTO if they are all Off, you would set the `Mode Output` column to HVAC_EMERG_HEAT in all the rows but the first (in which at least one of the four inputs has a value of 1) and to HVAC_AUTO in the first row (where all four inputs are 0), as follows:


DE_FB 1 - LonPoint Object Properties - [Subsystem 1\LP_DO 1\DE_FB 1]
Digital Encoder
Lookup Tables
Heartbeats
Status
OK
Cancel
Apply
Help

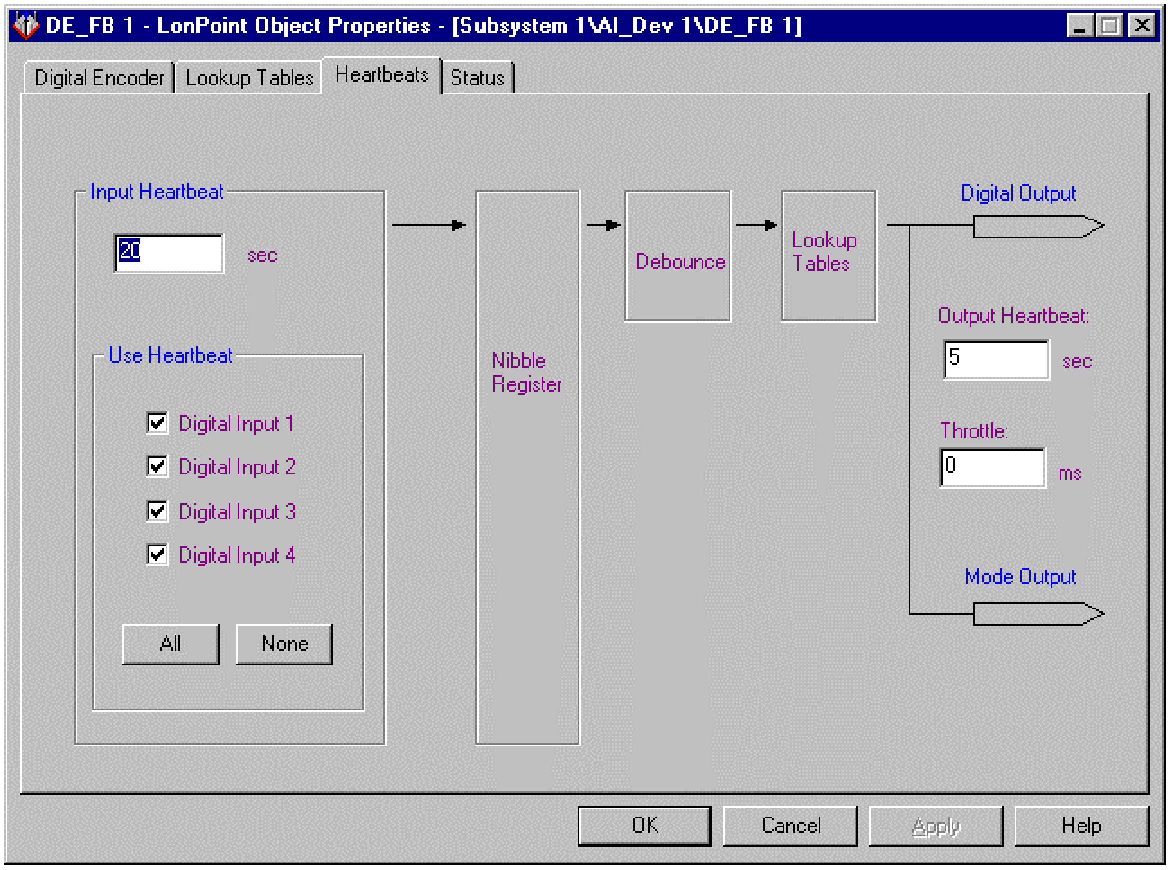
Digital Inputs D4	D3	D2	D1	Digital Output	Mode Output	Digital Inputs D4	D3	D2	D1	Digital Output	Mode Output
0	0	0	0	0	HVAC_AUTO	1	0	0	0	0	HVAC_EMERG_HEAT
0	0	0	1	0	HVAC_EMERG_HEAT	1	0	0	1	0	HVAC_EMERG_HEAT
0	0	1	0	0	HVAC_EMERG_HEAT	1	0	1	0	0	HVAC_EMERG_HEAT
0	0	1	1	0	HVAC_EMERG_HEAT	1	0	1	1	0	HVAC_EMERG_HEAT
0	1	0	0	0	HVAC_EMERG_HEAT	1	1	0	0	0	HVAC_EMERG_HEAT
0	1	0	1	0	HVAC_EMERG_HEAT	1	1	0	1	0	HVAC_EMERG_HEAT
0	1	1	0	0	HVAC_EMERG_HEAT	1	1	1	0	0	HVAC_EMERG_HEAT
0	1	1	1	0	HVAC_EMERG_HEAT	1	1	1	1	0	HVAC_EMERG_HEAT

Digital Inputs:
0 = OFF/FALSE
1 = ON/TRUE

Digital Outputs:
0 = OFF/FALSE/0%
200 = ON/TRUE/100%

Heartbeats

This tab appears as follows:


DE_FB 1 - LonPoint Object Properties - [Subsystem 1\AI_Dev 1\DE_FB 1]

Input Heartbeat: 20 sec

Use Heartbeat:
- [✓] Digital Input 1
- [✓] Digital Input 2
- [✓] Digital Input 3
- [✓] Digital Input 4

Output Heartbeat: 5 sec

Throttle: 0 ms

This tab allows you to set heartbeat and throttle options for the input and output network variables on the Digital Encoder functional block. This tab contains the following fields:

Input Heartbeat	Determines the interval (in seconds) the network variables that have the `Use Heartbeat` option selected will wait for a heartbeat before registering a heartbeat failure. See *Heartbeats* in Chapter 1 for more information
Use Heartbeat	Determines whether heartbeat checking will be used for each of the four input network variables. The `All` and `None` buttons allow you to enable heartbeat checking for all input network variables or none of the input network variables, respectively.
Output Heartbeat	Determines the output heartbeat, in seconds, for both output network variables. This is the maximum amount of time that may pass without a network variable update before one is sent automatically. Set

	this field to zero to disable heartbeats for both output network variables.
Throttle	Determines the throttle, in milliseconds, for both output network variables. This is the minimum amount of time between two network variable updates. Set this field to zero to disable throttling for both output network variables.

Status

This tab allows you to view and change the status of a Digital Encoder functional block. See *Status* in Chapter 2 for more information.

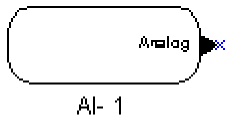
6

The Analog Input Functional Block: Application and Plug-in

This chapter describes how to configure an Analog Input functional block using the LonPoint Plug-in.

The Analog Input Functional Block

The Analog Input functional block measures the voltage, current, or resistance of a hardware transducer. This value is then translated into an appropriate value to send on the `Analog` output network variable. For example, a temperature transducer might return a value of 10,000 to 15,000 ohms. The Analog Input functional block translates this raw value into the appropriate temperature value using a translation table. The following figure and table summarizes the output of the Analog Input functional block:



Output Network Variables

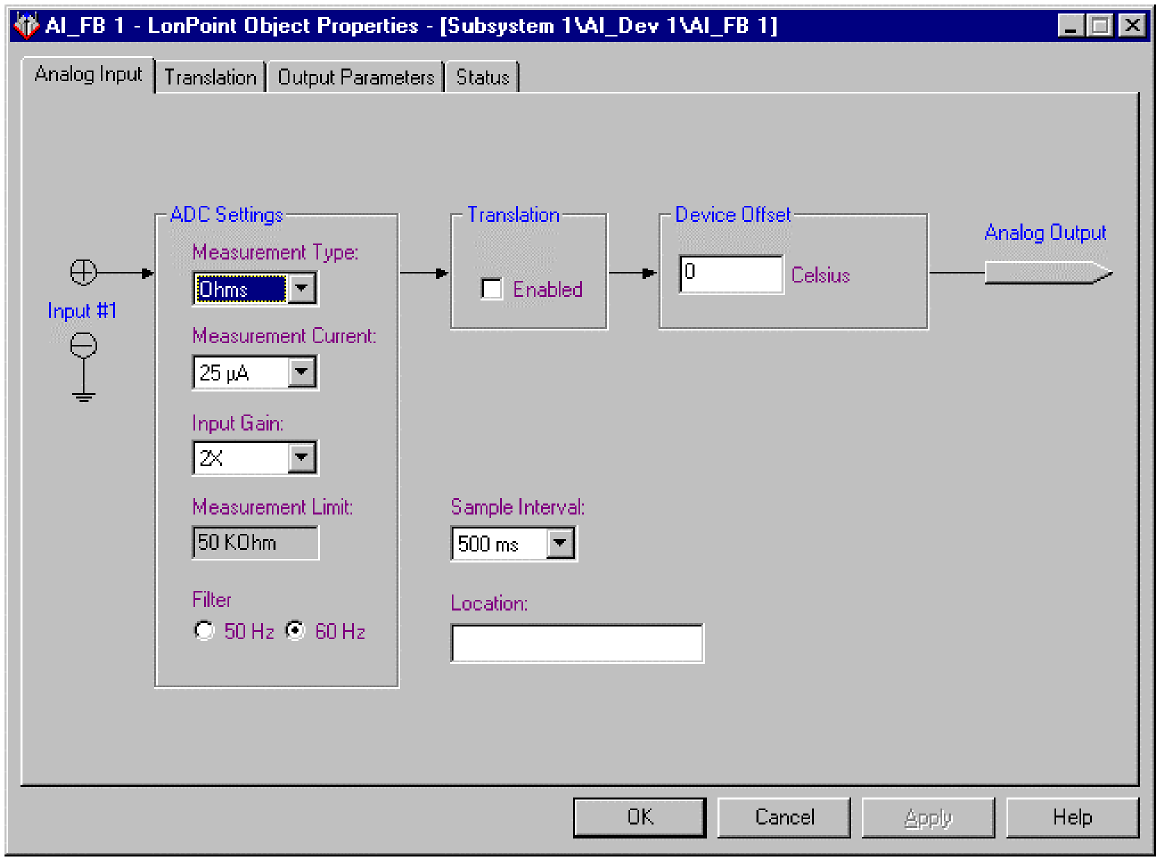
Default name	**Default type**	**Description**
Analog	`SNVT_temp_f` (changeable)	The Analog output network variable driven by the sensor.

Configuring an Analog Input with the LonPoint Plug-in

Right-click an Analog Input functional block and select *Configure* from the shortcut menu to open the LonPoint Plug-in. The Analog Input window of the LonPoint Plug-in contains the following tabs: `Analog Input`, `Translation`, `Output Parameters`, and `Status`.

Analog Input

This tab appears as follows:


AI_FB 1 - LonPoint Object Properties - [Subsystem 1\AI_Dev 1\AI_FB 1]

Analog Input | Translation | Output Parameters | Status

ADC Settings
Measurement Type: Ohms
Measurement Current: 25 µA
Input Gain: 2X
Measurement Limit: 50 KOhm
Filter: ○ 50 Hz ● 60 Hz

Translation: ☐ Enabled

Device Offset: 0 Celsius

Sample Interval: 500 ms

Location:

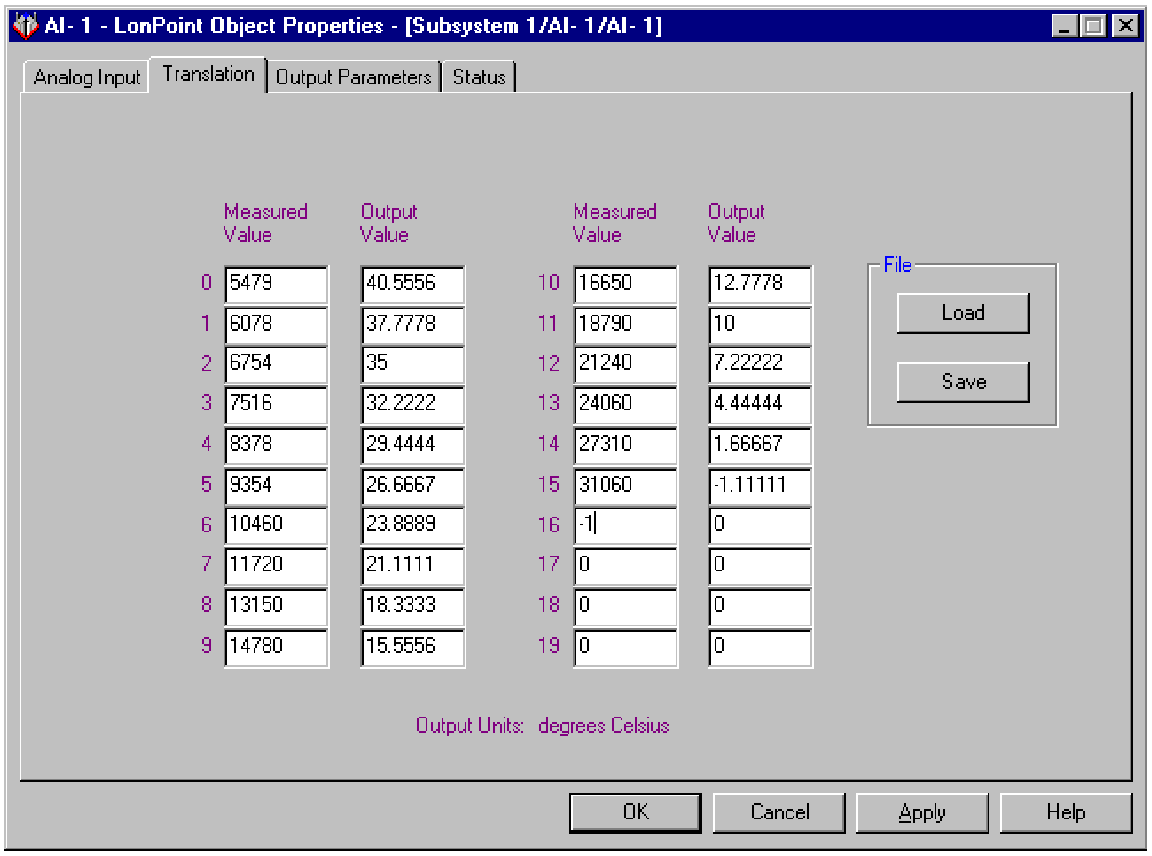
This tab shows the flow of information through the Analog Input functional block. It contains the following fields:

Measurement Type	Determines whether the input reads voltage (Volts), current (Amps), or resistance (Ohms). The value in this field must match the hardware settings on the associated LonPoint module. See the *LonPoint Hardware and Installation Guide* for information on changing the hardware settings.
Measurement Current	If the *Measurement Type* field is set to Ohms, this field determines whether a current of 25 µA or 400 µA is used in measuring resistance. If the *Measurement Type* field is set to Volts or Amps, this field is not used.
Input Gain	Determines the gain for the input. See *Analog Input Gain and Current Settings* later in this chapter for more information on gain settings.
Measurement Limit	Displays a calculation of the highest value that may be read from the transducer, as determined by the

	Measurement Type, *Measurement Current*, and *Input Gain* values.
Filter	Sets the input noise filtering on the device to reject either 50 Hz or 60 Hz noise.
Translation Enabled	Determines whether the translation table, set up in the `Translation` tab, will be used. If this option is disabled, the untranslated, or raw, values will be used as read from the transducer.
Device Offset	The value set in this field is added to the sensed reading to account for device-to-device transducer variations. The units match the units being sent on the network variable. This offset is applied after the translation. For example, if a 100 Ω RTD measures 0.12° C at the ice-point, the offset should be –0.12.
Sample Interval	Controls the rate that the analog input samples the input hardware and updates its output network variable. This field may be set to 500 ms, 1 sec, 2 sec, 4 sec, 8 sec, 16 sec, or 32 sec.
Location	The location string for this analog input. This property can be used to document the associated input's location within the plant so it can be easily found. This field may contain up to 30 characters. This value is separate from the device's location string.

Translation

This tab appears as follows:



AI- 1 - LonPoint Object Properties - [Subsystem 1/AI- 1/AI- 1]

Analog Input | Translation | Output Parameters | Status

	Measured Value	Output Value		Measured Value	Output Value
0	5479	40.5556	10	16650	12.7778
1	6078	37.7778	11	18790	10
2	6754	35	12	21240	7.22222
3	7516	32.2222	13	24060	4.44444
4	8378	29.4444	14	27310	1.66667
5	9354	26.6667	15	31060	-1.11111
6	10460	23.8889	16	-1	0
7	11720	21.1111	17	0	0
8	13150	18.3333	18	0	0
9	14780	15.5556	19	0	0

Output Units: degrees Celsius

This tab allows you to set up a translation table that translates the raw measured data into the values to be sent over the Analog output network variable.

The translation table provides a means for mapping any input set of values into a different set of output values via pairs of input/output values. The Analog Input software automatically linearly interpolates to determine translation values that lie between defined input/output pairs.

The table must contain at least two input/output pairs and may contain as many as twenty.

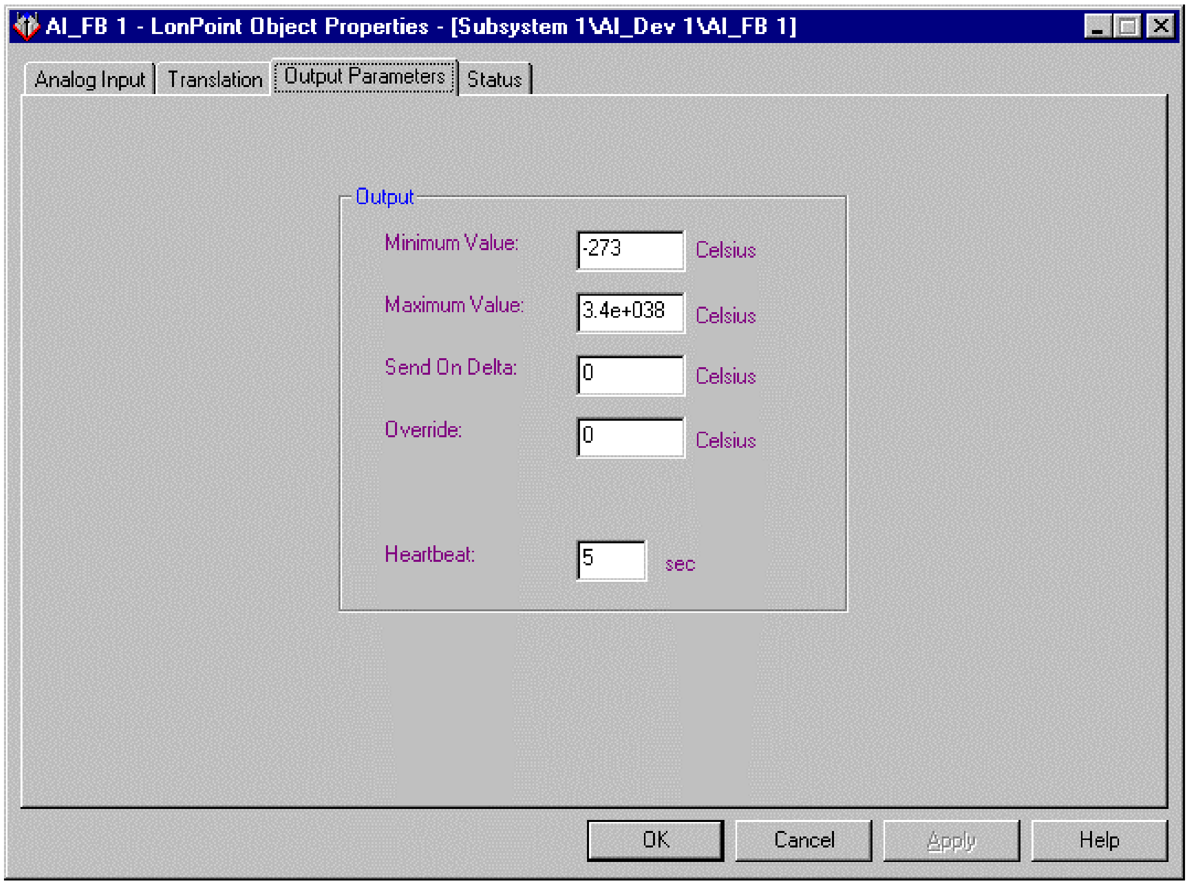
The only constraint on the Measured Value column is that the entries increase monotonically (i.e. each value in the column must be greater than the previous value). To mark the end of the table, enter a measured value less than the preceding value.

Click the Save button to save translation table data into a tab-delimited text file. Click the Load button to load data from a tab-delimited text file into a translation table. Use these functions to save and reuse commonly used translation tables for your hardware inputs. You can also use custom LonMaker shapes to save frequently used configurations.

All fields on this tab will be disabled if the *Translation Enabled* option is not set on the `Analog Input` tab.

Output Parameters

This tab appears as follows:


AI_FB 1 - LonPoint Object Properties - [Subsystem 1\AI_Dev 1\AI_FB 1]

Output

Minimum Value:	-273	Celsius
Maximum Value:	3.4e+038	Celsius
Send On Delta:	0	Celsius
Override:	0	Celsius
Heartbeat:	5	sec

This tab determines how often data is sent on the `Analog` network variable and the minimum, maximum, and override values. This tab contains the following fields:

Minimum Value	The minimum value that will be sent by the output network variable. If this limit is exceeded the output will be clipped at this limit.
Maximum Value	The maximum value that will be sent by the output network variable. If this limit is exceeded the output will be clipped at this limit.
Send on Delta	The sensed value must change by at least this amount before a new value is sent. A new value will still be sent if the heartbeat time expires.
Override	The value that will be sent by the `Analog` output network variable if this functional block is put into override.

Heartbeat	Determines how often the functional block sends a heartbeat over the network. The behavior of the system in case of a failed heartbeat is determined by the functional blocks which fail to receive the heartbeat. Setting this property to 0 disables the heartbeat for this functional block. Disabling the heartbeat causes the output network variable to only be transmitted in response to a changed hardware input value.

Status

This tab allows you to view and change the status of an Analog Input functional block. See *Status* in Chapter 2 for more information.

Analog Input Gain and Current Settings

The analog-to-digital (A/D) converters on the LonPoint module must be properly configured to match the `ADC Settings` set in the `Analog Input` tab of the plug-in. These can be set via hardware jumpers as described in the *LonPoint Hardware and Installation Guide.*

The A/D converters on the module can be programmed for four different input gains: 1, 2, 32, and 128. The gain settings affect the Ohms, Volts, and Amps mode settings in different ways, and each will be discussed separately. In addition, the measurement current for the Ohms setting can be set to one of two different values.

Amps Measurement Type (2WIR and 4WIR Jumper Settings)

As a rule, the A/D gain should be set to the maximum allowable level so as not to create an overflow condition. This allows the maximum possible resolution. Based on the gain setting, the software converts each raw number reading to the correct value. For current measurement settings (the 2WIR and 4WIR modes), the gain should be set to 1 for any standard 4-20mA current loop. For example, a gain of 2 would reduce the maximum allowable current to 12.5mA, which would not be useful for a 4-20mA current loop.

There are no other software settings associated with the 2WIR or 4WIR settings.

Ohms Measurement Type (RES Jumper Setting)

The resistance measurement setting configures the LonPoint module to measure the voltage drop from a constant current source through the resistor to be measured.

There are two resistance measuring current settings: 25µA and 400µA. The 25µA setting is typically used for thermistors, while the 400µA setting is typically used for RTDs. The determination whether to use the 25µA or 400µA current setting is determined by two factors: absolute range and self-heating limitations.

The absolute range refers to the upper input voltage limit of the A/D converter after voltage input scaling: 2.500VDC (scaling allows measurement of voltages higher than this limit). Any voltage higher than this limit will not be measurable. The maximum resistance that can be measured at 25µA is 2.5V/25e-6A = 100kΩ, and at 400µA is 2.5V/400e-6A = 6.25kΩ. Resistances higher than 100kΩ at 25µA, or 6.25kΩ at 400µA, will not damage the LonPoint module but they will not be measurable.

Self-heating is another consideration that may merit the use of the lower 25µA current setting, even if the absolute limit is not a factor. Heat is generated whenever a current is passed through a resistor, but when you're trying to measure temperature this heat generation affects the accuracy of the reading. How much the accuracy is affected depends on the current, the thermal mass of the input, airflow, and so on.

The amount of self-heating power can be calculated by squaring the current and multiplying by the resistance. For example, a 2kΩ thermistor would have 1.25µW of dissipation at the 25µA setting (25µA x 25µA x 2000Ω), and 320µW of self-heating energy dissipation at the 400µA current setting (400µA x 400µA x 2000Ω). The error resulting from heat-dissipation depends on the dissipation constant of the sensor. For example, Kele PreCon thermistors have a dissipation constant of 2.7mW/°C in still air at room temperature. To calculate the measurement error, the power dissipation is divided by the dissipation constant, as follows:

Current	**Measurement Error**
25µA	1.25µAW/2700µW/dC = 0.0005°C (0.0008°F)
400µA	320µAW/2700µW/dC = 0.12°C (0.22°F)

Depending on the system, these self-heating errors may be small or large compared to the accuracy of most thermistors; however, they should be taken into consideration when selecting a current setting.

RTDs have a larger thermal mass, larger dissipation area, and low initial resistance, and are therefore not as susceptible as thermistors to significant self-heating errors.

Once the measurement current has been set, the gain should be adjusted to the maximum value that doesn't generate over-range errors. The resistor ranges for the various settings are shown in this table:

AI-10 Analog Input Module RES Settings

Gain Setting	Range @ 400µA	Ω/step @ 400µA	Range @ 25µA	Ω/step @ 25µA
x1	0 - 6.25kΩ	0.0953	0 - 88kΩ	1.525
x2	0 - 3.12kΩ	0.0477	0 - 50kΩ	0.763
x32	0 - 195Ω	0.00298	0 - 3.12kΩ	0.048
x128	0 - 49Ω	0.000745	0 - 781Ω	0.0119

Volts Measurement Type (VOLT Jumper Setting)

Voltage measurement involves applying voltage across the Base Plate input terminals (10-11 or 12-13), making sure to note the correct polarity, setting the gain accordingly, and then reading the voltage. The input resistance of the AI-10 in voltage mode is 7,246Ω ± 0.3%, and the attached sensor must be able to drive that load in order to obtain consistent results.

Based on the gain setting, the input ranges available are shown in the following table:

AI-10 Analog Input Module Gain Setting and Voltage Range

Gain Setting	Voltage Range
x1	0 - 20V
x2	0 - 10V
x32	0 - 625mV
x128	0 - 156mV

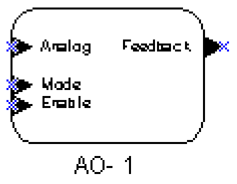
7

The Analog Output Functional Block: Application and Plug-in

This chapter describes how to configure an Analog Output functional block using the LonPoint Plug-in.

The Analog Output Functional Block

The Analog Output functional block translates an input network variable value to a value that is used to drive a hardware output. The functional block may be configured to translate the value into voltage or current, depending on the requirements of the hardware output being driven. The following figure and tables summarize the inputs and outputs of the Analog Output functional block:



Input Network Variables

Default name	Default type	Description
Analog	`SNVT_lev_cont_f`	The Analog input network variable that drives the actuator.
Mode	`SNVT_hvac_mode` (changeable)	The Mode input network variable. See the `Presets` tab for more information.
Enable	`SNVT_switch`	The Enable input network variable. If this network variable is set to On, the functional block will function normally. If this network variable is set to Off, the output to the actuator driven by this functional block will return to its default state.

Output Network Variables

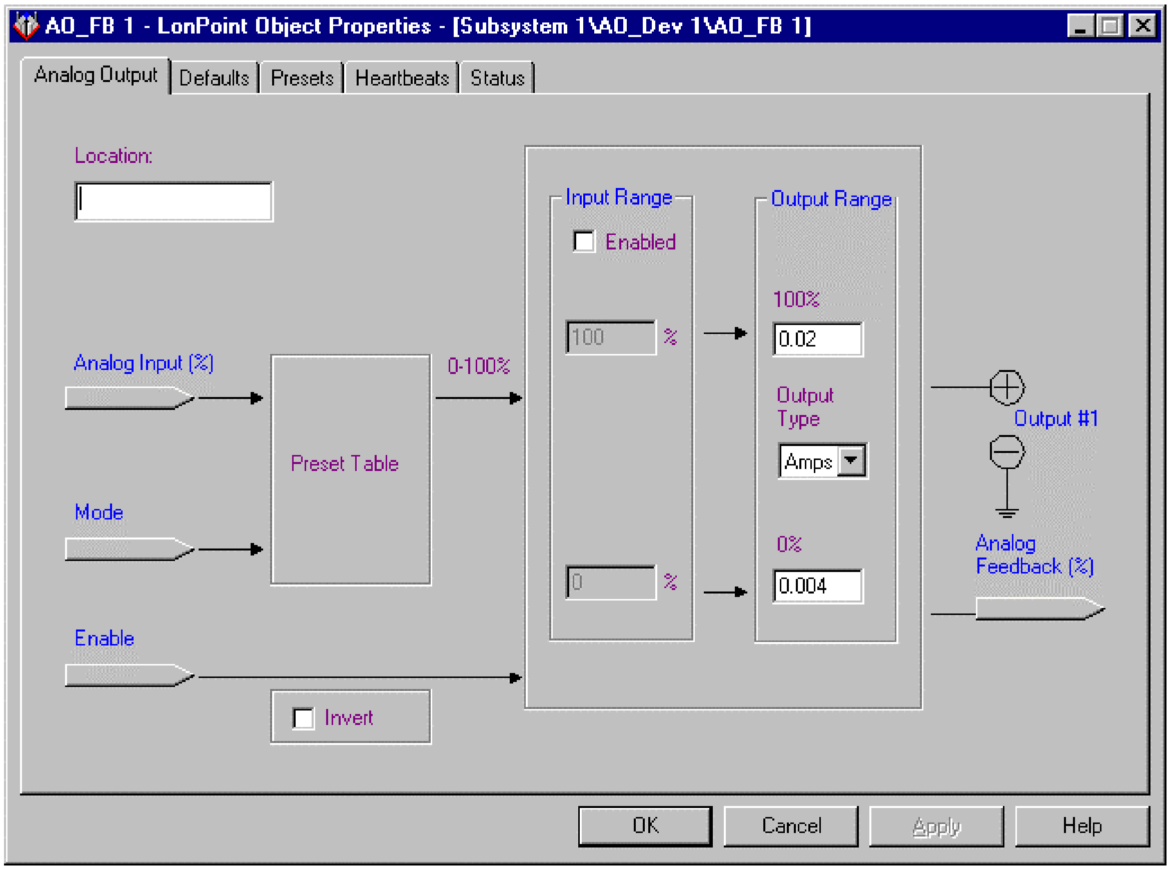
Default name	Default type	Description
Feedback	`SNVT_lev_cont_f`	The sensed output of the actuator driven by this functional block.

Configuring an Analog Output

Right-clicking on an Analog Output functional block and select *Configure* from the shortcut menu to open the LonPoint Plug-in. The Analog Output window of the LonPoint Plug-in opens. This window contains the following tabs: `Analog Output`, `Defaults`, `Presets`, `Heartbeats`, and `Status`.

Analog Output

This tab appears as follows:



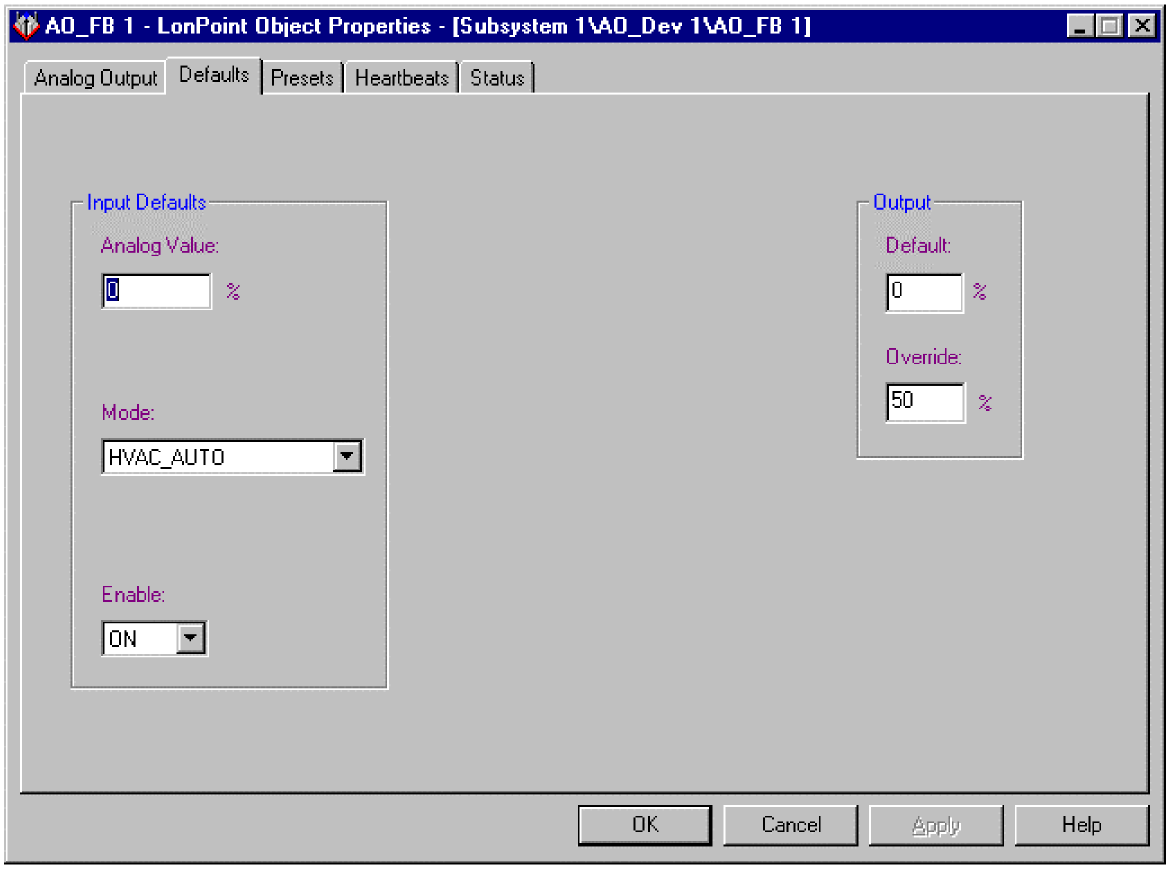
This tab sets configuration properties that determine how the values received by the input network variables are processed by the functional block and what values are sent to the physical actuator. This tab allows you to set the following fields:

Location	The location string for this analog output. This property can be used to document the associated actuator's location within the plant so it can be easily found. This field may contain up to 30 characters. This value is not the device's location property.
Invert	Inverts the value received on the `Enable` network variable. If this option is selected, sending an Off value to the `Enable` network variable will enable the input.
Input Range	Defines the minimum and maximum input values that correspond to the minimum and maximum output values. If the input falls below the minimum, the output is set to the minimum output value. If the input exceeds that maximum, the output is set to the maximum output value. If the `Enabled` option is not

	selected, the input range is 0 to 100%. Input ranging is useful for split-range applications where a single value drives multiple actuators, but each actuator is active over a different input range.
	The analog output functional block can be configured to be reversing (i.e. go from 0% to 100% as the input goes from 100% to 0%) by switching either the Input Range or the Output Range values.
Output Range	Defines the minimum and maximum range of the output. The Output Type field determines whether a current or voltage will be sent to the actuator. These units must match the hardware jumpers set in the device (see the *LonPoint Module Hardware and Installation Guide*).

Defaults

This tab appears as follows:


AO_FB 1 - LonPoint Object Properties - [Subsystem 1\AO_Dev 1\AO_FB 1]

Input Defaults

Analog Value: 0 %

Mode: HVAC_AUTO

Enable: ON

Output

Default: 0 %

Override: 50 %

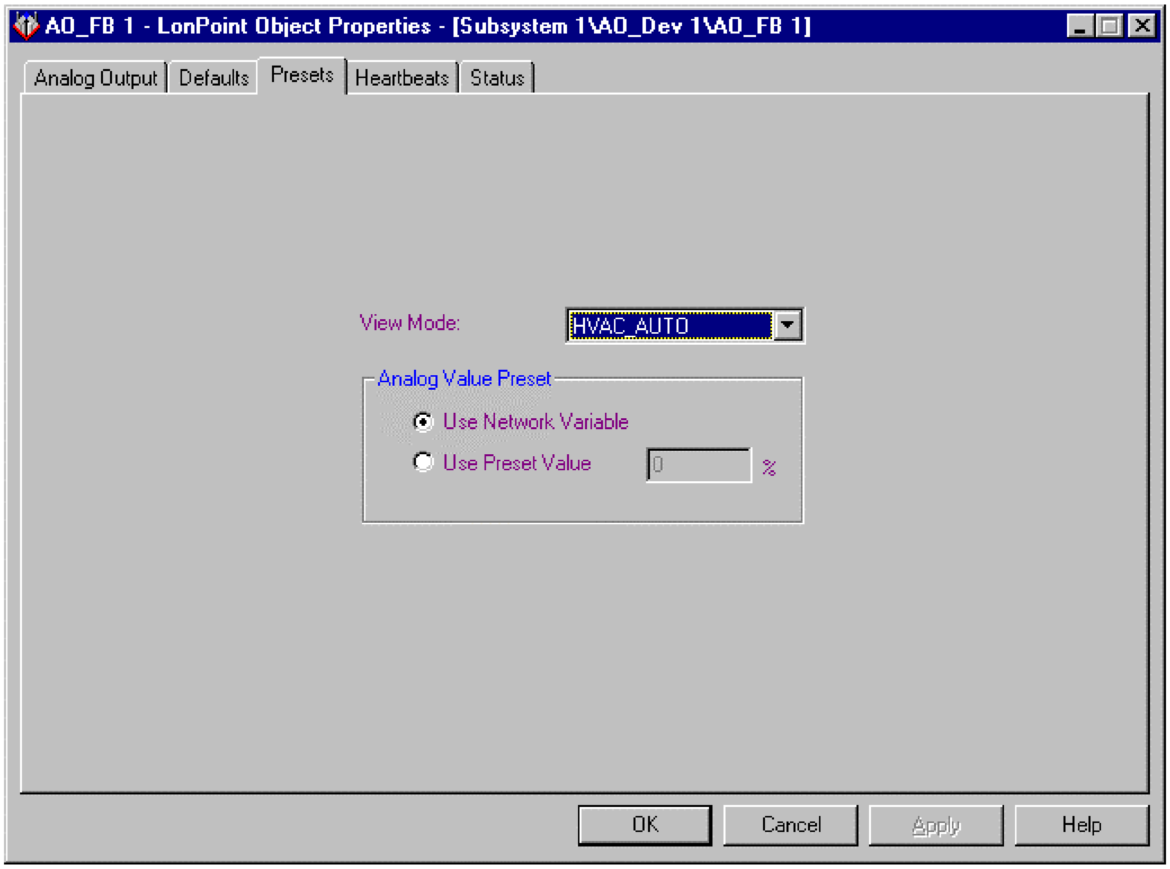
This tab sets the default and override values for the analog output functional block. This tab contains the following fields:

Analog Value Default	This value will be used if the `Analog` network variable is not connected or has not received an update since the last device reset.

Mode Default	This value will be used if the `Mode` network variable is not connected or has not received an update since the last device reset.
Enable Default	This value will be used if the `Enable` network variable is not connected or has not received an update since the last device reset.
Output Default	This value will be sent to the physical actuator if the functional block is disabled or has detected a heartbeat failure.
Override	This value will be sent to the hardware output if the functional block is put into override.

Presets

This tab appears as follows:


AO_FB 1 - LonPoint Object Properties - [Subsystem 1\AO_Dev 1\AO_FB 1]

View Mode: HVAC_AUTO

Analog Value Preset

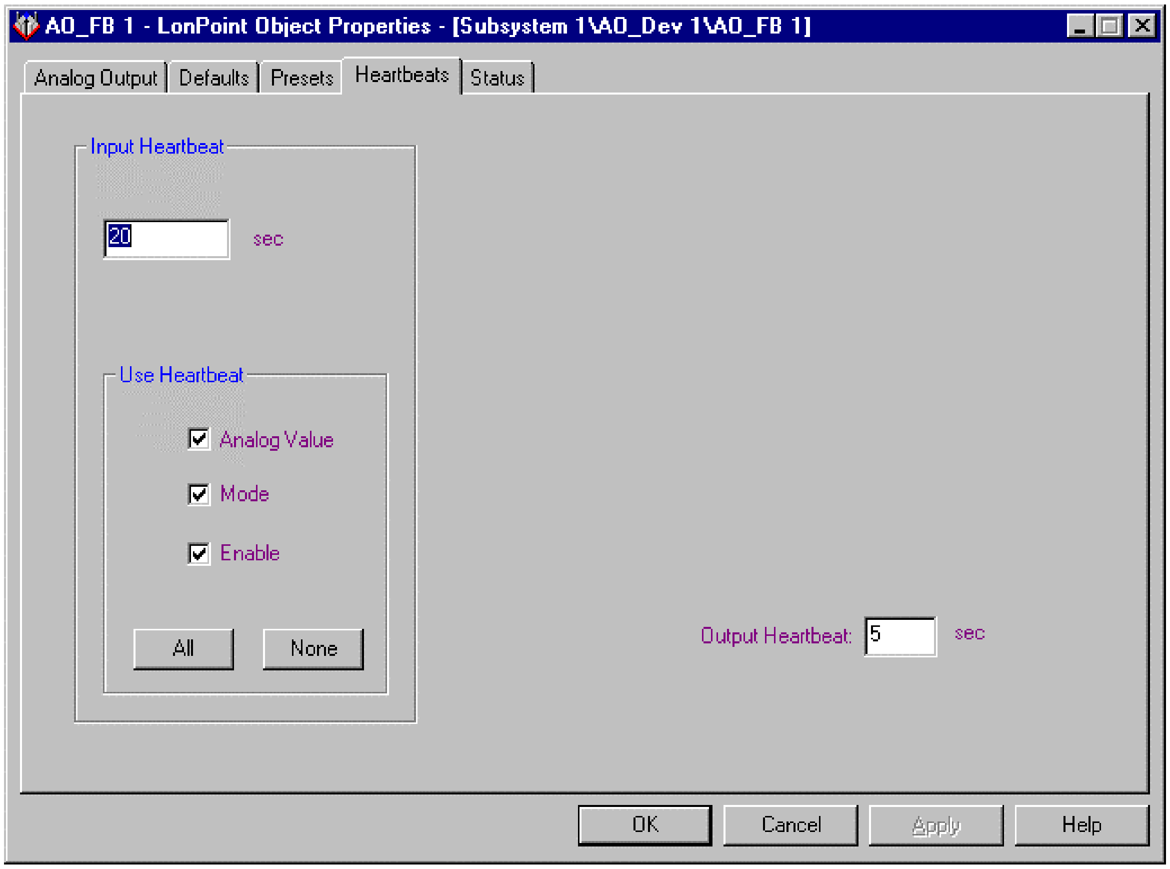
(●) Use Network Variable

( ) Use Preset Value 0 %

This tab allows you to set up the preset table that determines how the Analog output behaves for each mode value. Each value in the View Mode list corresponds to a mode value. For each value that may be received by the `Mode` network variable, you can specify whether the output value to the actuator will be taken from the `Analog` input network variable or set to a preset value. Select `Use Network Variable` to use the network variable value for the selected mode. Select `Use Preset Value` to use the specified preset value.

Heartbeats

This tab appears as follows:


AO_FB 1 - LonPoint Object Properties - [Subsystem 1\AO_Dev 1\AO_FB 1]
Analog Output
Defaults
Presets
Heartbeats
Status
OK
Cancel
Apply
Help

Input Heartbeat: 20 sec

Use Heartbeat:
- [✓] Analog Value
- [✓] Mode
- [✓] Enable

All | None

Output Heartbeat: 5 sec

This tab allows you to configure the input heartbeat rates. This tab contains the following fields:

Input Heartbeat	Determines the interval (in seconds) the network variables that have the `Use Heartbeat` option selected will wait for a heartbeat before registering a heartbeat failure.
Use Heartbeat	Specifies whether or not heartbeat checking will be used for the selected input network variables. Click the All or None buttons to turn heartbeats on or off for all network variables.
Output Heartbeat	Determines the heartbeat send time in seconds for the `Feedback` network variable. A value of 0 disables the output heartbeat.

Status

This tab allows you to view and change the status of an Analog Output functional block. See *Status* in Chapter 2 for more information.

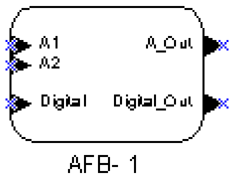
8

The AFB Functional Block: Application and Plug-in

This chapter describes how to configure an Analog Function Block (AFB) functional block using the LonPoint Plug-in.

The Analog Function Block (AFB) Functional Block

The AFB functional block performs mathematical, logical, or enthalpy functions using up to 2 analog inputs and a digital input (not all inputs are used for all functions). The following figure and tables summarize the inputs and outputs of the AFB functional block:



Input Network Variables

Default name	Default type	Description
A1/A2	`SNVT_temp_f` (changeable)	Two Analog input network variables that are processed by the AFB functional block.
Digital	`SNVT_switch`	A Digital input network variable that is processed by the AFB functional block.

Output Network Variables

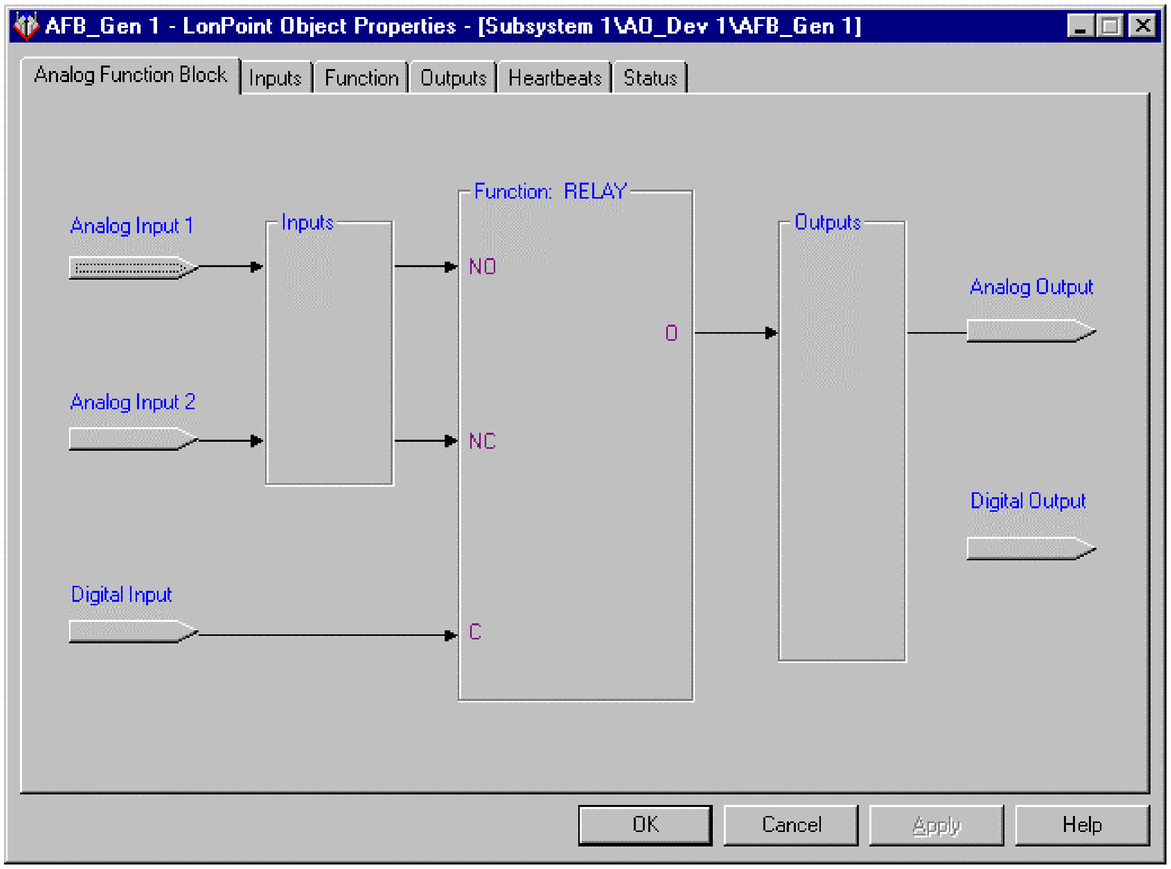
Default name	Default type	Description
A_Out	`SNVT_temp_f` (changeable)	An Analog output network variable set by the AFB functional block.
Digital_Out	`SNVT_switch`	A Digital output network variable set by the AFB functional block.

Configuring the AFB Functional Block using the LonPoint Plug-in

Right-click an AFB functional block shape and select *Configure* from the shortcut menu to open the AFB window of the LonPoint Plug-in. The AFB window contains six tabs: `Analog Function Block`, `Inputs`, `Function`, `Outputs`, `Heartbeats`, and `Status`.

Analog Function Block

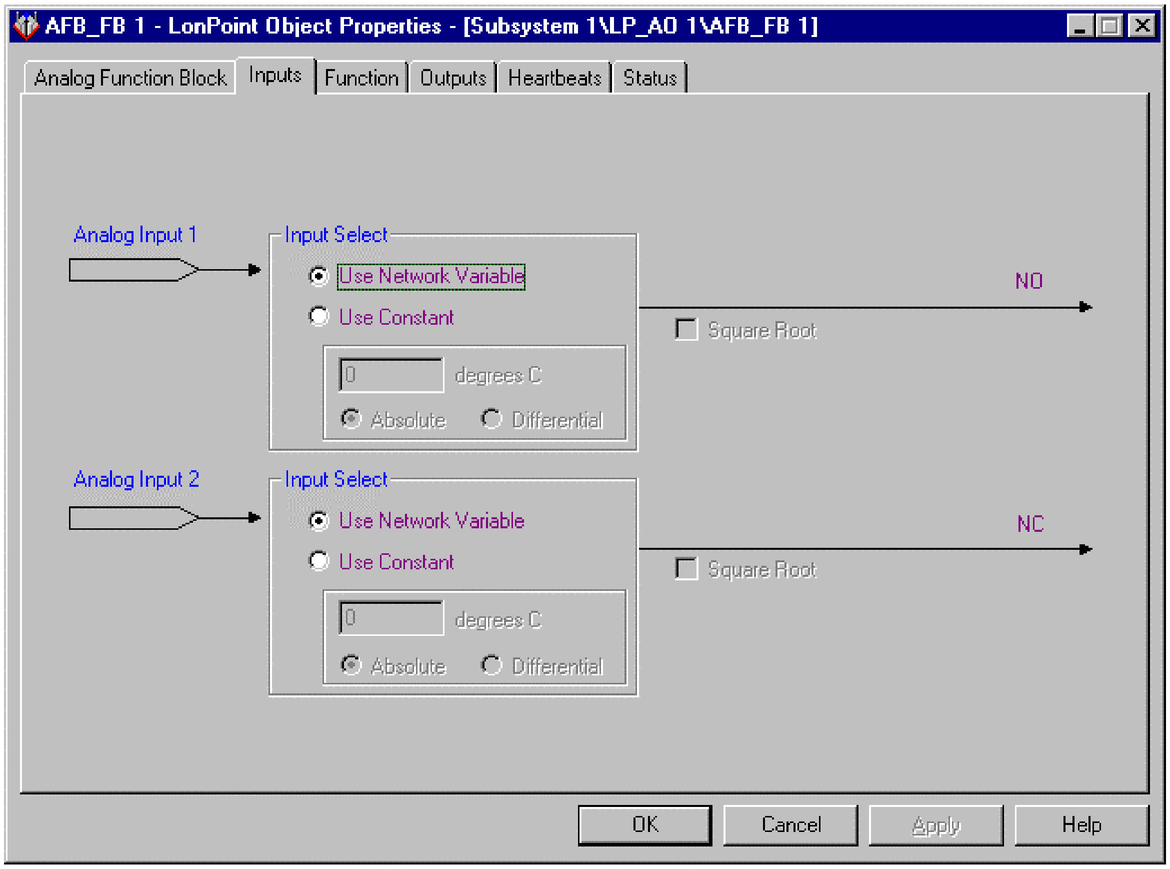
This tab appears as follows:



This tab displays the flow of information through the AFB functional block. For each of the analog inputs, use the `Inputs` tab to determine whether the `A1` or `A2` network variable value or a constant value will be used. Use the `Function` tab to specify the function to be performed on the two values.

Inputs

This tab appears as follows:


AFB_FB 1 - LonPoint Object Properties - [Subsystem 1\LP_AO 1\AFB_FB 1]

Analog Function Block | Inputs | Function | Outputs | Heartbeats | Status

Analog Input 1

Input Select
- [✓ Use Network Variable]
- Use Constant

0 degrees C
- [✓ Absolute]
- Differential

NO

☐ Square Root

Analog Input 2

Input Select
- [✓ Use Network Variable]
- Use Constant

0 degrees C
- [✓ Absolute]
- Differential

NC

☐ Square Root

OK | Cancel | Apply | Help

This tab determines what values are used for the analog operands. For each of the two operands, you can use the associated network variable by selecting `Use Network Variable`, or use a constant by selecting `Use Constant` and setting the corresponding field to the desired constant value. The constant type depends on the input network variable type specified on the Analog Function Block tab. Select the `Square Root` option to have the square root of the associated value be passed to the calculation. You can use the `Square Root` option only if you choose the `Use Network Variable` option and you select a math function. None of the values or options on this tab can be changed if the `Enthalpy` mode is selected on the `Function` tab.

The `Absolute` and `Differential` options apply to the constant values and are only enabled if the corresponding analog input network variable is using the `SNVT_temp_f` network variable type with the Fahrenheit format (`SNVT_temp#US`). If `Absolute` is set, the network variable or constant input will be interpreted as a point on the temperature scale (i.e. 32 degrees Fahrenheit would be interpreted as 0 degrees Celsius). If `Differential` is set, the network variable or constant will be treated as a difference between two temperatures (i.e. 0 degrees Fahrenheit would be interpreted as 0 degrees Celsius). If the

Analog Input network variable is formatted as `SNVT_temp_f#US_diff`, this option is disabled and `Differential` is automatically selected.

This option can only be modified by the user if you are using a `Logic` function (see the *Function* tab). All temperatures are internally represented in degrees Celsius and all calculations are performed in Celsius, therefore, if you are using Fahrenheit values, you must indicate whether the Fahrenheit temperature from each input is to be interpreted as an actual temperature or a differential. For example, if you were comparing a Fahrenheit temperature input to a setpoint constant to determine which one was greater, select the `Absolute` setting. If you were comparing the difference between two Fahrenheit temperatures (as calculated by another AFB functional block) to a constant to determine whether the difference between them was larger than the constant, select the `Differential` setting.

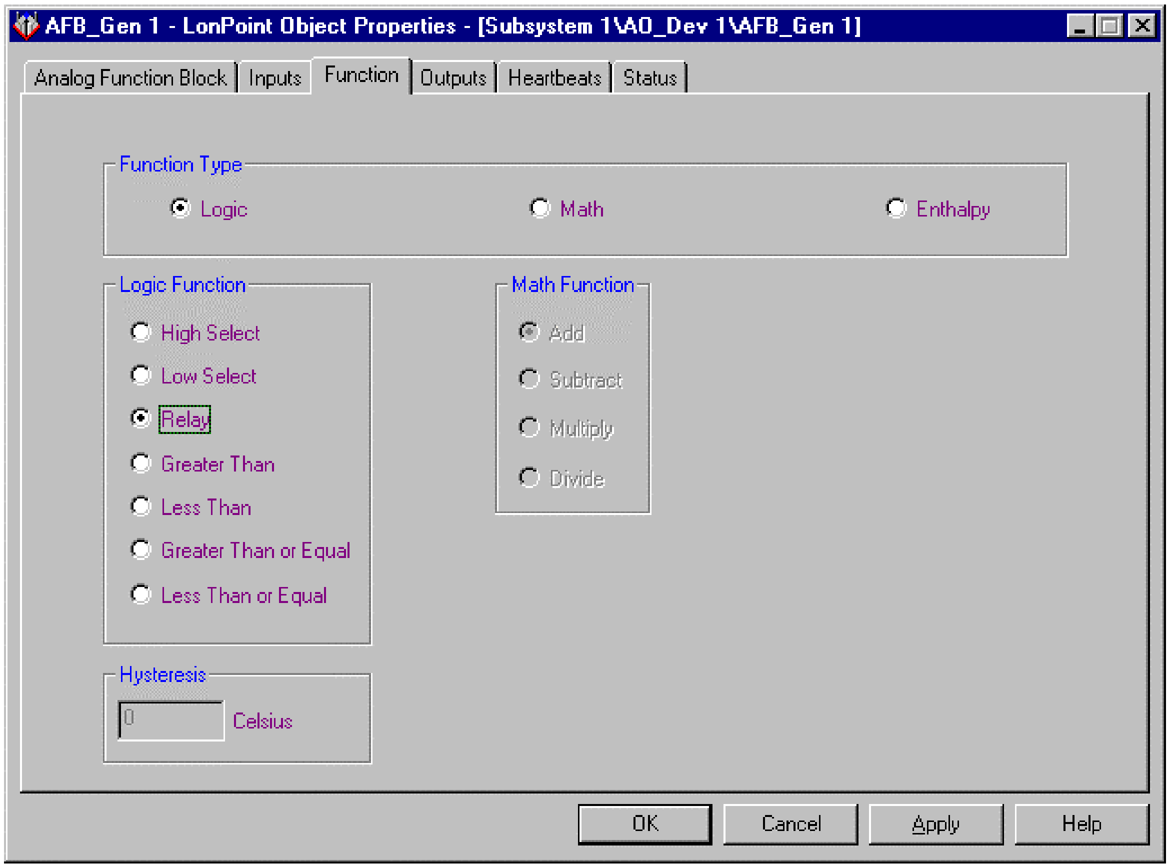
If the *Add* or *Subtract* `Math` function is chosen (see the *Function* tab), this option is automatically set to `Differential`. If the *Multiply* or *Divide* `Math` function is chosen, this option is automatically set to Absolute for input network variables (input constants will be treated as unitless). Multiplying temperatures displayed as Fahrenheit may yield unexpected results, since calculations are performed in Celsius and `32 * 2` is not equivalent to `0 * 2`.

Note:

> The `Absolute/Differential` setting is not saved by the plug-in. When you start the plug-in, the default setting is `Absolute`. If you previously entered a constant value using the `Differential` setting, you need to reselect this setting each time you start the plug-in for the value to be interpreted correctly.

Function

This tab appears as follows:


AFB_Gen 1 - LonPoint Object Properties - [Subsystem 1\AO_Dev 1\AFB_Gen 1]

Analog Function Block | Inputs | Function | Outputs | Heartbeats | Status

Function Type: [✓ Logic] Math Enthalpy

Logic Function: High Select, Low Select, [✓ Relay], Greater Than, Less Than, Greater Than or Equal, Less Than or Equal

Math Function: [✓ Add], Subtract, Multiply, Divide

Hysteresis: 0 Celsius

OK Cancel Apply Help

Function Type	Determines whether the analog function block performs a `Logic`, `Math`, or `Enthalpy` function.

The `Logic Function` options determine what operation will be used if the function type is set to `Logic`. The following options are available:

High Select	The higher of the two analog operands will be sent to the analog output.
Low Select	The lower of the two analog operands will be sent to the analog output.
Relay	The digital input will select which of the two analog operands is sent to the analog output. If the digital input is Off, the *Operand 1* value is sent to `Analog Output`. If the digital input is On, the *Operand 2* value is sent to `Analog Output`.
Greater Than	The `Digital_Out` network variable will be set to On if the value from *Operand 1* is greater than the value from *Operand 2*. Otherwise, an Off value will be sent. Changes to the output value are affected by the `Hysteresis` value (see below).

Less Than	The `Digital_Out` network variable will be set to On if the value from *Operand 1* is less than the value from *Operand 2*. Otherwise, an Off value will be sent. Changes to the output value are affected by the `Hysteresis` value (see below).
Greater Than or Equal	The `Digital_Out` network variable will be set to On if the value from *Operand 1* is greater than or equal to the value from *Operand 2*. Otherwise, an Off value will be sent. Changes to the output value are affected by the `Hysteresis` value (see below).
Less Than or Equal	The `Digital_Out` network variable will be set to On if the value from *Operand 1* is less than or equal to the value from *Operand 2*. Otherwise, an Off value will be sent. Changes to the output value are affected by the `Hysteresis` value (see below).

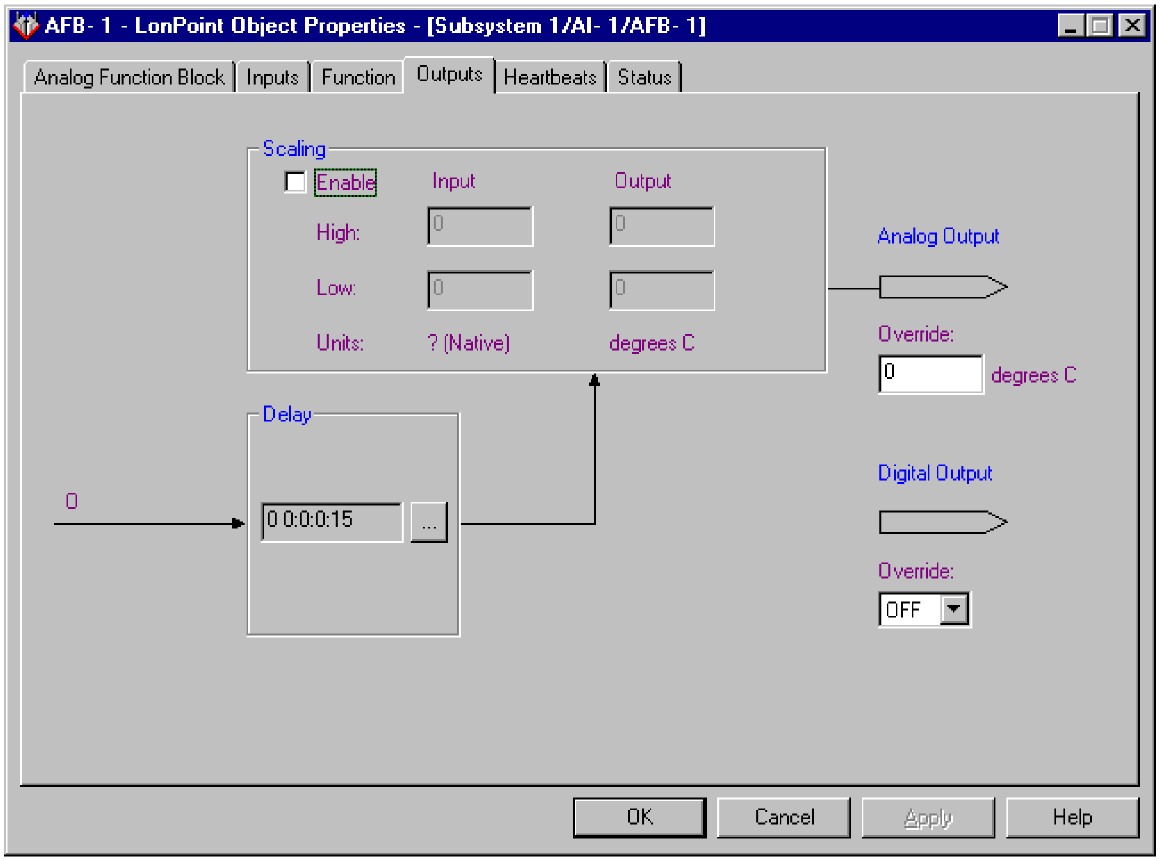
The `Math Function` options determine what operation is to be performed if the `Math` function type was selected in the `Function Type` field. The available options are `Add`, `Subtract`, `Multiply`, and `Divide`. The order of the equation is always `Operand 1` [plus\minus\times\divided by] `Operand 2`.

The Hysteresis field determines how great a difference, as a floating point number, must exist between the analog values for the `Greater Than`, `Less Than`, `Greater Than or Equal`, and `Less Than or Equal` logic functions to change the digital output. You can use this field only if the mode is set to one of these comparison logic operations.

Selecting the `Enthalpy` function calculates the heat content of air, based on the temperature and humidity. `Operand 1` must be a `SNVT_temp_f` type, and `Operand 2` must be a `SNVT_lev_cont_f` type. There is no SNVT defined for enthalpy that can be used for the `Analog Output` network variable, but the Echelon user type file (ECHELON.TYP) provides a `UNVT_enthalpy` user type that you can use for this purpose.

Outputs

This tab appears as follows:



This tab determines what value is sent to the `A_Out` and `Digital_Out` network variables based on the output from the Math, Logic, or Enthalpy function performed by the Analog Function Block.

Delay	Specifies a time that the result of the calculation made by the AFB must remain constant to be sent on the appropriate output network variable. The value can be up to 48 days, 23 hours, 59 minutes, 59 seconds, and 999 milliseconds. To change the value, click the button to the right of the time value to be changed and enter the new values to be used for the delay.
Scaling	Set up scaling for analog values. The `Input` column represents the value derived from the AFB function, and the `Output` column represents the value that will be sent on the `A_Out` network variable. For each side, a `High` and `Low` value must be set. As the result of the function moves from the `Low` to the `High` value on the left, the value sent to the network moves from the `Low` to the `High` value on the right. The scaling is done linearly.

The type of the values in the Output column depend on the Analog Output network variable type you selected on the Analog Function Block tab.

The type of the values in the Input column cannot always be determined. For Add and Subtract functions the units are normally displayed. All other cases show units as ? (Native) and the values shown are in (or derived from) the native units of the input network variables (normally SI units). Caution should be used when scaling Fahrenheit temperatures. You can temporarily change the format of the input network variables to SI units to verify your values.

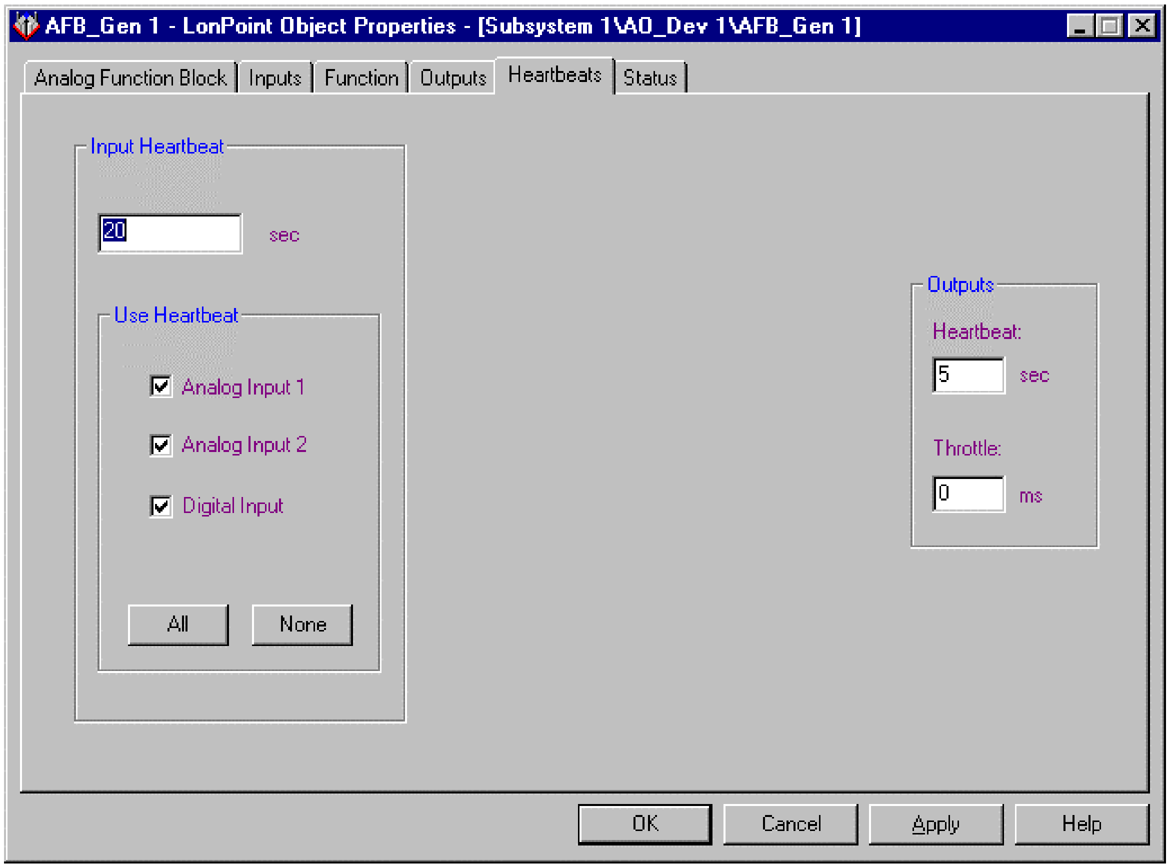
Scaling can be set up to be Reverse Acting (i.e. the output decreases as the input increases) by setting the High value lower than the Low value for either the Input or the Output column (but not both).

Scaling is applicable to all analog output function types except Enthalpy.

Override This field exists for both the A_Out and Digital_Out network variables. This value will be sent on the corresponding output network variable when the AFB functional block is put into override.

Heartbeats

This tab appears as follows:



Input Heartbeat: 20 sec

Use Heartbeat:
- [✓] Analog Input 1
- [✓] Analog Input 2
- [✓] Digital Input

Outputs:
Heartbeat: 5 sec
Throttle: 0 ms

This tab determines the heartbeat and throttle values for the input and output network variables. This tab contains the following fields:

Input Heartbeat	Determines the interval (in seconds) the network variables that have the Use Heartbeat option selected will wait for a heartbeat before detecting a heartbeat failure.
Use Heartbeat	Determines whether the associated input network variable uses heartbeat checking. The `All` and `None` buttons allow you to enable heartbeat checking for all input network variables or none of the input network variables, respectively.
Output Heartbeat	Determines the output heartbeat, in seconds, for the `A_Out` and `Digital_Out` network variables. This is the maximum amount of time that may pass between network variable updates. Set this value to zero to disable heartbeats.
Throttle	Determines the throttle, in milliseconds, for the `A_Out` and `Digital_Out Output` network variables. This is the minimum amount of time between

network variable updates. When the throttle time expires, only the most recent change to the output value is transmitted. Set this value to zero to disable throttling.

Status

This tab allows you to view and change the status of an AFB functional block. See *Status* in Chapter 2 for more information.

9

The PID Controller Functional Block: Application and Plug-in

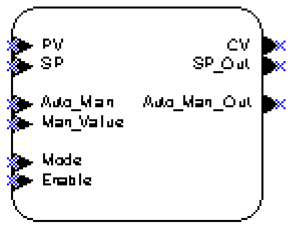
This chapter describes how to configure a PID controller functional block using the LonPoint Plug-in.

The PID Controller Functional Block

The PID Controller functional block controls an output network variable value based on an input process variable and setpoint. The process variable is obtained from a sensor that measures an environmental condition, such as temperature or air pressure. The setpoint indicates the desired value of the process variable. The PID controller reads these values and based on the PID algorithm, outputs a value known as the controlled variable. This variable is used to drive an actuator that effects the environmental condition which is read by the sensor that produces the process variable.

For example, a PID functional block may be used to control the temperature of a room. The process variable would be connected to a sensor that reads the current temperature of the room, the setpoint would be determined manually through the use of a dial or control panel, and the controlled variable would be connected to an actuator that drives a VAV damper.

The following figure and tables summarize the inputs and outputs of the AFB functional block:



PID- 1

Input Network Variables

Default name	**Default type**	**Description**
PV	`SNVT_temp_f` (changeable)	The process variable network variable.
SP	`SNVT_temp_f` (changeable)	The setpoint network variable.
Auto_Man	`SNVT_switch`	The auto/manual network variable. If this network variable is set to On, the PID controller will be in Auto. If this network variable is set to Off, the PID controller will be in manual. See the `Presets` tab for more information.
Man_Value	`SNVT_lev_cont_f`	The manual value network variable. See the `Presets` tab for more information.
Mode	`SNVT_hvac_mode` (changeable)	The mode network variable. See the `Presets` tab for more information.

Enable	`SNVT_switch`	The enable network variable. If this network variable is set to On, the PID Controller functional block will function normally. If this network variable is set to Off, the output network variables on this functional block will return to their defaults.

Output Network Variables

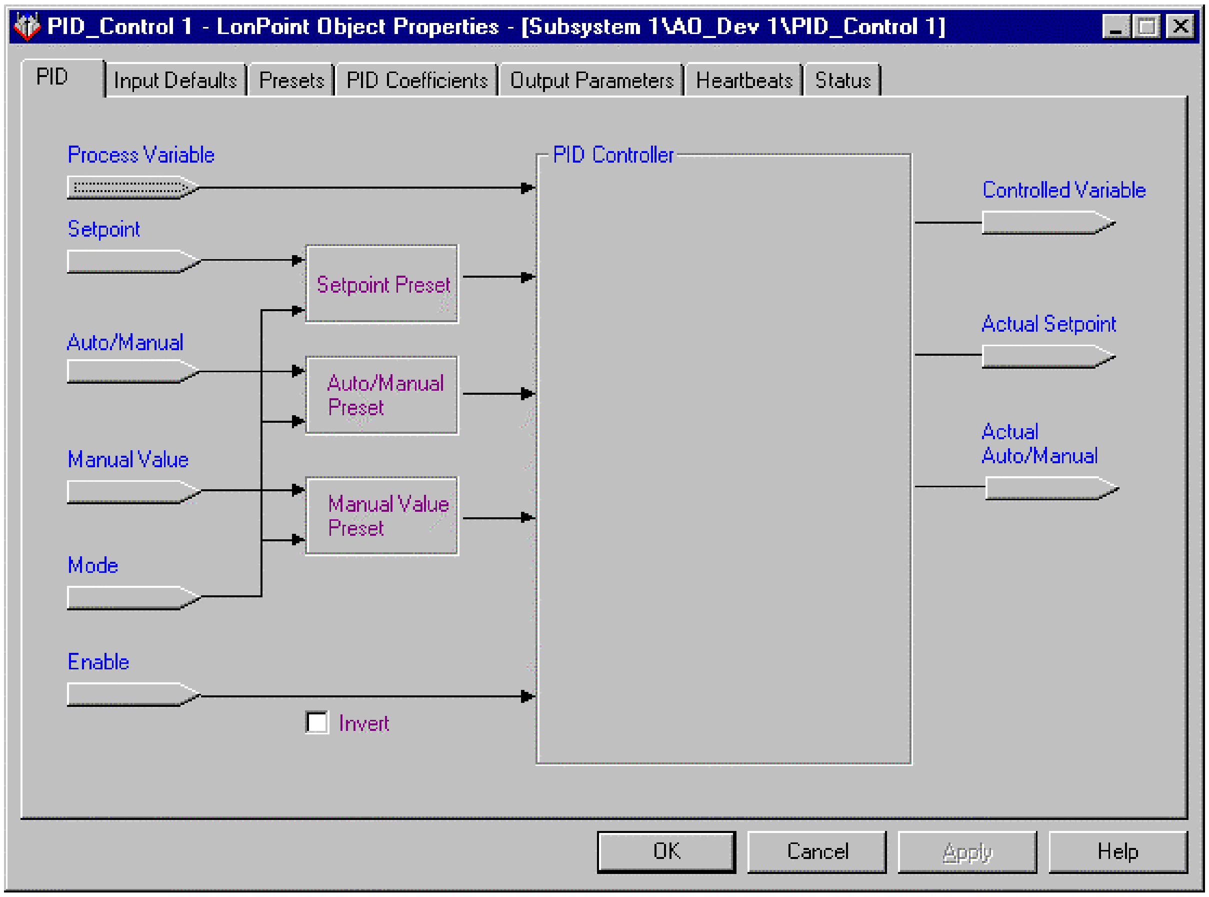
Default name	Default type	Description
CV	`SNVT_lec_cont_f`	The control variable network variable.
SP_Out	`SNVT_temp_f` (changeable)	The setpoint output network variable. This network variable outputs the setpoint being used by this PID Controller. This may differ from the value aquired from the SP network variable. See the `Presets` tab for more information.
Auto_Man_Out	`SNVT_switch`	The auto/manual output network variable. If this network variable is set to On, it indicates that this PID Controller is functioning in Auto. If this network variable is set to Off, it indicates that this PID controller is operating in Manual. See the `Presets` tab for more information.

Configuring a PID Controller with the LonPoint Plug-in

Right-click the functional block shape and select *Configure* from the shortcut menu to open the PID window of the LonPoint Plug-in. This window contains the following tabs: `PID`, `Input Defaults`, `Presets`, `PID Coefficients`, `Output Parameters`, `Heartbeats`, and `Status`.

PID

This tab appears as follows:

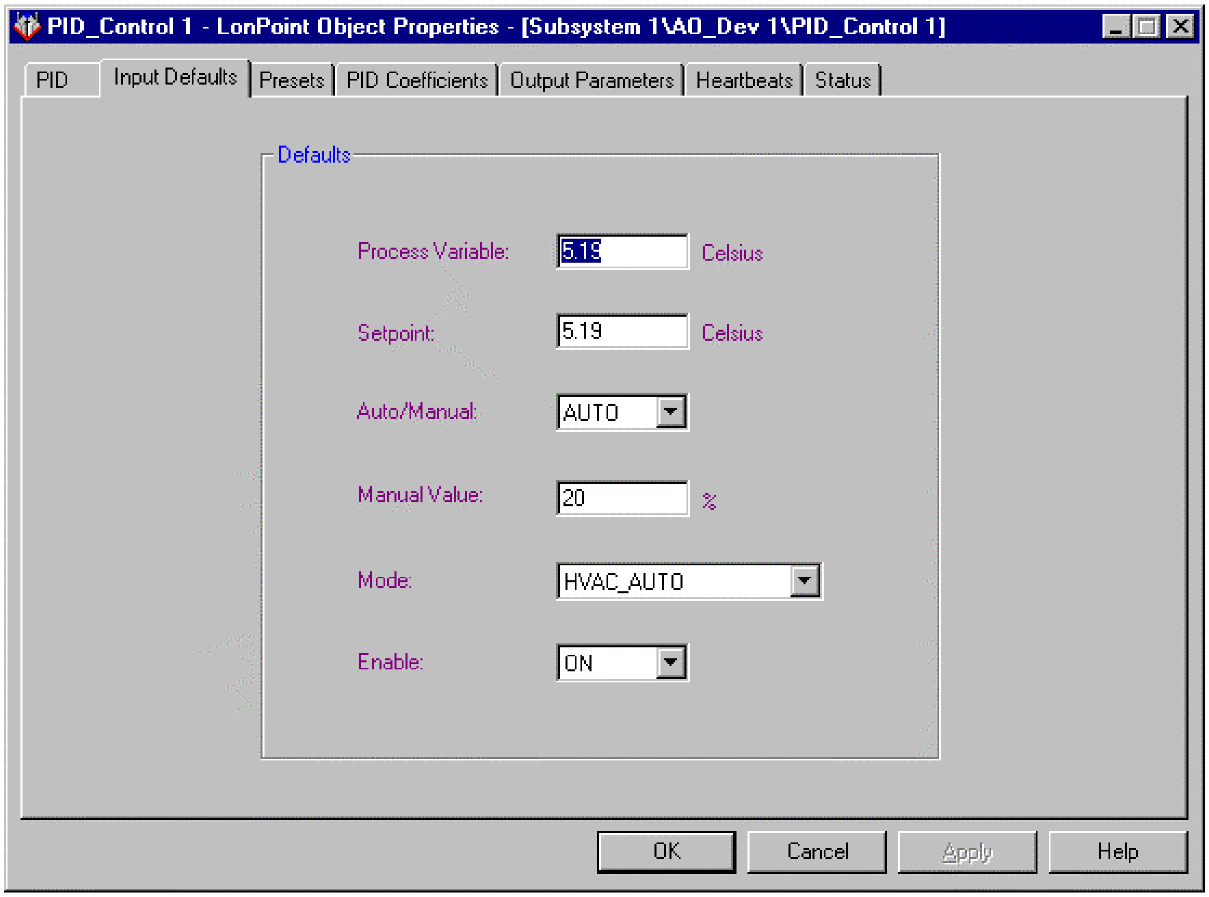


This tab displays the flow of information through the PID functional block. The input values are taken from network variables on the left. The information is passed through the preset tables to determine what values are sent to the PID controller. The PID controller calculates the controlled variable and outputs it to the controlled variable network variable on the right, along with the actual setpoint and auto/manual settings used.

The `Invert` option specifies whether or not the Enable input is inverted. If `Invert` is checked, the PID functional block is enabled when the `Enable` network variable is set to Off.

Input Defaults

This tab appears as follows:



PID_Control 1 - LonPoint Object Properties - [Subsystem 1\AO_Dev 1\PID_Control 1]

Defaults

Process Variable: 5.19 Celsius

Setpoint: 5.19 Celsius

Auto/Manual: AUTO

Manual Value: 20 %

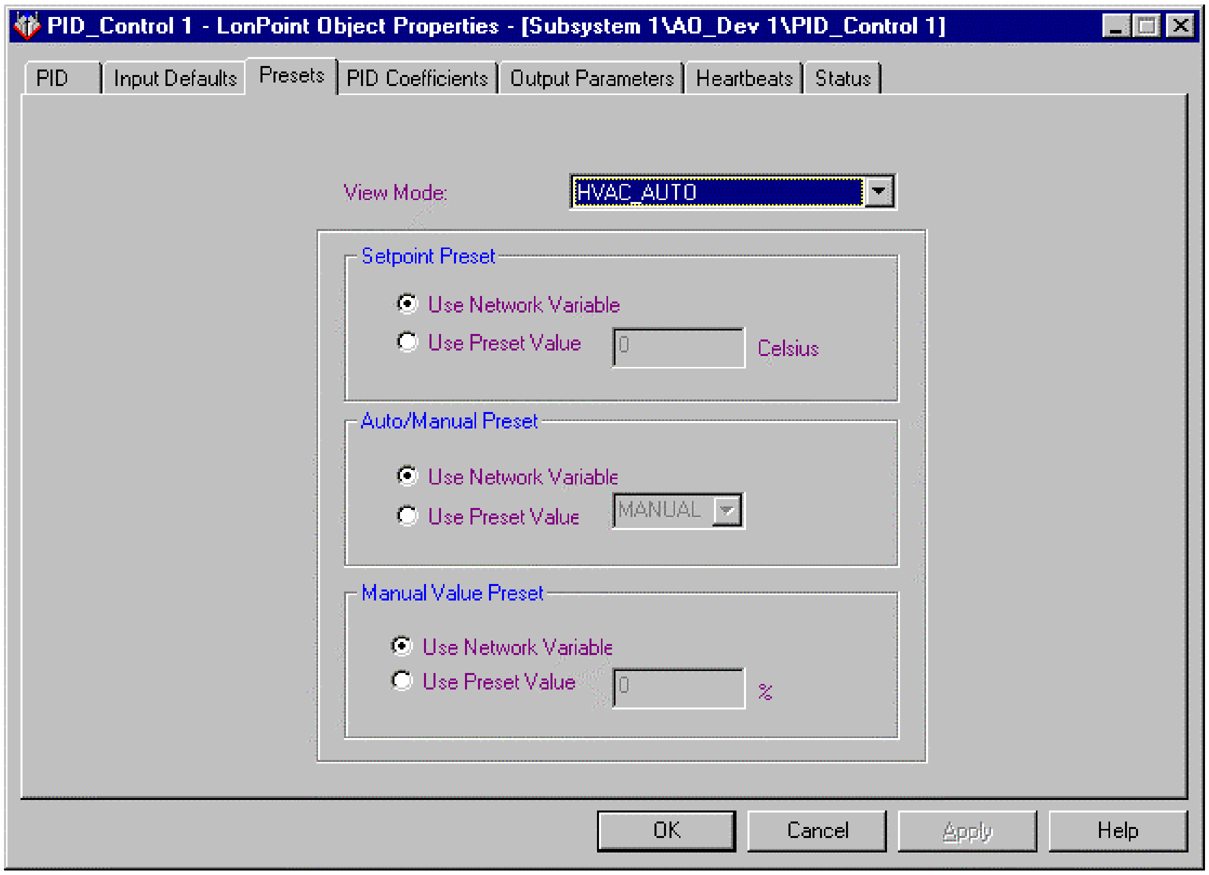
Mode: HVAC_AUTO

Enable: ON

This tab sets the default input values for the input network variables. As described in *Default Values* in Chapter 1, these values will be used if the corresponding input network variables are not connected, or have not received an update since the last device reset. The types of the `Process Variable`, `Setpoint`, and `Mode` values depend on the types selected or the corresponding network variables on the PID tab.

Presets

This tab appears as follows:


PID_Control 1 - LonPoint Object Properties - [Subsystem 1\AO_Dev 1\PID_Control 1]

View Mode: HVAC_AUTO

Setpoint Preset
[✓ Use Network Variable]
Use Preset Value 0 Celsius

Auto/Manual Preset
[✓ Use Network Variable]
Use Preset Value MANUAL

Manual Value Preset
[✓ Use Network Variable]
Use Preset Value 0 %

This tab configures the preset tables. Each preset table can contain up to 16 entries. The `View Mode` field determines which one of the entries is currently being configured. When the PID is running, the preset tables are indexed by the PID's `Mode` network variable to determine which value will be used.

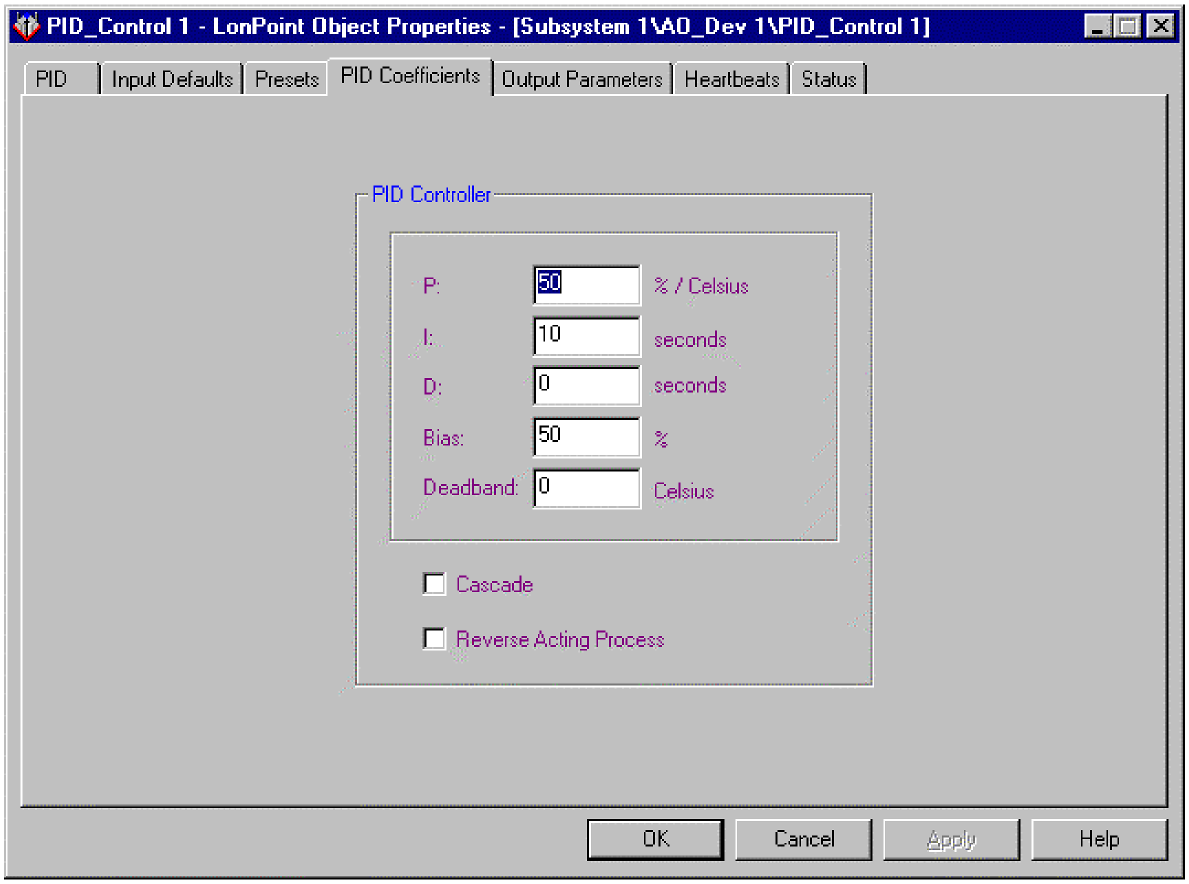
Each entry in a preset table specifies the source of the value controlled by the preset. The source for each value may be the associated network variable or a fixed preset value. If `Use Network Variable` is selected, the value from the network variable is used. If `Use Preset Value` is selected, the value entered in the corresponding preset field is used (i.e., the network variable is not used).

The `Auto/Manual` presets determines whether the PID is in `Auto` or `Manual` operation for this mode. When the PID controller is in `Auto` operation the PID Controller will attempt to make the process variable equal to the setpoint. When the PID controller is in `Manual` operation the controlled variable will be set to the value indicated by the `Manual Value` preset table (i.e., the PID Controller will be bypassed). This allows you to create a preset in which the controlled variable is set to a value sent to the PID through the `Manual Value` network variable or to a preset value.

The `Manual Value` and `Setpoint` presets determine how these values are obtained. They can be read from their corresponding network variables or set to a preset value. The `Manual Value` will only be used when the `Auto/Manual` value is set to `Manual`, and the `Setpoint` will only be used if it is set to `Auto`.

PID Coefficients

This tab appears as follows:



PID Controller

P:	50	% / Celsius
I:	10	seconds
D:	0	seconds
Bias:	50	%
Deadband:	0	Celsius

☐ Cascade

☐ Reverse Acting Process

This tab sets values that effect the algorithm used by the PID controller. This tab allows you to set that following configuration properties:

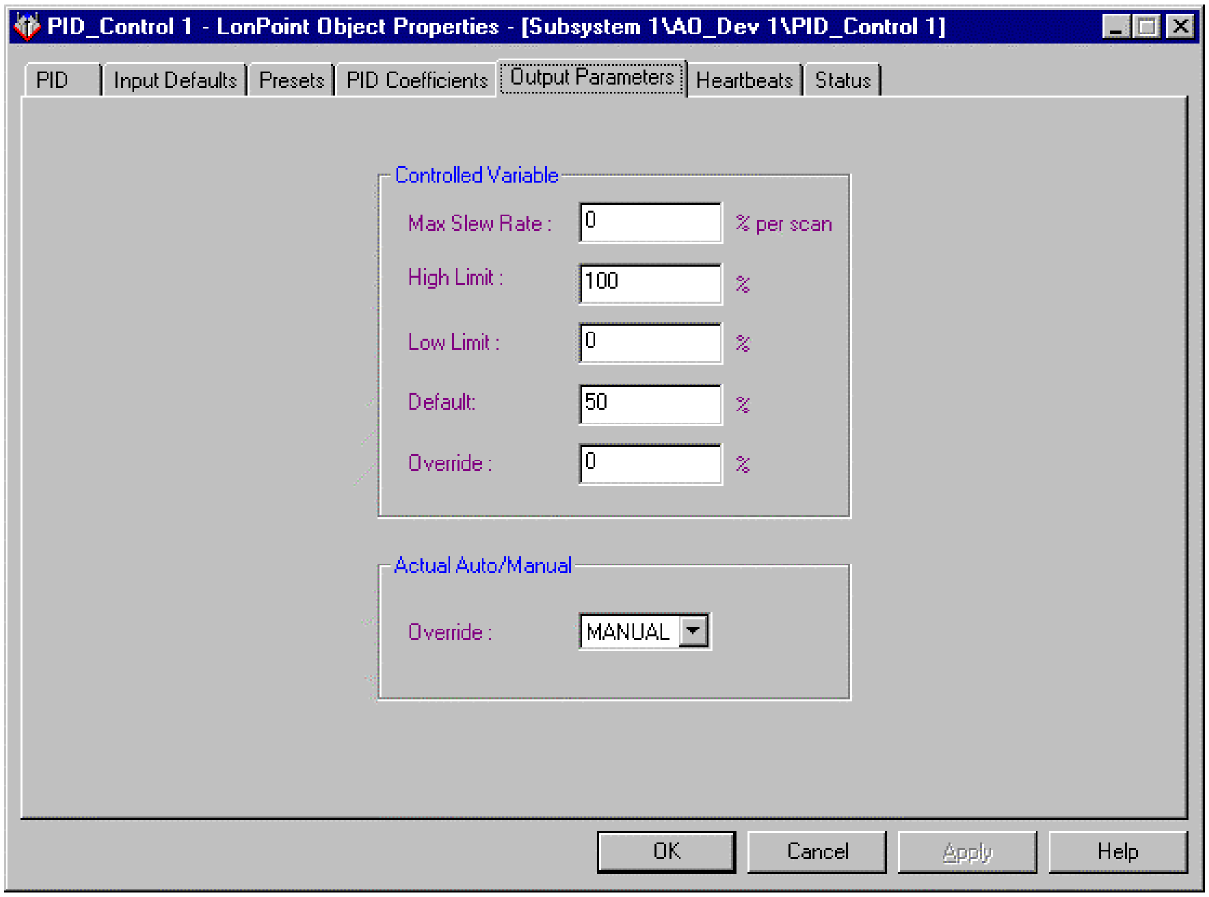
P	The proportional term of the PID controller. Its units are percent output per process variable unit as specified on the PID tab. For example, if the process variable units are degrees C, then the units of this property are percent output/degree C.
I	The integral term of the PID controller. A larger value results in a smaller action (output change per scan). Its units are seconds. Enter 0 to disable this coefficient.
D	The derivative term of the PID controller. A larger value results in a larger action. Its units are seconds. Enter 0 to disable this coefficient.

Bias	Sets the controlled variable output for a P-only controller when there is zero error. This is only used if the integral (I) term is 0.
Deadband	When the difference between the setpoint and the process variable is less than this value, no controlled variable adjustment is made. The process variable is considered close enough. Deadband may be used to reduce actuator wear.
Cascade	Causes the controlled variable to react more strongly to changes in the setpoint for the PI and PID configurations. If you need to track setpoint changes quickly, turn *Cascade* on. If you want good response to load disturbances and setpoint response is less important, turn *Cascade* off. This is option is called *Cascade* because when PID controllers are in a cascaded configuration, with the outer PID's controlled variable feeding the inner controller's setpoint, setpoint changes to the inner controller are as important as process disturbances. Make sure to turn *Cascade* on for the inner controller in a cascade arrangement. When this option is set, two changes are made to the control algorithm: The setpoint value is not used when proportional contribution is calculated and the derivative contribution is given by `d/dt(error)` instead of `d/dt(PV)`.
Reverse Acting Process	Specifies whether or not the process is reversing. If an increase to the controlled variable causes the process variable to decrease, then the process is considered to be reversing. The process includes the sensors and actuators. A simple open-loop test can be made to determine how to set this property. Put the controller in override and wait for the process variable to stabilize, then increase the override value. If the process variable decreases, the process is reversing.

See *LonPoint PID Controller Tuning*, later in this chapter, for more information on setting these values.

Output Parameters

This tab appears as follows:


PID_Control 1 - LonPoint Object Properties - [Subsystem 1\AO_Dev 1\PID_Control 1]

PID | Input Defaults | Presets | PID Coefficients | Output Parameters | Heartbeats | Status

Controlled Variable

Max Slew Rate : 0 % per scan

High Limit : 100 %

Low Limit : 0 %

Default: 50 %

Override : 0 %

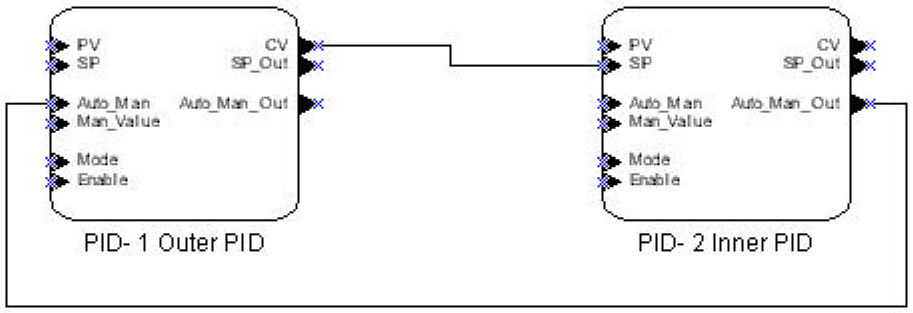
Actual Auto/Manual

Override : MANUAL

This tab sets properties that effect the value output from the PID controller, and also determines the override values.

Max Slew Rate	Limits how fast the controlled variable output can change. This limit is in effect only during soft start. Soft start begins when the functional block is reset and it ends when the process variable value reaches the setpoint. Units are percent output change per scan interval (set in the `Heartbeats` tab). A value of 0 disables slew rate limiting (a value of 100 or greater also disables limiting since the output can change across the whole range each time the PID executes).
High Limit/Low Limit	Determines the upper and lower limits for the controlled variable. If the PID tries to set the controlled variable to a value outside of this range, the value will be clipped to the specified limit.

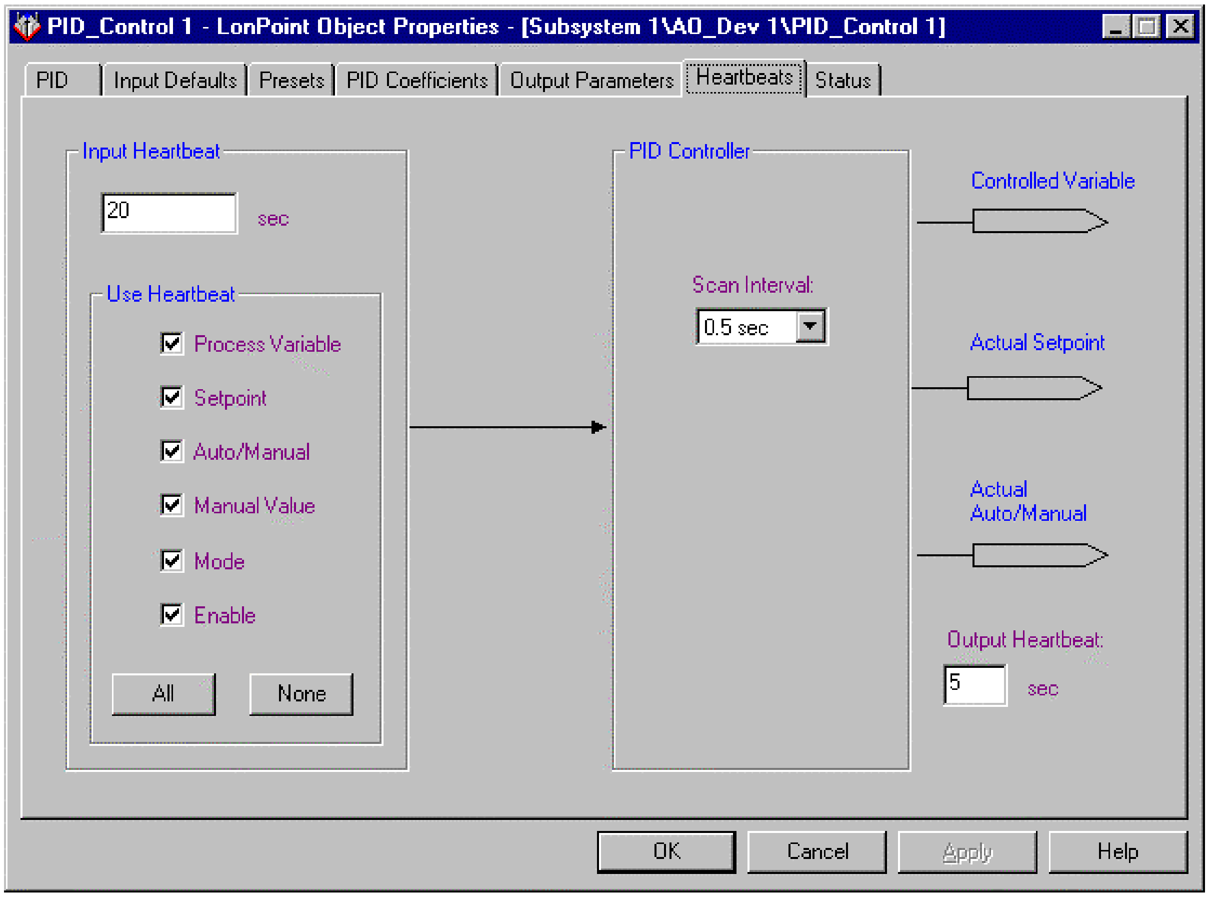
Default Values	The default value for the controlled variable output. This value will be set in case of a heartbeat failure or if the PID is disabled using the `Enable` input.
Controlled Variable Override	Sets the value of the controlled variable output when this PID functional block is put into override mode.
Actual Auto/Manual Override	The `Auto_Man_Out` network variable indicates whether the PID is in Auto (On `Auto_Man_Out` output) or Manual (Off `Auto_Man_Out` output) operation. In situations where PID controllers are cascaded, with the inner-loop controller taking its setpoint from an outer-loop controller's `CV` network variable, the outer-loop controller must know the `Auto/Manual` setting of the inner loop controller. This is illustrated in the following figure. When the inner loop controller is in override, it is not controlling and is in `Manual` operation.



This configuration property determines the value of the `Auto_Man_Out` output when this PID functional block is put into override. Set the value to `Manual` to accurately reflect the actual auto/manual setting when the PID is in override.

Heartbeats

This tab appears as follows:


PID_Control 1 - LonPoint Object Properties - [Subsystem 1\AO_Dev 1\PID_Control 1]

Input Heartbeat: 20 sec

Use Heartbeat:
- [✓] Process Variable
- [✓] Setpoint
- [✓] Auto/Manual
- [✓] Manual Value
- [✓] Mode
- [✓] Enable

Scan Interval: 0.5 sec

Output Heartbeat: 5 sec

This tab determines input and output heartbeats, as well as the execution rate of the PID controller.

Input Heartbeat	Determines the interval (in seconds) the network variables that have the Use Heartbeat option selected will wait for a heartbeat before registering a heartbeat failure.
Use Heartbeat	Determine whether the corresponding network variable checks its heartbeat. Click the All or None buttons to enable heartbeat checking for all input network variables or none of the input network variables, respectively.
Scan Interval	Indicates how often (in milliseconds) the PID controller will execute (i.e. compare the setpoint and the process variable and send a value to the controlled variable). This may be set to 0.5 sec, 1 sec, 2 sec, 4 sec, 8 sec, 16 sec, or 32 sec.
Output Heartbeat	Determines how often the output network variables send a heartbeat. The expected heartbeat rate and

failure conditions are determined by the functional blocks that contain the corresponding input network variables.

Status

This tab allows you to view and change the status of a PID controller functional block. See *Status* in Chapter 2 for more information.

LonPoint PID Controller Tuning

This section describes a simple method that can be used to set initial values for the PID coefficients. An open-loop experiment is done to determine the static gain and the basic dynamic properties of the process being controlled. These values are entered into a program that calculates the P, I, and D coefficients for the PID controller. The coefficients are loaded into the PID controller using the LonPoint Plug-in and the LonMaker for Windows Integration Tool. The coefficients match or tune the controller for the controlled process.

Figure 8.1 shows an example PID loop for reference. The Analog Inputs come in from the left and connect to the PID setpoint (SP)and process variable (PV) inputs. AI- 2 controls the setpoint, which can range from zero to one hundred percent. In practice the setpoint and process variable will be in terms of some engineering unit, temperature in degrees C for example. The PID calculates the control value, CV, and sends it to the Analog Output, AO- 1. AO- 1 is wired to an actuator that drives the plant or process. The process is sampled by the Analog Input, AI- 1, and connecting this back to the PV input of the controller closes the loop.

In this example there are unconnected inputs on the PID. When inputs are not connected, default inputs values are used. These defaults can be set on the *Input Defaults* tab.

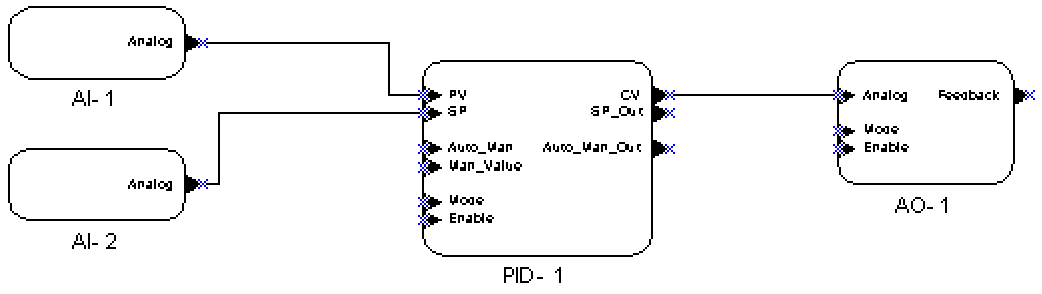
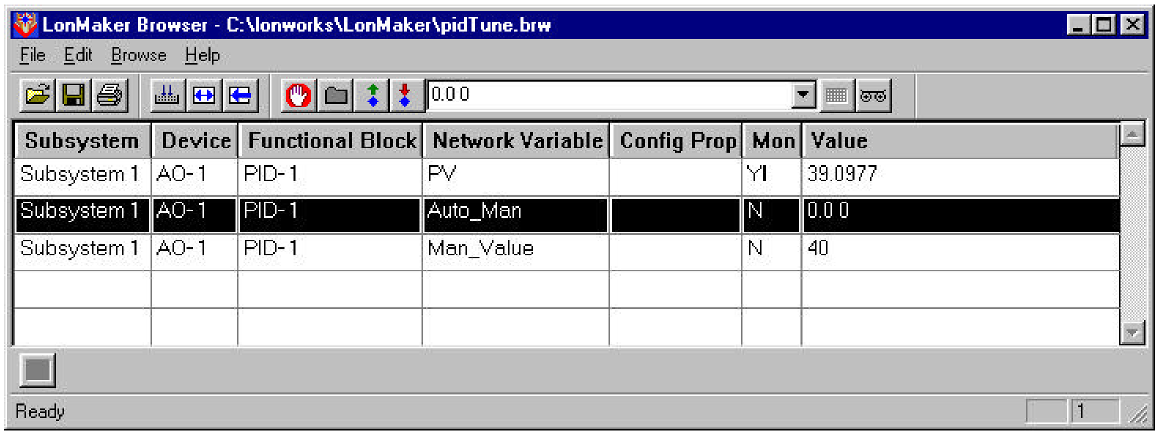


Figure 8.1 PID Example Drawing

To calculate the PID coefficients, you can perform an experiment. To do the tuning experiment, put the PID controller into manual mode, set the manual value and wait for the sensed process variable to reach equilibrium. Then the manual value is changed to a different value; this is the ‘step’ part of the experiment. The proper step size depends on the controlled process. With some trial and error experience, it should not be difficult to choose a reasonable step size. The largest possible step is not necessarily better than a small step; you have to wait longer for the large step to reach its new process value. Too small of a step will make it too hard to see when the process has settled. This example uses a 20% step size.

To perform the experiment and calculate the PID coefficients, follow these steps:

1. Open your design with the LonMaker tool, making sure to put your network OnNet.
2. Right-click the PID controller function block and select Browse from the context menu.
3. Use the Delete key to remove unnecessary data from the Browser window. Leave just the rows as shown below.
4. Change the value of the Auto_Man input by entering 0.0 0 in value field.
5. Set the `Man_Value` network variable to the initial value for the experiment. The example shows `Man_Value` as 40%. This puts the PID controller is in manual mode, and forces its control output to 40%.
6. Right-click the PV network variable and select Monitor Value to turn on monitoring of the process.
7. Wait for the process reading to stabilize.
8. Record the manual value and the process variable. For the example, these are 40% and 39%.



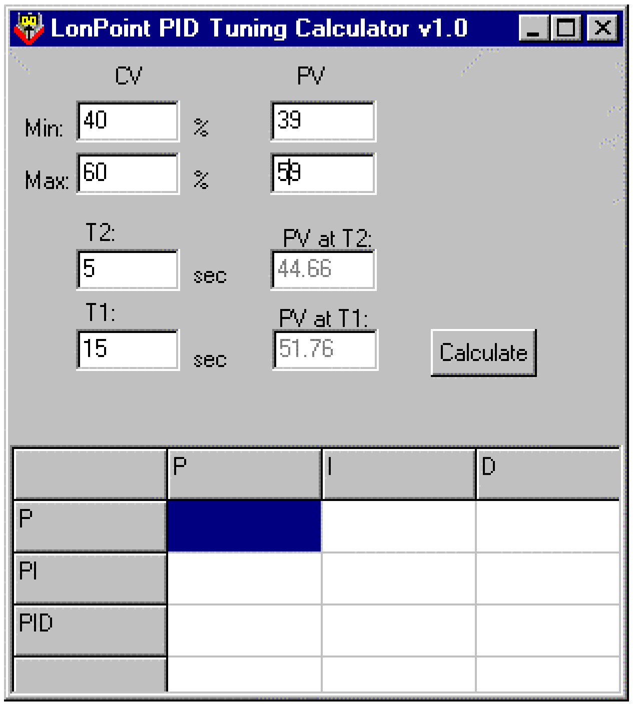
Subsystem	Device	Functional Block	Network Variable	Config Prop	Mon	Value
Subsystem 1	AO-1	PID-1	PV		YI	39.0977
Subsystem 1	AO-1	PID-1	Auto_Man		N	0.0 0
Subsystem 1	AO-1	PID-1	Man_Value		N	40

9. Change the `Man_Value` network variable to a new value. This example uses 60.
10. Wait for the process to stabilize and record the data. For this example the results are 60% and 63.11%.
11. Change `Man_Value` back the original setting, 40% for this example. The process will stabilize.

The LonPoint PID Tuning Calculator calculates the PID coefficients from the raw data. Once the `Man_Value` step is done, the initial and final process variable readings are known. This gives the static process gain; we know how much the process variable changes for a given change in the control variable. The process variable delta, `delta PV`, is the difference between the final and initial process variable readings; the control variable step causes a process variable change of `delta PV`.

In the next part of the experiment, the dynamic properties of the process are found. There are two time intervals that need to be measured. They are named T1 and T2. Both intervals start when the control variable step change is made. T2 is the time required to attain 28.8% of delta PV. T1 is the time required to reach 63.2% of delta PV. These specific percentage values are dictated by the two-point tuning method used in this example.

To start the LonPoint PID Tuning Calculator, open the Windows Start menu, select the LonPoint Device Software program folder, then select PID calculator. Enter the CV and PV data as shown below. Make a note of the PV at T2 and PV at T1 values. This says that T2 is defined as the time it takes for the process variable to reach a value of 44.66. T1 is the time it takes to reach 51.76.



	CV		PV
Min:	40	%	39
Max:	60	%	59

T2:	5	sec
PV at T2:	44.66	
T1:	15	sec
PV at T1:	51.76	

	P	I	D
P			
PI			
PID			

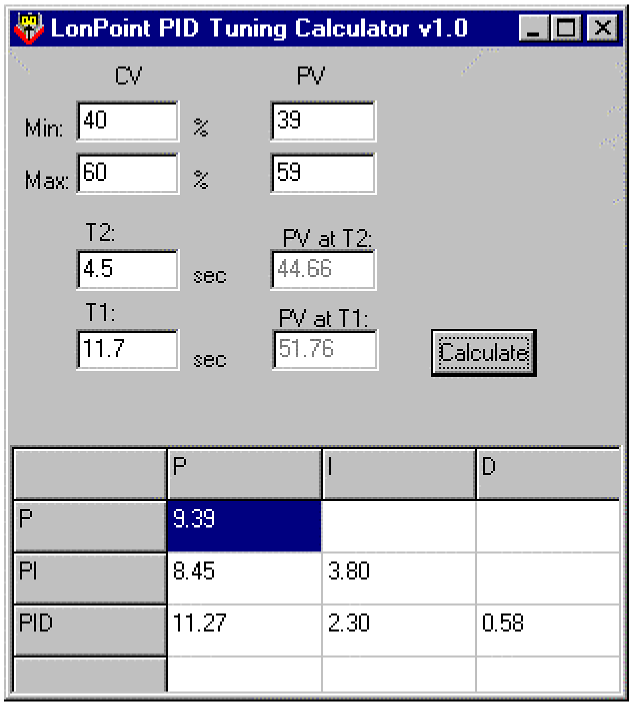
In this part, we will step the process again, watch for the process variable to reach the PV at T2 and PV at T1 values, and record the T1 and T2 times:

1. Using the LonMaker Browser, change the Man_Value network variable to the step value and start timing. This example uses a final step value of 60.

2. Watch the process variable, which is being monitored, to see when it reaches the calculated T2 value; 45 in this example.

3. Record the time and keep watching PV, record the time when the T1 value is reached; 52 in this example.

4. Enter the T1 and T2 times into the calculator and click the `Calculate` button. The PID coefficients are shown in the grid:



LonPoint PID Tuning Calculator v1.0

	CV		PV
Min:	40	%	39
Max:	60	%	59

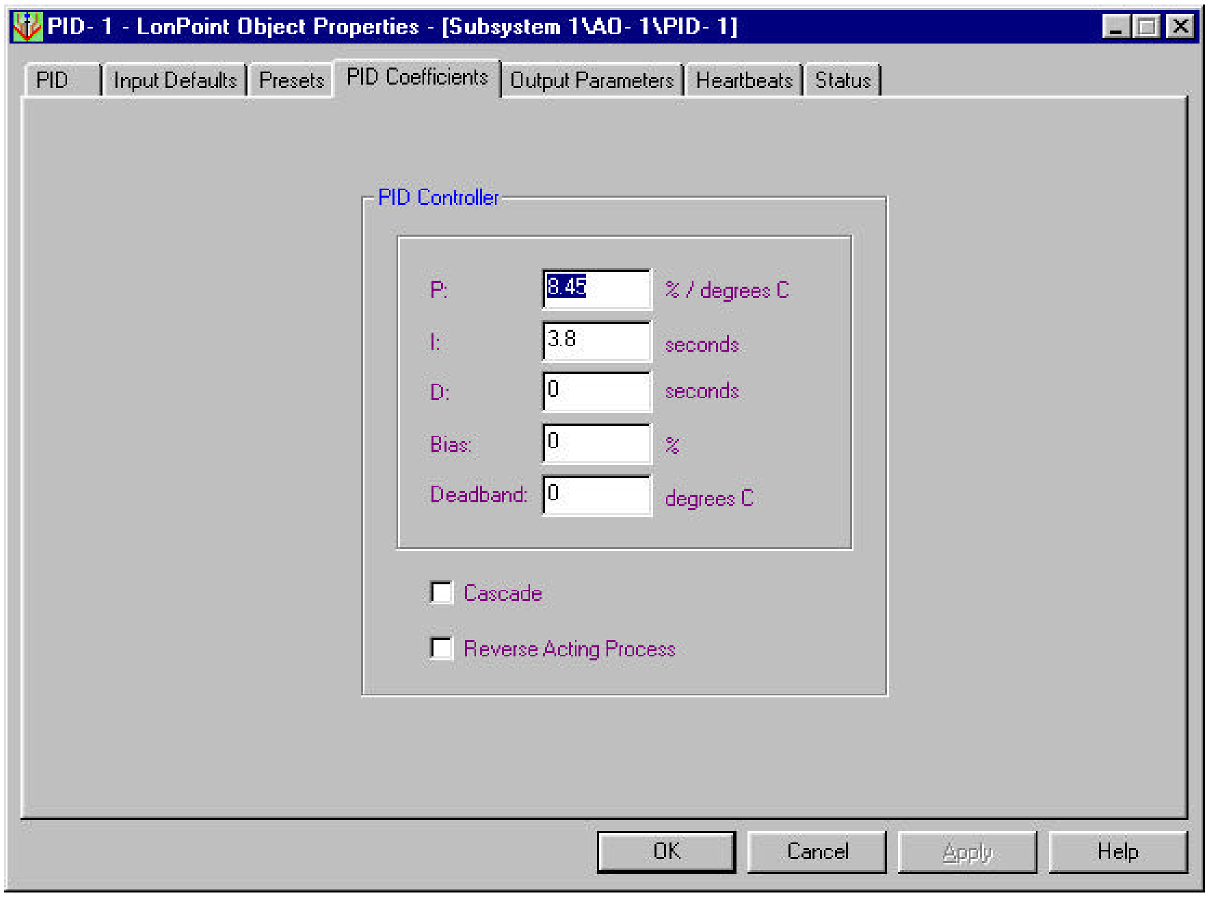
T2: 4.5 sec   PV at T2: 44.66

T1: 11.7 sec   PV at T1: 51.76

	P	I	D
P	9.39		
PI	8.45	3.80	
PID	11.27	2.30	0.58

The various options are shown in the left-hand column. Sets of coefficients for a P-only controller, a PI controller and a PID controller are shown. Blanks represent zero values for the associated parameters. For example, if you want a PI controller, set P = 8.45, I = 3.80, and D = 0.

Go back to the LonMaker tool to configure the PID. Right-click the PID Controller functional block and select *Configure* from the shortcut menu. Enter the PID coefficients using the PID coefficients tab as shown below.


PID- 1 - LonPoint Object Properties - [Subsystem 1\AO- 1\PID- 1]

PID Controller

P: 8.45 % / degrees C
I: 3.8 seconds
D: 0 seconds
Bias: 0 %
Deadband: 0 degrees C

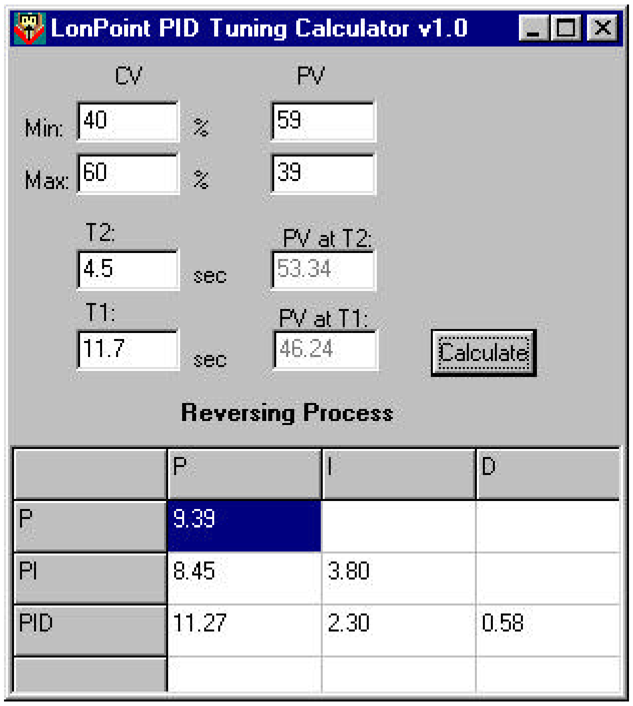
☐ Cascade
☐ Reverse Acting Process

A cascade arrangement is where an outer PID controller's control variable is used to drive the setpoint of an inner PID controller. The inner controller must react strongly to changes in its setpoint.

To make the PID controller react strongly to changes in setpoint, check Cascade – even if a cascaded PID is not being used. Normally, precise tracking of setpoint changes is less important that good tracking across process disturbances. Cascade is unchecked in this example.

An increase in the I-term gives less action. An increase in the D-term gives more action. A controller with more action will react more strongly to a given error.

The `Reverse Acting` process option is not used in this example. This is because the process is forward acting; an increase in the control variable causes an increase in the process variable. For a reversing process, the `Reverse Acting` process option must be selected, so the controller moves the control variable in the proper direction. If the process is reversing, the LonPoint PID Calculator will show a reversing process message as shown below:


LonPoint PID Tuning Calculator v1.0

	CV		PV
Min:	40	%	59
Max:	60	%	39

T2:		PV at T2:
4.5	sec	53.34

T1:		PV at T1:
11.7	sec	46.24

Reversing Process

	P	I	D
P	9.39		
PI	8.45	3.80	
PID	11.27	2.30	0.58

10

The Scheduler Functional Blocks: Application and the LonPoint Schedule Maker™

This chapter describes how to configure a LonPoint SCH-10 Scheduler Module containing Real-Time Clock, Event Scheduler, and State Machine functional blocks, using the LonPoint Schedule Maker utility. Chapter 11 describes the LonPoint Schedule Keeper utility that can be used by ad operator to modify an existing schedule.

Using the SCH-10 Scheduler Module

The SCH-10 Scheduler Module is a general-purpose controller that can be used as a source of system time, a time-based event scheduler, and a general-purpose state machine. The SCH-10 module includes a downloadable flash memory that is preprogrammed with the SCH-10 scheduler application. The SCH-10 module also includes a real-time clock with battery backup. The SCH-10 application can be used to control equipment startup, shutdown, and other control system operating procedures.

The SCH-10 application includes three functional blocks that provide scheduling and control functions. The LonPoint Schedule Maker utility is a Windows application that is used to create a supervisory application design that is downloaded to an SCH-10 module. These applications may be simple event-driven schedules, or complex state machines that control system behavior based on the interaction of many scheduled events and other network inputs. The LonPoint Schedule Maker utility includes a simulator that should be used to validate a supervisory application design before downloading the design to an SCH-10 module.

The SCH-10 hardware can be changed to a data logger by downloading the DL-10 application as described in Chapter 13.

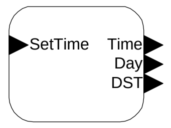
SCH-10 Functional Blocks

The SCH-10 application consists of the Real-Time Clock, Event Scheduler, and State Machine functional blocks.

The Real-Time Clock functional block, shown in the following figure, uses a battery-backed clock in the SCH-10 module to provide the following network variable outputs:

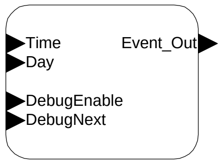
Time	The current time and date.
Day	The current day of the week.
DST	The daylight savings time flag. If set to TRUE, daylight savings time is in effect (i.e. during winter).

These outputs may be used by any device on the network, but are typically connected to the Event Scheduler functional block.



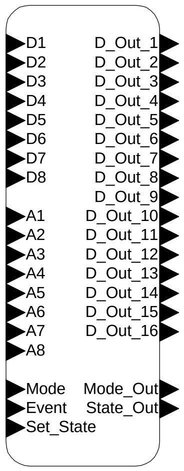
RTC-1

The Event Scheduler functional block, shown in the following figure, uses a time of day input from a Real-Time Clock functional block to generate scheduled events according to one of several 24-hour daily schedules. A default daily schedule can be defined for each day of the week, and override daily schedules can be defined for any specific date or range of dates. An entry in a daily schedule consists of a time (in hours and minutes) and an event. This event has a name and an On or Off value. For example, an event called evtEnableAlarm may be set to On at 7:00 PM and set to Off at 7:00 AM. When the time input to the Event Scheduler functional block equals the specified time for an event, the Event_Out output is set equal to the event. For the previous example, the Event_Out output of the Event Scheduler functional block would be set to `evtEnableAlarm On` at 7:00 PM and would be set to `evtEnableAlarm Off` at 7:00 AM. The Event Scheduler functional block also has two input network variables that are used to verify and debug operation of the schedule on the SCH-10 module during system commissioning.



ES-1

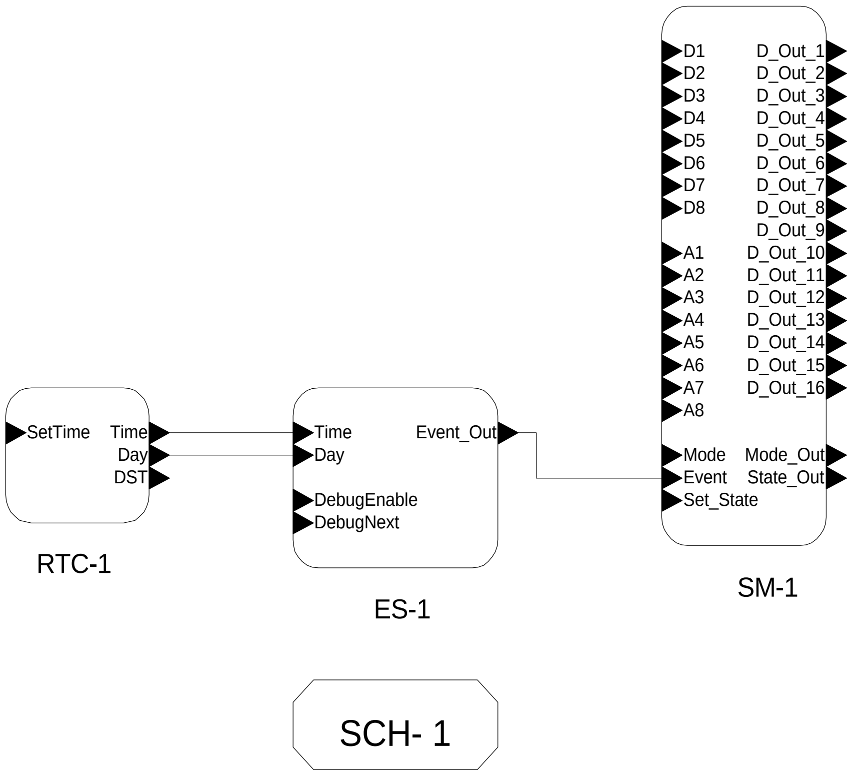
The State Machine functional block, shown in the following figure, maintains current state and calculates a next state based on the current state and inputs from up to 8 digital inputs, 8 analog inputs, a mode input, and a scheduled event input. The scheduled event input comes from an Event Scheduler functional block. The mode input may come from another State Machine functional block so that a system-wide State Machine functional block Mode output can be cascaded to multiple subsystem State Machine Mode inputs. Exit conditions can be defined for each state based on logical and arithmetic operations on the inputs to the State Machine. If an exit condition becomes true, the State Machine changes the current state, and sets up to 16 digital outputs and one mode output based on the last state and the selected exit condition. Many LonPoint functional blocks have mode inputs and can be configured to behave differently depending on what mode they are in.



SM-1

For example, in an air handler, a cndDayStart exit condition may be defined for the stNight state. The cndDayStart exit condition may be defined as the daytime scheduled event on, the smoke detector digital input off, and the room temperature analog input greater than 20ºC.

The following figure illustrates the typical configuration of the SCH-10 functional blocks in a LonMaker design.



Note: It is recommended that all changes to the SCH-10 functional blocks be made with the Schedule Maker and Schedule Keeper applications, not with the LonMaker Browser. If the LonMaker Browser must be used, reset the device after making changes. If the SCH-10 device is not reset, the SCH-10 device may not operate correctly.

LonPoint Schedule Maker

The LonPoint Schedule Maker utility is a Windows application for defining and simulating supervisory application designs, and downloading these designs to LonPoint SCH-10 modules. The LonPoint Schedule Maker utility allows designs to be saved to and loaded from supervisory application design files (.DSG extension) on your PC.

Unlike the other LonPoint configuration applications, the LonPoint Schedule Maker utility is not an LNS plug-in. To start this utility, click Start on the Windows task bar, open the Echelon LonPoint Device Software program folder, and select Schedule Maker.

Note:

Windows options must be configured to use Small Fonts in order to use this application. You can change this property in the *Settings* tab of the *Display* options in the Windows control panel.

Types of Schedules

The State Machine functional block can be used to implement two types of schedules – a digital output schedule and a state output schedule.

When using a *digital output schedule*, the digital outputs from the State Machine functional block are controlled directly by the Event_Out output from the Event Scheduler functional block. This type of schedule is useful for schedules that are strictly based on time of day, since the complexity of defining a state machine is eliminated.

When using a *state output schedule*, the state machine controls the digital outputs. This type of schedule is more complex to define, but allows the digital outputs to be controlled by combinations of time of day inputs as well as inputs from other digital and analog devices, as well as other LonPoint SCH-10 modules.

Overview of the Supervisory Application Development Process

Supervisory applications for the SCH-10 module should be designed in a LonMaker drawing before using the LonPoint Schedule Maker utility to define, download, and simulate the design. This simplifies the generation of as-built reports for the schedule design and significantly reduces engineering time.

A Schedule design stencil is included in the LonWorks `LonMaker\Visio` folder when the LonMaker software is installed. This stencil includes shapes that can be used to design a supervisory application within your LonMaker drawing. Open the LonMaker *File* menu, point to `Stencils`, then click `Open Stencil` or click the Open Stencil ( ) button on your drawing to open this stencil in your drawing. These shapes do not effect the function of the LonMaker network they are placed in. Place these shapes in a separate subsystem to document your schedule design within the network drawing.

To develop a supervisory application for a LonPoint SCH-10 module, follow these steps:

1. Define the sequence of operations
2. Create an event schedule design.
3. Create a state machine design.
4. Configure the Real-Time Clock functional block.
5. Configure the Event Scheduler functional block.
6. Configure the State Machine functional block.
7. Simulate the supervisory application design.
8. Download the design to an SCH-10 module.
9. Test the design in the SCH-10 module.

The first step is to create a text description of the sequence of operations for the supervisory application design. This description should be stored in your LonMaker drawing using a Visio text box. Steps 2 and 3 are completed within your LonMaker drawing using shapes from the Schedule stencil. You will manually transfer the schedule design that you create in these steps to the LonPoint Schedule Maker utility in steps 4 through 6. You will then simulate and download the design using the LonPoint Schedule Maker utility in steps 7 and 8, and test the design on an SCH-10 module in step 9. Steps 2 through 9 are described in more detail in the following sections.

Creating an Event Schedule Design

The event schedule design defines events based on time of day for a day of the week or a specified date. Schedules are defined as daily schedules that specify events for a 24-hour period, starting at midnight. Multiple daily schedules may be defined, each with a unique name. In the following examples, schedule names start with an *sch* prefix, so example schedule names are schOccupied and schUnoccupied. A default schedule may be assigned to each day of the week, and override schedules can be assigned to any range of dates.

To design an event schedule, follow these steps:

1. Define a daily schedule for each different set of events that occur over a 24-hour period.
2. Assign daily default schedules for each day of the week.
3. Assign override schedules for each range of dates that require a schedule other than the default daily schedule.

These steps are described in the following sections.

Defining Daily Schedules

A daily schedule defines times for On and Off events over a 24-hour period starting from midnight and continuing until 11:59PM. Up to 32 events may be defined for each daily schedule. Each event has a name. In the following examples, these names start with an *evt* prefix, so example events are evtOccupied On at 8:00 AM and evtOccupied Off at 7:00 PM.

To define a daily schedule, follow these steps:

1. Open the LonPoint Schedule stencil as described above.
2. Drag a Daily Schedule shape from the Schedule stencil to your LonMaker drawing.
3. Click the box to the right of the Schedule Name label. If the box is not selected, click it again.
4. Fill in a name for this daily schedule. Example schedule names are schOccupied and schUnoccupied.
5. For each event that will occur in the 24-hour period, fill in an event name, On or Off value, and time in the schedule.

6. Repeat steps 2 through 5 for each unique daily schedule. Assign a different schedule name to each.

Following is an example daily schedule.

Daily Schedule

Schedule Name:	schOccupied

Event Name	**ON or OFF**	**Time**
evtOccupied	Off	12:00 AM
evtWarmup	On	6:00 AM
evtWarmup	Off	8:00 AM
evtOccupied	On	8:00 AM
evtOccupied	Off	7:00 PM

1. Use an additional Daily Schedule shape for each 24 hour schedule.
2. Use a second Daily Schedule shape for additional events for the same schedule.
3. Event names must match Digital Output names for Digit Output Schedule designs.

Assigning Default Daily Schedules

Each day of the week must have a default daily schedule. Each day may have a different schedule, or you may assign the same schedule to multiple days. For example, Monday through Friday may have one schedule, and Saturday and Sunday may have a different schedule.

To define default daily schedules, follow these steps:

1. Drag a Default Daily Schedule shape from the Schedule stencil to your LonMaker drawing.

2. Fill in daily schedule names that you defined in *Defining Daily Schedules* to each day of the week. Fill in an optional description for each. The optional description is useful for as-built reports.

Following is an example default schedule.

Default Daily Schedule

Day	Daily Schedule Name	Optional Description
Sunday	schUnoccupied	
Monday	schOccupied	
Tuesday	schOccupied	
Wednesday	schOccupied	
Thursday	schOccupied	
Friday	schOccupied	
Saturday	schUnoccupied	

Assigning Override Daily Schedules

An override daily schedule may be assigned to any range of dates. Override schedules take precedence over default daily schedules, so they are useful for forcing special schedules on holidays and other special occasions.

To define override daily schedules, follow these steps:

1. Drag an Override Daily Schedule shape from the Schedule stencil to your LonMaker drawing.

2. For each range of dates to be overidden, fill in a start date, end date, and a daily schedule name that you defined in *Defining Daily Schedules* to each day of the week. Fill in an optional description for each. The optional description is useful for as-built reports.

Following is an example override daily schedule.

Override Daily Schedules

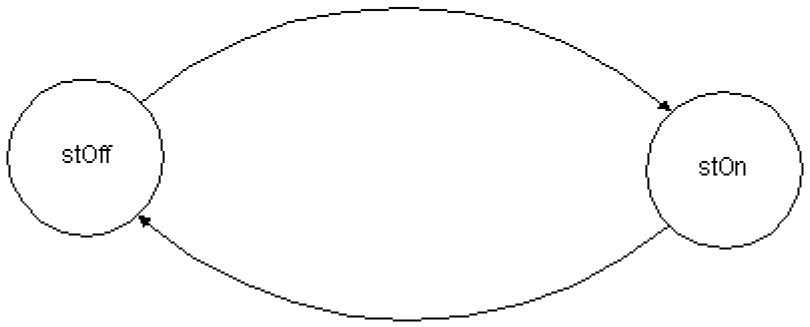
Date From	Date To	Daily Schedule Name	Optional Description
1/1/98	1/1/98	schUnoccupied	New Years Day
4/12/98	4/12/98	schUnoccupied	Easter
11/25/98	11/25/98	schUnoccupied	Thanksgiving
12/24/98	12/25/98	schUnoccupied	Christmas
12/31/98	12/31/98	schUnoccupied	New Years Eve

1. Use a second Override Daily Schedules shape for additional overrides.

Creating a State Machine Design

A state machine design defines one or more *states* for the State Machine functional block. In the following examples, state names start with an *st* prefix, so example state names are stOn and stOff. A simple system may have only these two states. An example of a more complex design is an HVAC system with additional states such as stOccupied, stUnoccupied, stWarmup, and stEmergency.

Each state has one or more *transitions* to other states. States and transitions are designed using a bubble diagram, with bubbles for each state and an arrow for each transition. For example, the following figure is a simple two-state design with two transitions.



For each transition, you will exactly one *exit condition*. The exit conditions define the inputs that must occur to follow a particular transition, i.e., to change to a specified state from the current state. Exit conditions are defined as logical and arithmetic operations on the Digital, Analog, Mode, and Event inputs to the State Machine functional block. For each exit condition, you will also define the Mode_Out and Digital output network variable values. When the state machine evaluates an exit condition as true, the state machine will update the outputs to the specified values before transitioning to the specified next state.

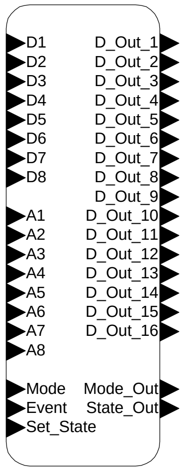
To design a state machine, follow these steps:

1. Define inputs to the state machine and outputs from the state machine.
2. Draw a state machine bubble diagram.
3. Define the exit conditions.
4. Define the state transitions.

These steps are described in the following sections.

Defining Inputs and Outputs

As shown in the following figure, the State Machine functional block has 8 digital input network variables, 8 analog input network variables, and 16 digital output network variables. These inputs and outputs may be connected to any other functional blocks in your system. To simplify using these inputs and outputs, you will assign names to each input or output that you plan to use. These names are only used in your schedule design, and are not downloaded to the SCH-10 module. In the following examples, input names start with an *i* prefix and output names start with an *o* prefix (except when using a digital output schedule as described in the next paragraph), so example names are iSystemEnable and oOccupied.



SM-1

When defining a digital output schedule, the digital output names must be the same as the event names used to control the outputs. This “wires” the event names to the digital outputs. For example, if the schedule defines evtOccupied On and Off events, and evtWarmup On and Off events, there must be digital outputs named evtOccupied and evtWarmup.

In addition to the digital inputs, the State Machine functional block has a mode input and output, an event input, and an internal power loss input. These additional inputs and outputs cannot be renamed.

To define input and output names, follow these steps:

1. Drag an Digital Inputs shape from the LonPoint Scheduler stencil to your LonMaker drawing.
2. Double-click the Digital Inputs shape. A drawing of the shape opens in a new window.
3. Assign a name to each of the digital input network variables that you plan to connect to other functional blocks. Fill in an optional description for each. The optional description is useful for as-built reports.
4. Close the Digital Inputs shape drawing window.
5. Repeat steps 1 through 4 for your analog input network variables using the Analog Inputs shape.
6. Repeat steps 1 through 4 for your digital output network variables using the Digital Outputs shape. The names for the digital outputs must be the same as your event names if your digital outputs will be controlled directly by the

Event Scheduler functional block, i.e., if you are using a digital output schedule.

Following are example digital and analog input name shapes with example names and descriptions.

Analog Inputs

	Analog Input Names	Optional Description and Type
A1	iReturnAirTemperature	Return air temperature
A2	iOutsideAirTemperature	Outside air temperature
A3		
A4		
A5		
A6		
A7		
A8		

Digital Inputs

	Digital Input Names	Optional Description and Type
	PowerLoss	
D1	iSystemEnable	System enable switch
D2	iEmergency	From fire panel
D3	iOccupiedOverride	Occupied override switch
D4		
D5		
D6		
D7		
D8		

Following is an example of the Digital Outputs shape for a digital output schedule, with example output names and descriptions. If you are designing a digital output schedule without state machine control of the mode output, this completes the scheduler design, so you can skip the remaining design steps and go to the section titled *Starting and Exiting the LonPoint Schedule Maker Utility*.

Digital Outputs

	Digital Output Names	Optional Description and Type
D_Out_1	evtOccupied	For setpoint and lighting control
D_Out_2	evtWarmup	Morning warmup control
D_Out_3		
D_Out_4		
D_Out_5		
D_Out_6		
D_Out_7		
D_Out_8		
D_Out_9		
D_Out_10		
D_Out_11		
D_Out_12		
D_Out_13		
D_Out_14		
D_Out_15		
D_Out_16		

Following is an example of the Digital Outputs shape for a state output schedule, with example names and descriptions.

Digital Outputs

	Digital Output Names	Optional Description and Type
D_Out_1	oOccupied	For setpoint control
D_Out_2	oSystemEnable	System enable
D_Out_3		
D_Out_4		
D_Out_5		
D_Out_6		
D_Out_7		
D_Out_8		
D_Out_9		
D_Out_10		
D_Out_11		
D_Out_12		
D_Out_13		
D_Out_14		
D_Out_15		
D_Out_16		

Drawing a State Machine Bubble Diagram

A state machine bubble diagram defines one or more *states* and *transitions* for the State Machine functional block. States are shown as bubbles on the diagram, and transitions are shown as arrows between the states.

You will configure one of the states as the *initial state.* This state is the first state selected by the State Machine functional block the first time a state machine is executed.

States do not generate outputs, they instead define a set of exit conditions to be monitored by the State Machine functional block. When an exit condition from the current state becomes true, the State Machine functional block changes to the next state defined for the transition associated with the exit condition, and starts to evaluate the exit conditions for the new state. The selected transition determines the outputs from the State Machine functional block.

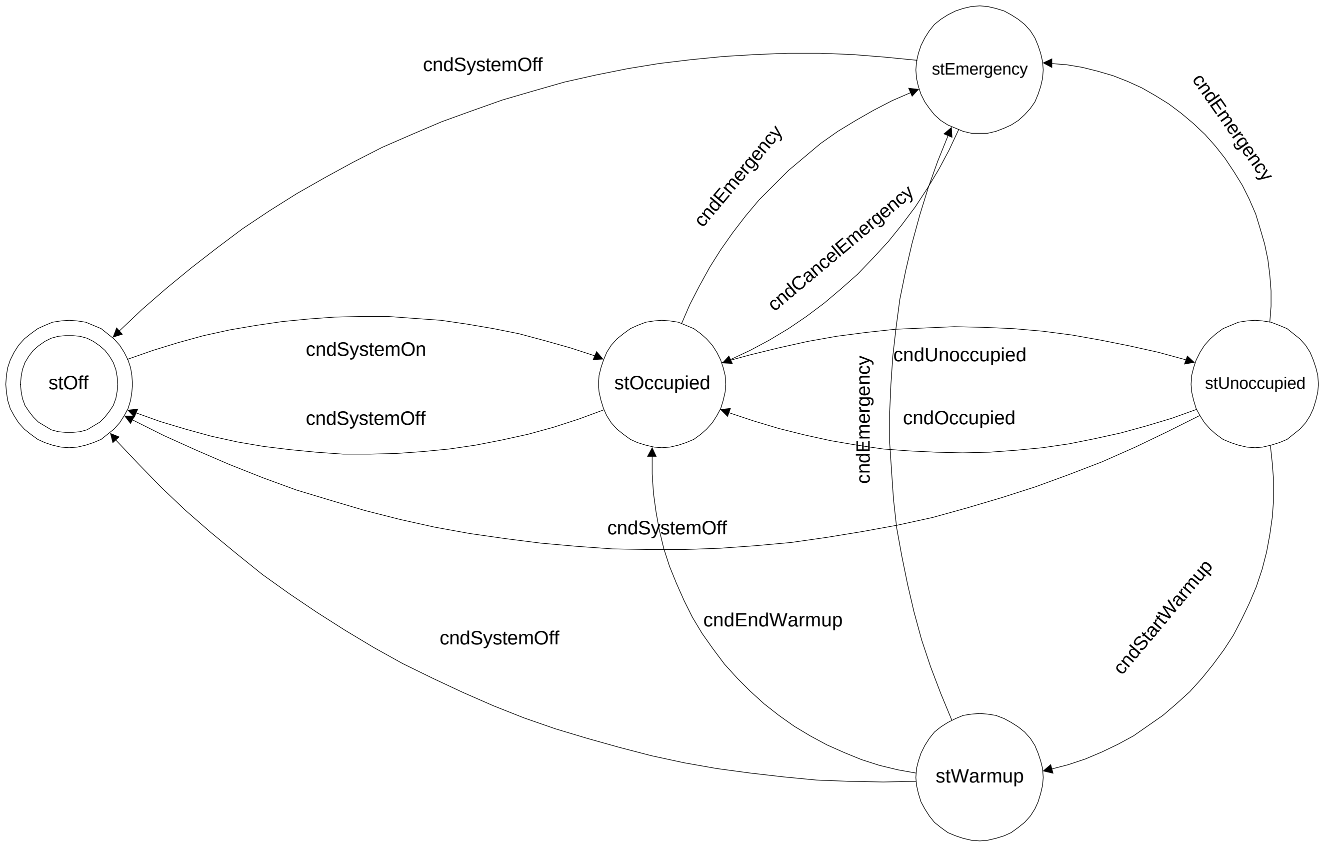
For example, you may have one transition between stOff and stOccupied that occurs when the system is enabled and the outdoor air temperature is hotter than the indoor air temperature. This transition may put the system into cooling mode. A second transition between stOff and stOccupied may occur when the system is enabled and the outdoor air temperature is cooler than the indoor air temperature. This transition may put the system into heating mode.

For each transition, you will define an *exit condition*. An exit condition is a formula that specifies logical and arithmetic operations on the State Machine inputs. When the State Machine functional block executes, it evaluates each of the exit conditions for the transitions defined out of the current state. The first exit condition that evaluates to true triggers the State Machine functional block to change states based on the transition being evaluated. The same exit condition may be used for multiple transitions, as long as each originates from a different state.

To draw a bubble diagram, follow these steps:

1. Drag an Initial State shape from the LonPoint Schedule stencil to your LonMaker drawing.
2. Type the name of your initial state, for example stOff.
3. For each of your remaining states, drag a State shape from the LonPoint Schedule stencil to your LonMaker drawing. Type a name for each state after you drag it to the drawing. In the following examples, state names start with an *st*. Example names are stOn, stOff, stNormal, stOccupied, stWarmup, and stEmergency.
4. For each transition, drag a Transition shape from the Schedule stencil to your LonMaker drawing. Drag the end of the Transition shape without an arrow to the center of the starting state for a transition, then drag the end of the transition shape with an arrow to the center of the ending state.
5. Click on each Transition shape and type an exit condition name. In the following examples, exit condition names start with a *cnd* prefix. Example exit condition names are cndStartWarmup, cndEmergency, and cndSystemOff. The same condition exit condition name may be used on multiple transitions.

Following is an example bubble diagram. This diagram defines five states and the transitions between them.



Defining State Transitions

A *state transition* defines the rules for changing from one state to a second state, and also defines the outputs from the State Machine functional block when the transition is followed. The State Machine functional block continues to send the defined outputs on the network at the configured heartbeat rate until the next transition is followed.

For each state transition, you will specify an *exit condition*. An exit condition is an expression that defines the rules for changing to the specified next state. You will define these expressions in the next section. You must define an exit condition for every transition defined in your state machine bubble diagram; however, the same exit condition may be used for multiple transitions as long as the transitions originate from different states. If multiple exit conditions become true at the same time, only the first exit condition that evaluates to true will affect the state machine.

For each state transition, you will also specify the network variable outputs from the State Machine functional block for the transition. You will specify values for the Mode_Out network variable, and the D_Out_1 through D_Out_16 digital output network variables.

To define state transitions, follow these steps for each state that you defined in the previous section:

1. Drag a State Definitions shape from the LonPoint Schedule stencil to your LonMaker drawing.

2. Double-click the State Definitions shape. A drawing of the shape opens in a new window.

3. Click on the box to the right of the State Name label.

4. Fill in the name of the state from the bubble diagram.

For each transition out of the state that you defined in the state machine bubble diagram, follow these steps:

1. Click an entry in the Exit Condition column and type an Exit Condition name. In the following examples, Exit Condition names start with a *cnd* prefix, so example Exit Condition names are cndOccupied and cndStartWarmup.

2. Click the corresponding entry in the Next State column and type the name of the next state for the transition. This is the state at the head of the arrow in your state machine bubble diagram.

3. Click the corresponding entry in the Mode_Out column and type a value for the Mode_Out network variable. By default, the Mode_Out network variable is an SNVT_hvac_mode type, but you can use the LonMaker Browser to change the type of this output to any single-byte output type. The value that you will enter in the LonPoint Schedule Maker application for the output is always specified as a SNVT_hvac_mode value, but the actual type of the output will be the type that you specify in the LonMaker Browser. See *Changing the Mode and Mode_Out Types* later in this chapter. Following is a table of output values for the Mode_Out network variable. If you are using the SNVT_hvac_mode type, select the HVAC_mode name for each output. If you are using another type, find the numeric value that corresponds to your desired output, and enter the corresponding SNVT_hvac_mode name.

Value	***Name***	***Notes***
0	HVAC_AUTO	Controller automatically changes between application modes
1	HVAC_HEAT	Heating only
2	HVAC_MRNG_WRMUP	Application-specific morning warmup
3	HVAC_COOL	Cooling only
4	HVAC_NIGHT_PURGE	Application-specific night purge
5	HVAC_PRE_COOL	Application-specific pre-cool

6	HVAC_OFF	Controller not controlling outputs
7	HVAC_TEST	Equipment being tested
8	HVAC_EMERG_HEAT	Emergency heat mode (heat pump)
9	HFAC_FAN_ONLY	Air not conditioned, fan turned on
255	HVAC_NUL	Value not available

4. Click the corresponding entry in the Digital Output Values column and type the names of all the digital output network variables that should be on after this transition. The digital outputs are all SNVT_switch values, and you will specify each as either On or Off.

When you are finished defining a State Definition shape, close the State Definitions shape drawing window.

Following is an example State Definitions shape. This shape defines the state transitions for the stUnoccupied state of the example described in *Drawing a State Machine Bubble Diagram* earlier in this chapter.

State Transition Definitions

State Name:	stUnoccupied

Exit Condition	**Next State**	**Mode_Out Value**	**Digital Output Values**
cndOccupied	stOccupied	HVAC_AUTO	oOccupied On oSystemEnable On
cndStartWarmup	stWarmup	HVAC_MRNG_WRMUP	oOccupied On oSystemEnable On
cndEmergency	stEmergency	HVAC_EMERG_HEAT	oSystemEnable On
cndSystemOff	stOff	HVAC_OFF	

1. Use an additional Transition Definitions shape for each State.
2. Use up to 4 Transition Definitions shapes per state.

Defining Exit Conditions

An exit condition is a specification for the conditions under which a transition should be followed in a state machine. An *exit condition* is defined as a combination of events from the Event Scheduler functional block and the values of the State Machine functional block's input network variables.

An exit condition is defined as a logical expression consisting of multiple terms separated by AND and OR. Each term is an expression with a True or False result. There are four types of terms: analog, mode input, digital input, and scheduler. Following is a description of each type:

- An analog term is an expression based on arithmetic operations and comparisons on the analog network variable inputs to the State Machine functional block. The network variable inputs are defined on your Analog Inputs shape as described earlier in this chapter. An analog term may include any number of the analog inputs, floating point numbers, and one comparison function.

  For example, an analog term is defined to return True if the average temperature from two temperature sensors is equal to or above 25º C. The term would be defined as the following expression:

  ```
  ((iTempSens1 + iTempSens2)/2) >= 25
  ```

  Note: *Operations on analog inputs are performed on the raw analog data in units native to the network variable. For example, operations on a variable of type `SNVT_temp_f` will always use units of "degrees Celsius," even if the system has been configured to display data in U.S. units.*

- A mode input term compares the Mode input network variable to one of its possible values. The expression can specify that the Mode input "Is" or "Is Not" equal to a possible Mode value. Following are two examples:

  Mode IS HVAC_AUTO

  Mode IS NOT HVAC_WARMUP

- A digital input term specifies that one of the digital inputs to the State Machine functional block is equal to ON or OFF. The digital inputs are declared as SNVT_Switch inputs. A SNVT_Switch input has two fields, a value expressed as a percentage, and a state expressed as a 0 or 1. If either field is 0, the input value is considered OFF, otherwise it is considered ON. Following are two examples:

  iSystemEnable ON

  iEmergency OFF

- A scheduler input term specifies an event defined for the Event Scheduler functional block. Following are two examples:

  evtOccupied ON

evtWarmup OFF

To define exit conditions, follow these steps:

1. Drag a Condition Definitions shape from the LonPoint Scheduler stencil to your LonMaker drawing.
2. Double-click the Condition Definitions shape. A drawing of the shape opens in a new window.
3. For each exit condition definition, click an entry in the Exit Condition column and type an Exit Condition name. Then click an entry in the Definition column and type an Exit Condition expression. Close the Exit Condition shape drawing window.

Following is an example Condition Definitions shape. This shape defines the exit conditions for the example described in *Drawing a State Machine Bubble Diagram* earlier in this chapter.

Exit Condition Definitions

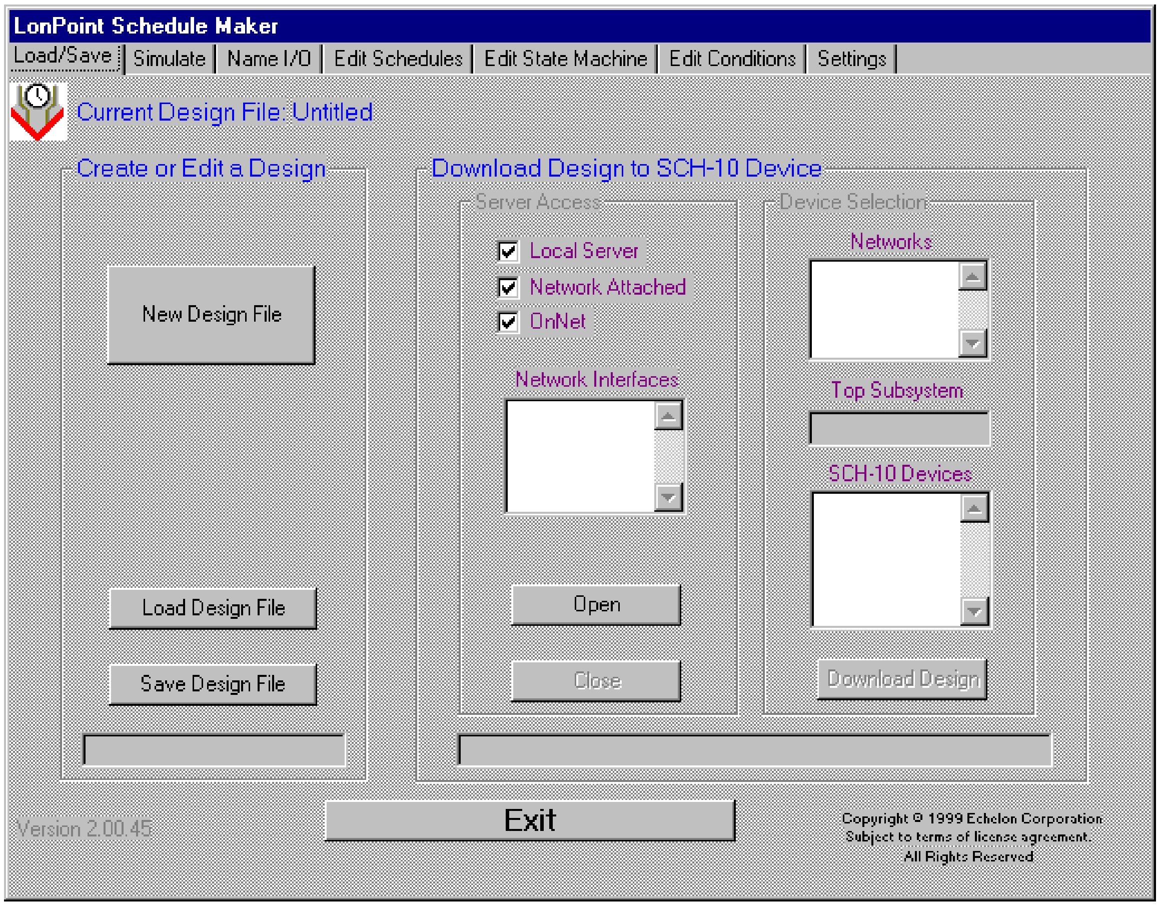
Exit Condition	Definition
cndSystemOn	iSystemEnable On
cndSystemOff	iSystemEnable Off
cndOccupied	(iOccupiedOverride On) OR (Powerloss On)
cndUnoccupied	(evtOccupied Off) AND (iOccupiedOverride Off) AND (evtWarmup Off)
cndStartWarmup	evtWarmup On
cndEndWarmup	(evtWarmup Off) OR (iReturnAirTemperature > 16) OR (Powerloss On)
cndEmergency	iEmergency On
cndCancelEmergency	(iOccupiedOverride On) OR (Powerloss On)

1. Use an additional Exit Condition Definitions shape for additional definitions.
2. **Always** use default SI units for constants.

Starting and Exiting the LonPoint Schedule Maker Utility

Once a schedule design is complete, the design must be manually transferred to the LonPoint Schedule Maker utility. LonPoint Schedule Maker is a stand-alone Windows application that is used to create, edit, and save schedule design files. LonPoint Schedule Maker is also used to simulate designs, and to download completed designs to SCH-10 modules.

To start the utility, click the Start menu on the Windows taskbar, open the LonPoint Device Software program folder, and select Schedule Maker. The following window opens:

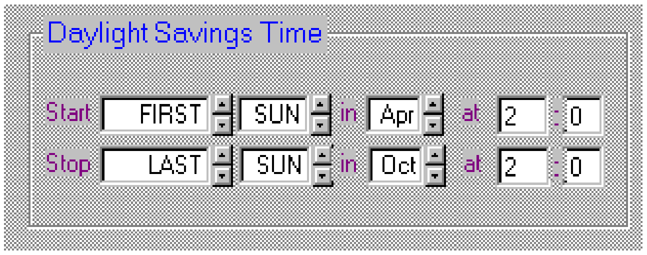


To exit, click the Exit button on the Load/Save tab.

Configuring the Real Time Clock Functional Block

The Real-Time Clock functional block reports the current date, day, and time as maintained by a battery-backed real-time clock on the SCH-10 module. The only configuration required is to define the starting and ending times for Daylight Savings Time.

To configure Daylight Savings Time, select the Settings tab and set the starting and ending dates/times in the following fields:



Start FIRST SUN in Apr at 2 : 0

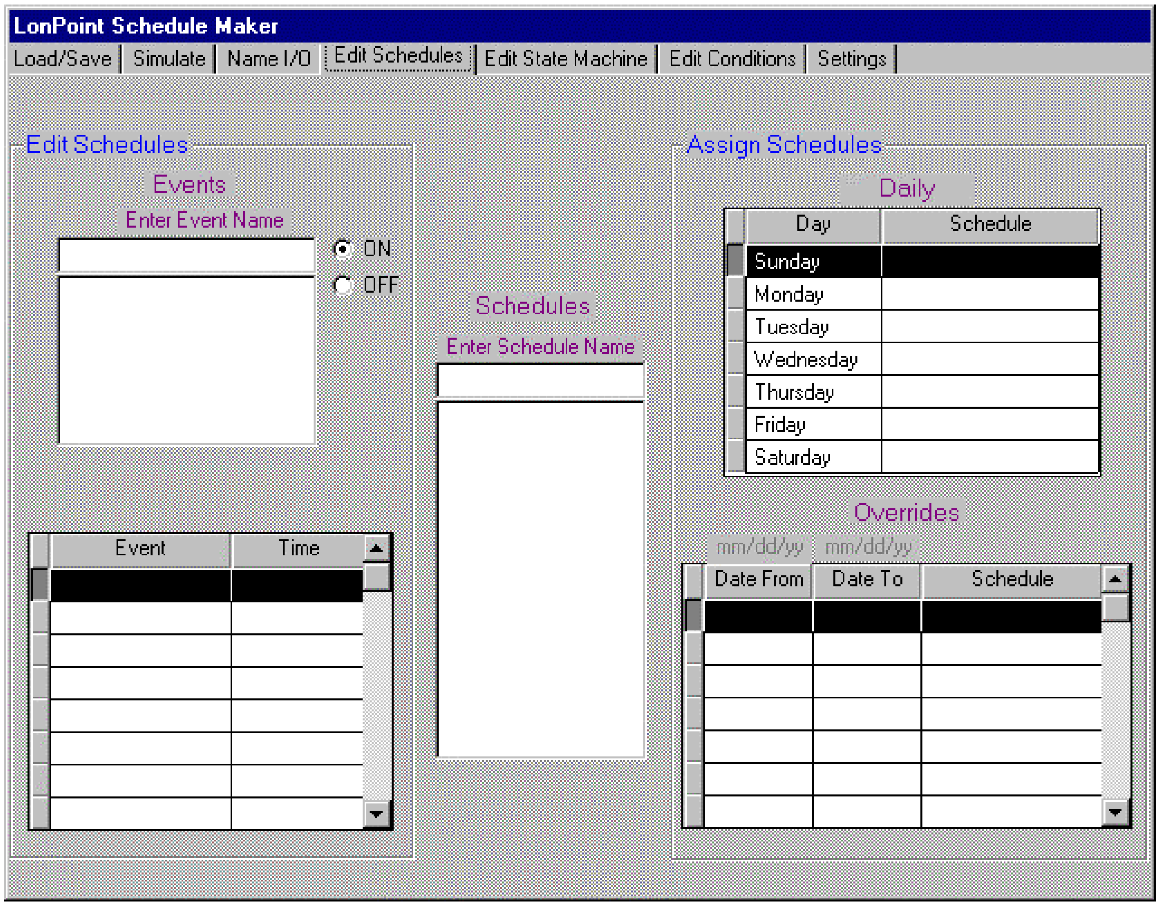
Stop LAST SUN in Oct at 2 : 0

Set the starting and ending dates and times to the same value to disable Daylight Savings Time.

The first time you use an SCH-10 module, you will also need to set the time using the LonMaker Browser. This procedure is described in *Setting the SCH-10 Time* later in this chapter.

Configuring the Event Scheduler Functional Block

The Event Scheduler functional block generates time-based events that are used by the State Machine functional block. It is configured with an event schedule design as described in *Creating an Event Schedule Design* earlier in this chapter. To configure the Event Scheduler functional block, you will manually transfer your event schedule design to the LonPoint Schedule Maker application using the Edit Schedules tab shown here:



Day	Schedule
Sunday	
Monday	
Tuesday	
Wednesday	
Thursday	
Friday	
Saturday	

The following section describes how to transfer the daily schedules, default daily schedules, and override daily schedules to the Edit Schedules tab.

Configuring Daily Schedules

A daily schedule defines times for On and Off events over a 24-hour period starting from midnight and going to 11:59PM. Daily schedules are described in *Defining Daily Schedules* earlier in this chapter. To transfer a daily schedule design to Schedule Maker, follow these steps:

1. Select the Edit Schedules tab.
2. Click the Enter Schedule Name field.
3. For each of your Daily Schedule shapes, enter the schedule name and press the Enter key.
4. Click the Enter Event Name field.
5. For each event name that you used in your Daily Schedule shapes, enter the event name and press the Enter key.

Warning: Schedule names must be entered before defining events and daily schedules.

For each of your Daily Schedule shapes, click the schedule name in the Schedules list, then for each line in the Daily Schedule shape, follow these steps:

1. Select the ON or OFF button in the Events list to determine whether an On event or an Off event will be generated at the specified time.
2. Click the event name in the Events list to be generated. This will cause the event to be added to the Daily Schedule.
3. Click the Time field for the event, enter the time that the specified event should be in `hh:mm:ss AM/PM` format, where `ss` is set to `00`, and press the Enter key.

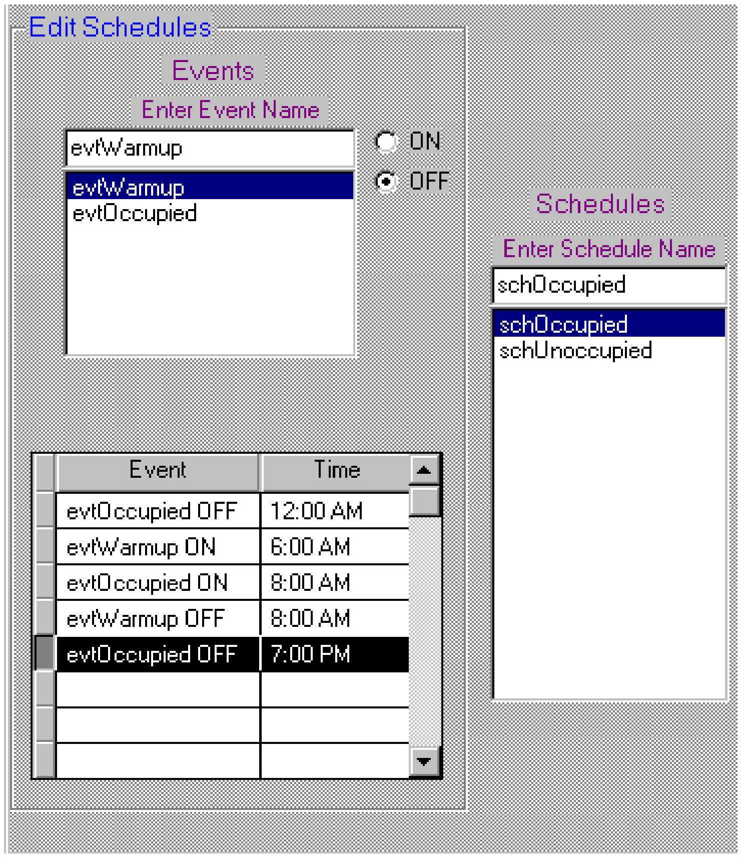
To delete a daily schedule, follow these steps:

1. Click the name of the daily schedule in the Schedules list.
2. When the selected daily schedule name is displayed in the Enter Schedule Name field, right-click it.
3. Select the Delete command from the shortcut menu.

To delete an event, follow these steps:

1. Click the name of the event in the Events list.
2. When the selected event name is displayed in the Enter Event Name field, right-click it.
3. Select the Delete command from the shortcut menu.

The following figure shows a daily schedule design for the schOccupied schedule design described in *Defining Daily Schedules*.



Event	Time
evtOccupied OFF	12:00 AM
evtWarmup ON	6:00 AM
evtOccupied ON	8:00 AM
evtWarmup OFF	8:00 AM
evtOccupied OFF	7:00 PM

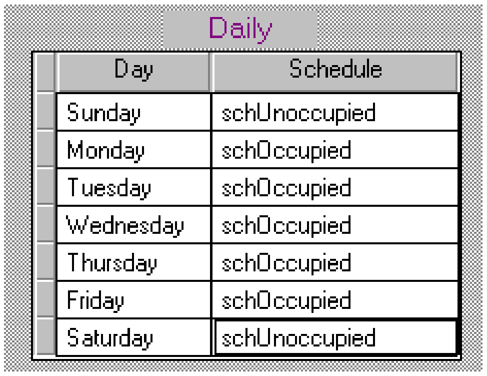
Configuring Default Daily Schedules

Default daily schedules specify a daily schedule to be used by default for each day of the week. These defaults may be overridden for any range of dates as described in the next section.

Default daily schedules are described in *Assigning Default Daily Schedules* earlier in this chapter. To transfer a default daily schedule design to the LonPoint Schedule Maker utility, select the Edit Schedules tab. Then, for each day of the week, follow these steps:

1. Click the schedule name to be used for the day of the week in the Schedules list.
2. Press Ctrl-C to copy the schedule name to the clipboard.
3. Click the Schedule column entry to the right of the day of the week in the Daily Defaults table.
4. Press Ctrl-V to type the schedule name in the Schedule column entry.

The following figure shows a default daily schedule for the schedule design described in *Assigning Default Daily Schedules*.



Day	Schedule
Sunday	schUnoccupied
Monday	schOccupied
Tuesday	schOccupied
Wednesday	schOccupied
Thursday	schOccupied
Friday	schOccupied
Saturday	schUnoccupied

Configuring Override Daily Schedules

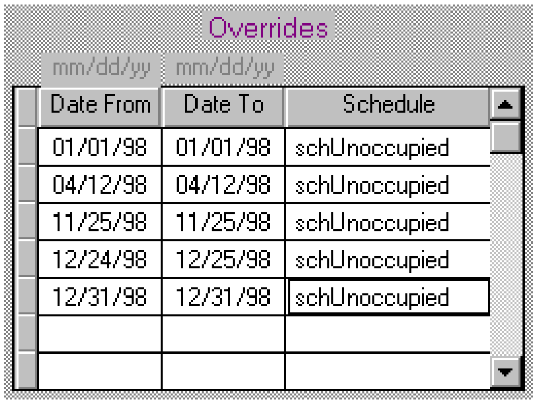
Override daily schedules specify a daily schedule to be used for a specified range of dates. Override daily schedules are described in *Assigning Override Daily Schedules* earlier in this chapter. To transfer an override daily schedule design to the LonPoint Schedule Maker utility, select the Edit Schedules tab. Then, for each range of dates to be overridden, follow these steps:

1. Click the schedule name to be used for the override schedule in the Schedules list.
2. Press Ctrl-C to copy the schedule name to the clipboard.
3. Click the next available entry in the Date From column.
4. Enter the Date From date in mm/dd/yy format. For the year entry, always enter the last two digits of the year. For example, enter 00 for the year 2000 or 01 for 2001.
5. Press the Tab key.
6. Enter the Date To date in mm/dd/yy format.
7. Press the Tab key.
8. Press Ctrl-V to enter the daily schedule name from the clipboard.
9. Press the Tab key to go to the next line.

To delete an override daily schedule, follow these steps:

1. Click on any entry for the line to be deleted in the Overrides list.
2. Right-click the same entry.
3. Select the Delete command from the shortcut menu.

The following figure shows the override schedules for the schedule design described in *Assigning Override Daily Schedules*.



Overrides

Date From (mm/dd/yy)	Date To (mm/dd/yy)	Schedule
01/01/98	01/01/98	schUnoccupied
04/12/98	04/12/98	schUnoccupied
11/25/98	11/25/98	schUnoccupied
12/24/98	12/25/98	schUnoccupied
12/31/98	12/31/98	schUnoccupied

Configuring the State Machine Functional Block

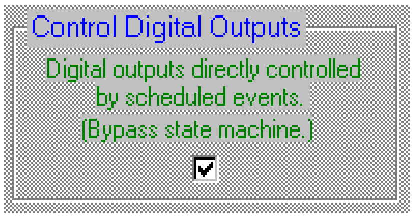
The State Machine functional block controls up to 16 digital outputs and a mode output based on transitions between states. Each transition has an exit condition that defines when the transition should be followed. States do not generate outputs, they instead define a set of exit conditions to be monitored by the State Machine functional block. When an exit condition from the current state becomes true, the State Machine functional block changes to the next state defined for the transition associated with the exit condition, and starts to evaluate the exit conditions for the new state. The selected transition determines the outputs from the State Machine functional block.

A state machine design consists of input and output names, a state machine bubble diagram, state transition definitions, and exit condition definitions as described in *Creating a State Machine Design* earlier in this chapter. You will manually transfer this design to the Name I/O, Edit State Machine, and Edit Conditions tabs as described in the following sections.

Configuring a Digital Output Schedule

As described in *Types of Schedules* earlier in this chapter, the digital outputs from the State Machine functional block can be controlled directly by the Event_Out output from the Event Scheduler functional block. This type of schedule is called a digital output schedule, and is useful for schedules that are strictly based on time of day.

To specify that you are using a digital output schedule, select the Settings tab, and enable the Control Digital Outputs option as shown in the following figure:



Control Digital Outputs

Digital outputs directly controlled by scheduled events. (Bypass state machine.)

☑

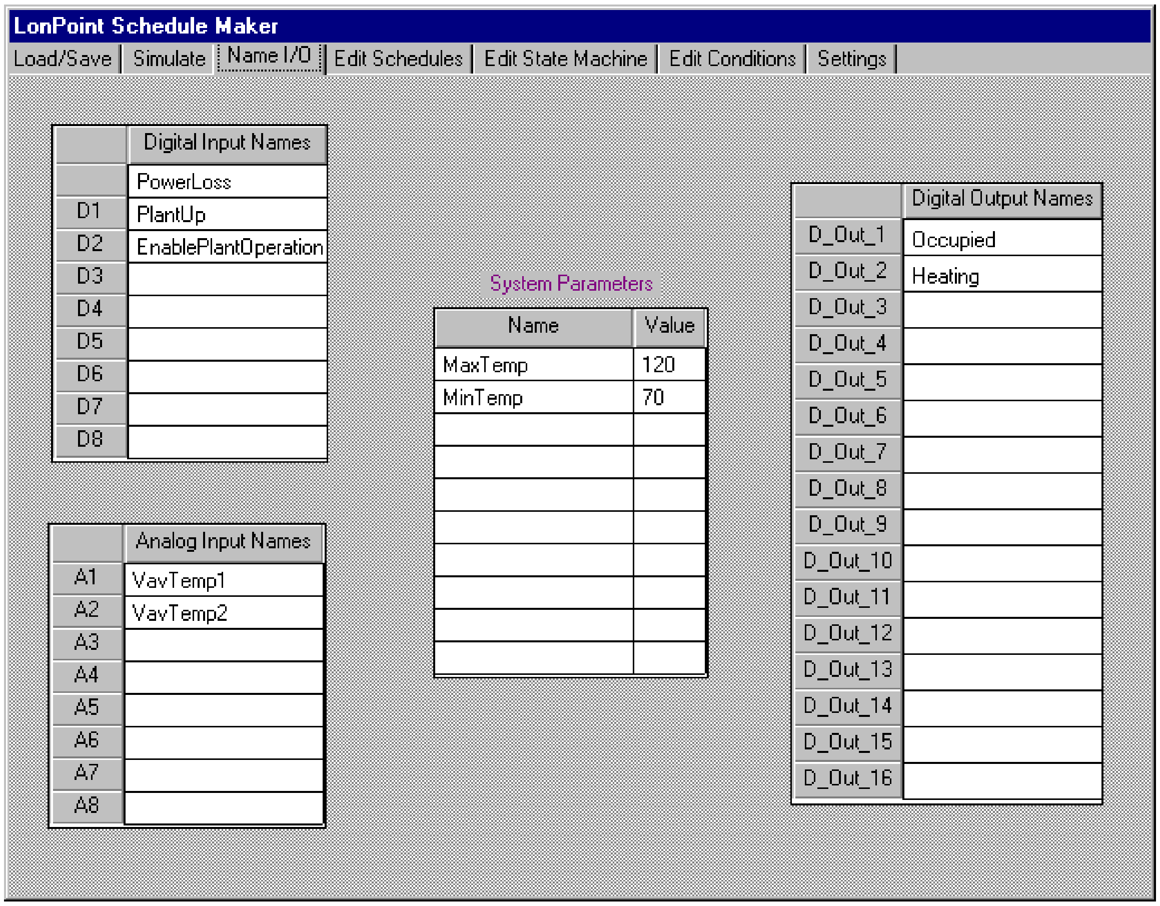
When this option is selected, the digital output names must match the event names as described in *Defining Inputs and Outputs* earlier in this chapter. The outputs must be named as described in *Assigning Input/Output Names* in the next section, and at least one state with one exit condition must be defined, even if you are not using the Mode_Out output from the State Machine functional block.

Configuring a State Machine

Three tabs are used to configure a state machine. The Name I/O tab is used to define names for each of the network variable inputs and outputs for the State Machine functional block; the Edit State Machine tab is used to define the states and transitions for the state machine; and, the Edit Conditions tab is used to define the exit conditions.

Assigning Input/Output Names

The Name I/O tab, shown in the following figure, allows you assign names to each of the State Machine functional block input and output network variables and to define System Parameters.



	Digital Input Names
	PowerLoss
D1	PlantUp
D2	EnablePlantOperation
D3	
D4	
D5	
D6	
D7	
D8	

	Analog Input Names
A1	VavTemp1
A2	VavTemp2
A3	
A4	
A5	
A6	
A7	
A8	

System Parameters

Name	Value
MaxTemp	120
MinTemp	70

	Digital Output Names
D_Out_1	Occupied
D_Out_2	Heating
D_Out_3	
D_Out_4	
D_Out_5	
D_Out_6	
D_Out_7	
D_Out_8	
D_Out_9	
D_Out_10	
D_Out_11	
D_Out_12	
D_Out_13	
D_Out_14	
D_Out_15	
D_Out_16	

This tab contains four tables that allow you to assign names to each of the 8 analog inputs, 8 digital inputs, and 16 digital outputs, and to define up to 10 system parameters.

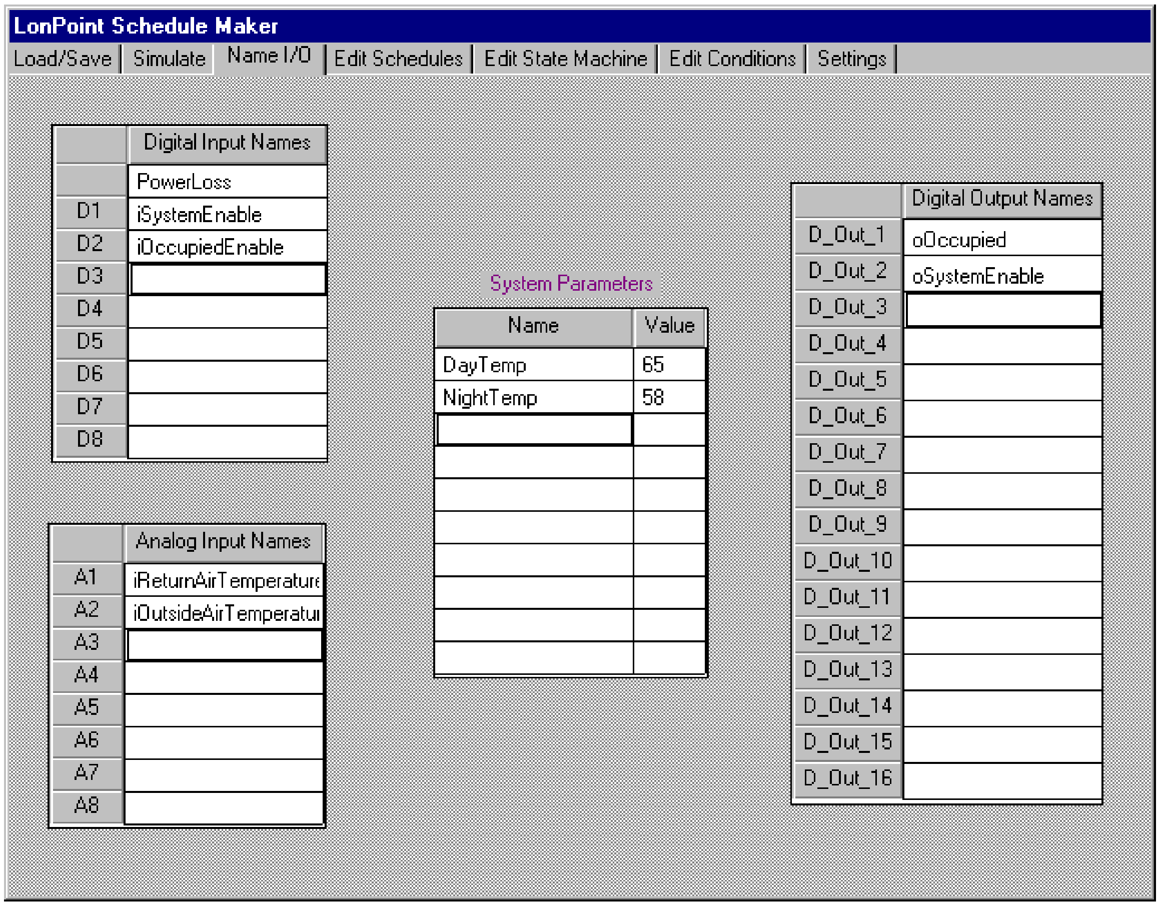
Only network variable names that will be used in other tabs need be assigned; the other names may be blank. These network variable names are used only within the LonPoint Schedule Maker utility to provide meaningful names for the network variables when used in exit condition definitions. They have no meaning within other applications such as the LonMaker tool, the LonMaker Browser, or the LonPoint plug-in. By default, these other applications will use the input and output names shown in the left column of the input and output tables, for example, "D1", "A1", and "D_Out_1".

The `System Parameters` table allows you to define up to 10 constants which may be used in the `Edit Conditions` tab. The values of these constants can be modified either in this tab or by using the LonPoint Schedule Keeper utility (see Chapter 11).

To assign names to the input and output network variables and system parameters, follow these steps:

1. Select the Name I/O tab.
2. Click the D1 input name field if your design includes any digital inputs.
3. For each digital input defined in your Digital Inputs shape, enter the digital input name and press Enter.
4. Click the A1 input name field if your design includes any analog inputs.
5. For each analog input defined in your Analog Inputs shape, enter the analog input name and press Enter.
6. Click the D_Out_1 output name field if your design includes any digital outputs.
7. For each digital output defined in your Digital Outputs shape, enter the digital output name and press Enter.
8. Click the first field in the System Parameters table if you wish to define one or more system parameters.
9. For each system parameter you wish to define, enter the name (which may contain only letters, digits, and the underscore character) and the initial value for the constant. The system parameters defined here will be available under `Analog Inputs` in the `Edit Conditions` tab.

The following figure shows the input/output names for the schedule design described in *Defining Inputs and Outputs* earlier in this chapter as well as several named constants.



	Digital Input Names
	PowerLoss
D1	iSystemEnable
D2	iOccupiedEnable
D3	
D4	
D5	
D6	
D7	
D8	

	Analog Input Names
A1	iReturnAirTemperature
A2	iOutsideAirTemperatui
A3	
A4	
A5	
A6	
A7	
A8	

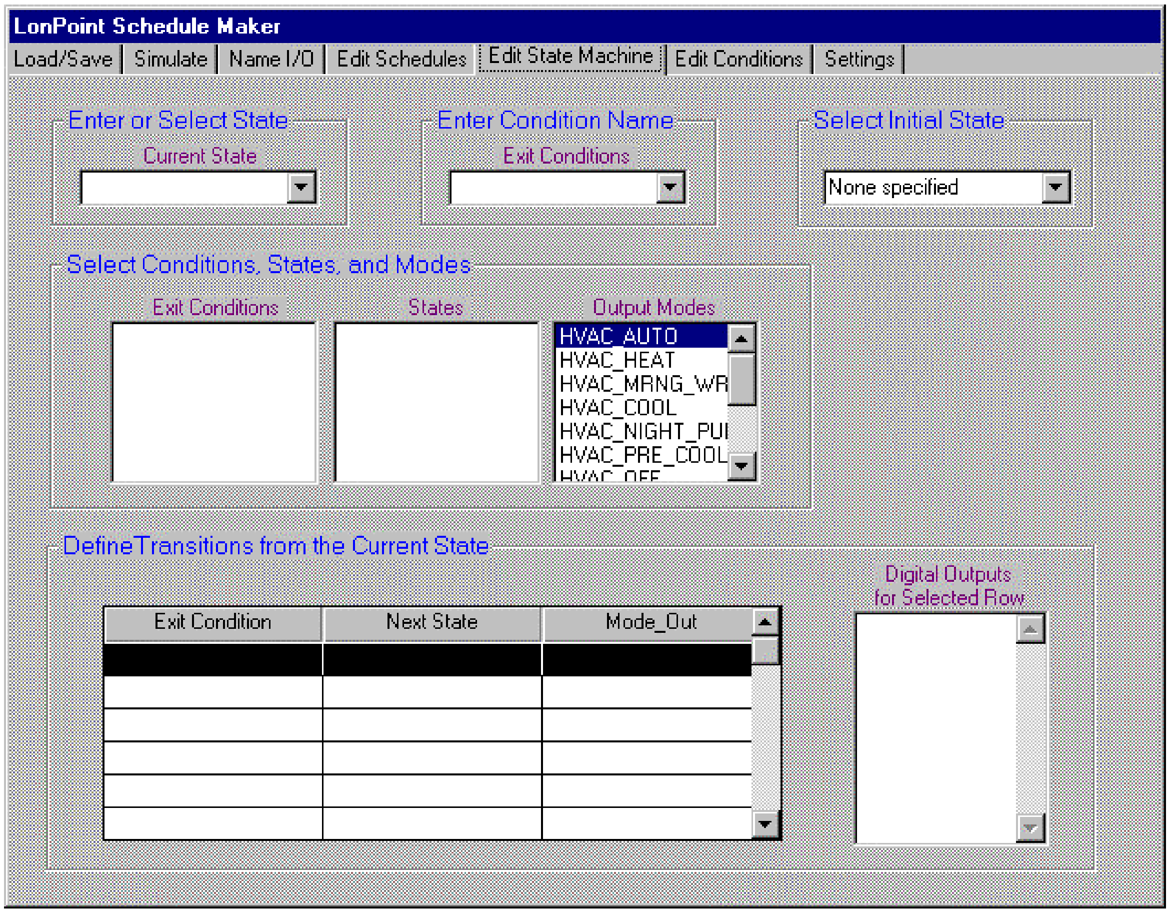
System Parameters

Name	Value
DayTemp	65
NightTemp	58

	Digital Output Names
D_Out_1	oOccupied
D_Out_2	oSystemEnable
D_Out_3	
D_Out_4	
D_Out_5	
D_Out_6	
D_Out_7	
D_Out_8	
D_Out_9	
D_Out_10	
D_Out_11	
D_Out_12	
D_Out_13	
D_Out_14	
D_Out_15	
D_Out_16	

Defining the State Machine

The Edit State Machine tab, shown in the following figure, allows you to define the state names and transitions for the State Machine functional block.



Select Initial State: None specified

Output Modes: HVAC_AUTO, HVAC_HEAT, HVAC_MRNG_WR, HVAC_COOL, HVAC_NIGHT_PUI, HVAC_PRE_COOL, HVAC_OFF

The Current State field is used to enter new state names, and to select a state to define. The state table, contained in the `Define Transitions from the Current State` frame, allows you to define the state transitions from the state selected in the `Enter or Select State` field. The state table contains 3 columns. The first, `Exit Condition`, allows you to list the conditions, created in this tab and defined in the `Edit Conditions` tab, that will cause the state machine to change states from the selected state. Select the conditions from the `Exit Conditions` list. Each condition may only be used once in the table for an individual state. When the specified exit condition evaluates to True, the state machine will read the other two columns in the row. The `Next State` column determines what state the machine will be put into when the given condition is True. The `Mode_Out` column determines what mode value will be sent to the Mode_Out network variable of the State Machine functional block when the exit condition is true. The `Digital Outputs for Selected Row` list determines what values will be sent to each of the scheduler's digital outputs when the exit condition evaluates as true.

To define the state and exit condition names, follow these steps:

1. Select the Edit State Machine tab.

2. Click the Current State field. For each bubble in your bubble diagram, enter the name of the state and press the Enter key. The new state is added to the States list.

3. Select the initial state identified in your bubble diagram in the Select Initial State field. **Warning**: If you forget to select an initial state, your schedule will not simulate or operate correctly.

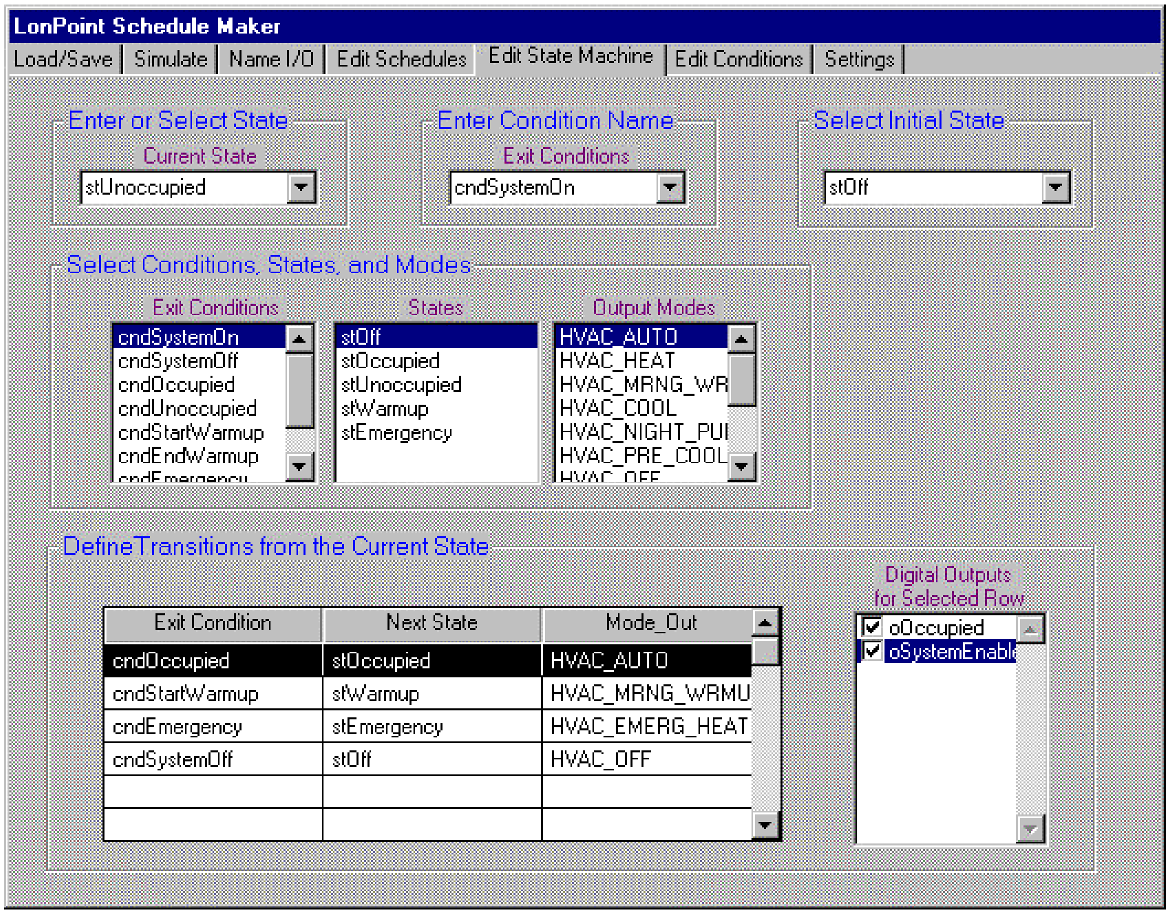
4. Click the Exit Conditions field. For each line in your Condition Definitions shape(s), enter the exit condition name and press the Enter key. The new exit condition name is added to the Exit Conditions list.

For each of your State Transition Definition shapes, follow these steps:

1. Select the state name in the Current State list.

2. For each row of the definition, click on a blank row in the transition table, then click on the entries in the Exit Conditions, States, and Output Modes lists that match the entries in your design. Also click on any outputs that you have defined to be On for the transition.

The following figure shows the state machine definition for the stUnoccupied state defined in the schedule design described in *Defining State Transitions*.



Current State: stUnoccupied
Exit Conditions: cndSystemOn
Select Initial State: stOff

Exit Conditions list: cndSystemOn, cndSystemOff, cndOccupied, cndUnoccupied, cndStartWarmup, cndEndWarmup, cndEmergency

States list: stOff, stOccupied, stUnoccupied, stWarmup, stEmergency

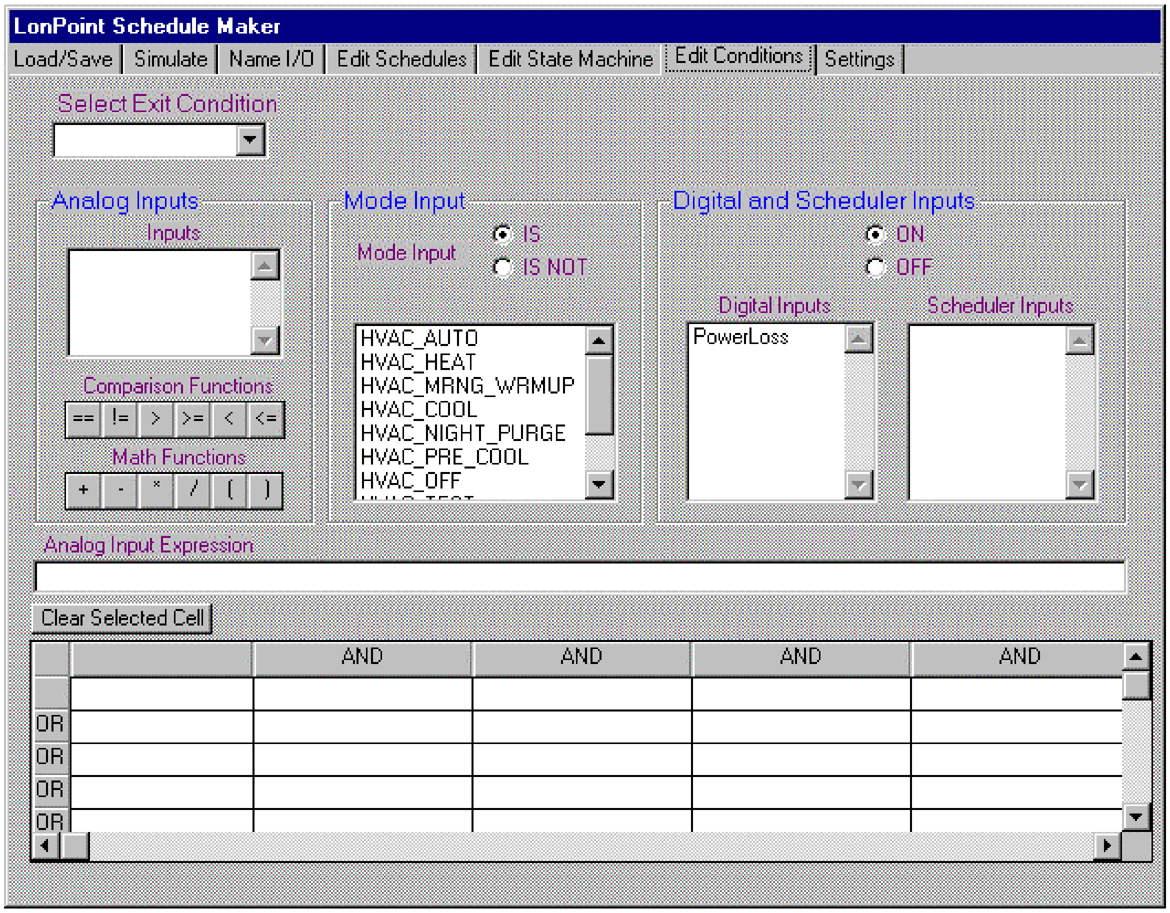
Output Modes list: HVAC_AUTO, HVAC_HEAT, HVAC_MRNG_WR, HVAC_COOL, HVAC_NIGHT_PUI, HVAC_PRE_COOL, HVAC_OFF

Exit Condition	Next State	Mode_Out
cndOccupied	stOccupied	HVAC_AUTO
cndStartWarmup	stWarmup	HVAC_MRNG_WRMU
cndEmergency	stEmergency	HVAC_EMERG_HEAT
cndSystemOff	stOff	HVAC_OFF

Digital Outputs for Selected Row: ☑ oOccupied, ☑ oSystemEnabl

Defining Exit Conditions

The Edit Conditions tab, shown in the following figure, allows you to define the exit conditions that you created in the Edit State Machine tab.



To define exit conditions, select the Edit Conditions tab. For each exit condition, follow these steps:

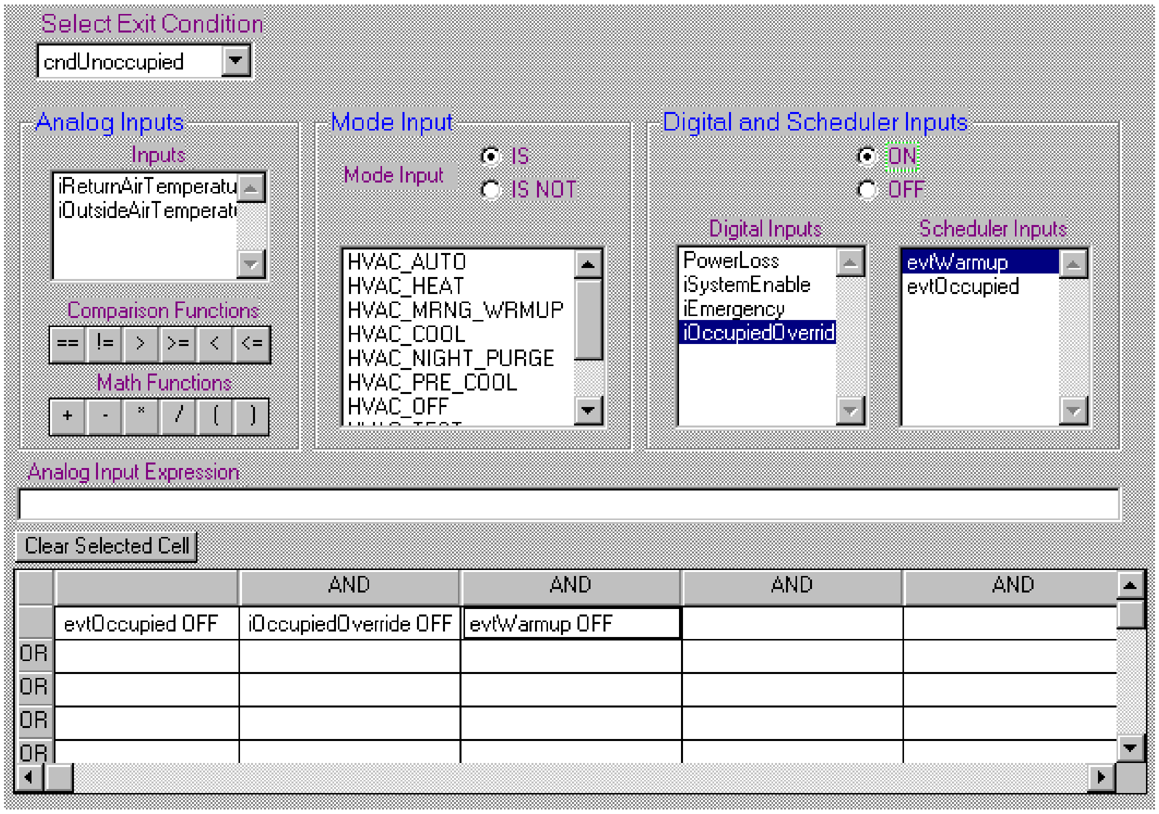
1. Select the exit condition name in the Select Exit Condition list.

2. Enter the condition definition in the AND/OR table at the bottom of the tab. Your definition should consist of terms separated by AND and OR statements. Each term may be an analog input expression, a mode input value, a digital input value, or a scheduler event value. For each term, select a field in the AND/OR table. Enter the first term in the top-left cell of the table. Enter the next term to the right of this cell if it is preceded by an AND statement, and below this cell if it is preceded by an OR statement. Continue this for each term. To enter a term, select the cell and follow one of the following steps:

   - For an analog term, create an expression in the Analog Input Expression field using the analog inputs, floating point numbers, arithmetic functions and one comparison function. You can create the expression by clicking on the appropriate variables and functions, or you can type the expression. When you have completed the expression, press Enter to transfer the expression to the selected cell in the AND/OR table.

- For a mode input term, select the IS or IS NOT modifier, and click on the mode value to copy the term to the current cell in the AND/OR table.
- For a digital input term, select the ON or OFF value and click on the digital input name in the Digital Inputs list value to copy the term to the current cell in the AND/OR table.
- For a scheduler input term, select the ON or OFF value and click on the event name value to copy the term to the current cell in the AND/OR table.

The following figure shows the exit condition definition for the cndUnoccupied definition in the schedule design described in *Defining Exit Conditions*.


Select Exit Condition: cndUnoccupied

Analog Inputs — Inputs: iReturnAirTemperatu, iOutsideAirTemperat

Mode Input: [✓ IS] IS NOT

Mode Input list: HVAC_AUTO, HVAC_HEAT, HVAC_MRNG_WRMUP, HVAC_COOL, HVAC_NIGHT_PURGE, HVAC_PRE_COOL, HVAC_OFF

Digital and Scheduler Inputs: [✓ ON] OFF

Digital Inputs: PowerLoss, iSystemEnable, iEmergency, iOccupiedOverrid (selected)

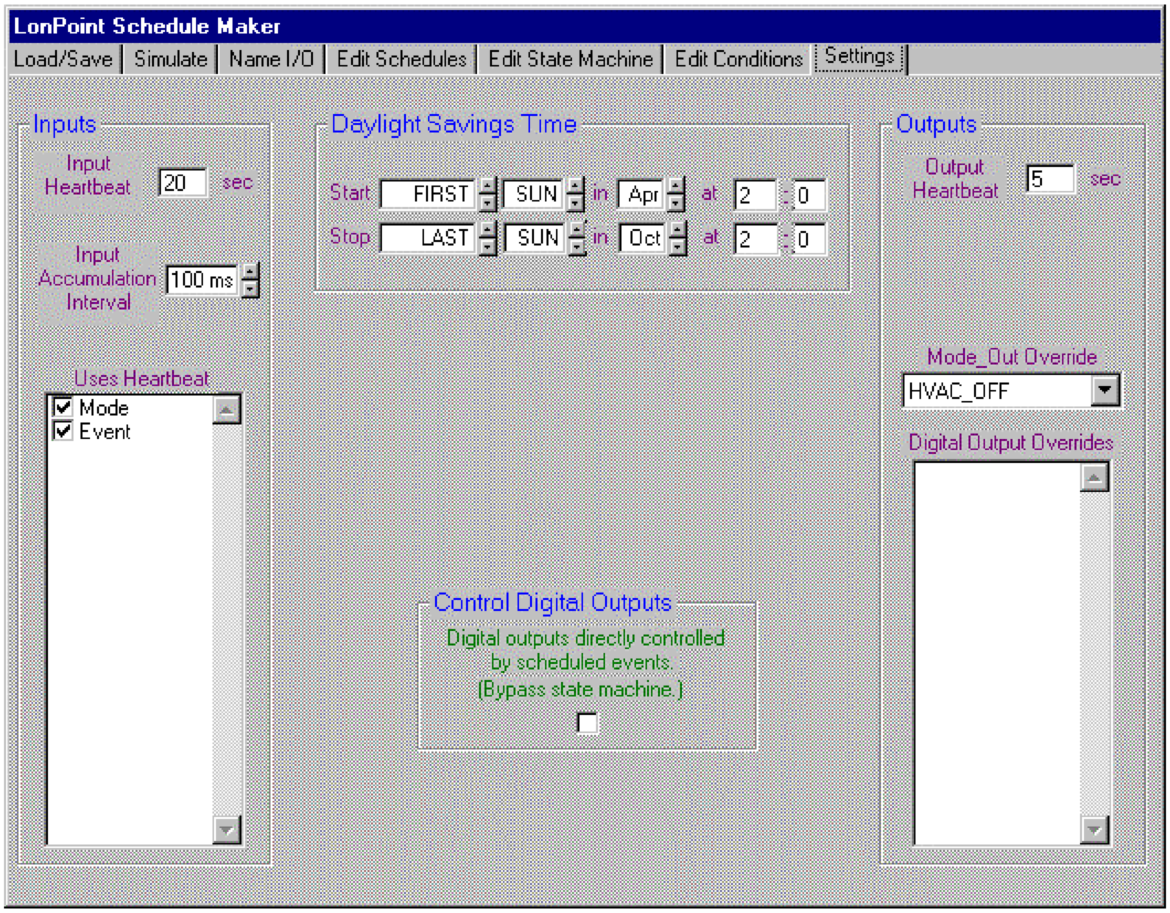
Scheduler Inputs: evtWarmup (selected), evtOccupied

Analog Input Expression:

		AND	AND	AND	AND
	evtOccupied OFF	iOccupiedOverride OFF	evtWarmup OFF		
OR					
OR					
OR					
OR					

Setting Scheduler Options

The Settings tab, shown in the following figure, allows you to determine data flow configuration properties for the Event Scheduler and State Machine functional blocks.



Input Heartbeat 20 sec

Input Accumulation Interval 100 ms

Uses Heartbeat
- [✓] Mode
- [✓] Event

Start FIRST SUN in Apr at 2 : 0

Stop LAST SUN in Oct at 2 : 0

Output Heartbeat 5 sec

Mode_Out Override HVAC_OFF

Digital Output Overrides

Digital outputs directly controlled by scheduled events. (Bypass state machine.) ☐

This tab contains the following properties:

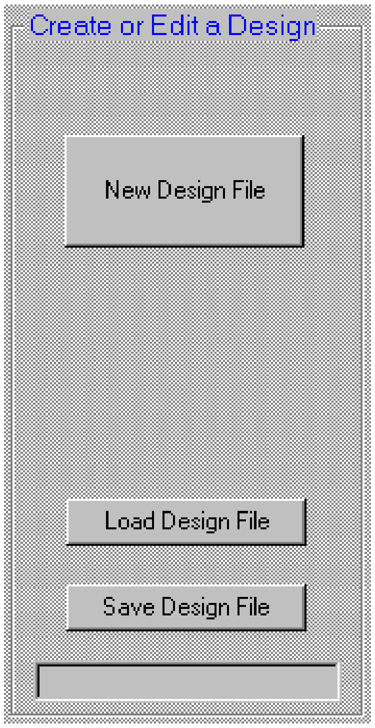
Input Heartbeat	Determines the interval (in seconds) the network variables that have the *Uses Heartbeat* option selected will wait for a heartbeat before registering a heartbeat failure. See *Heartbeats* in Chapter 1 for more information.
Input Accumulation Interval	The time period for accumulating changes on the State Machine functional block's input network variables before taking action. When an external input, such as an analog sensor value, is received, an internal timer is started. The timer runs for the number of milliseconds specified by this entry. While this timer is running, the input is not processed and changes that occur during this time are treated as simultaneous. This allows originally simultaneous sensor inputs that have lost synchronization to be re-synchronized. This should typically be left at the

	default value of 100ms.
Uses Heartbeat	Specifies whether or not heartbeat checking will be used for the selected input network variables. Each of the input network variables named in the Name I/O tab plus the Mode and Event input network variables are included.
Daylight Savings Time	Defines when Daylight Savings Time starts and stops. See *Configuring the Real Time Clock Functional Block* earlier in this chapter.
Control Digital Outputs	Determines whether digital outputs are controlled by the State Machine functional block (a state output schedule) or by the Event Scheduler function block (a digital output schedule). When checked, the digital output schedule mode is enabled and the state machine is bypassed for the digital outputs. See *Configuring a Digital Output Schedule* earlier in this chapter.
Output Heartbeat	Specifies how often output heartbeats are sent by the State Machine functional block's output network variables. See *Heartbeats* in Chapter 1 for more information.
Mode_Out and Digital Output Override Values	Defines the values that will be sent on the output network variables when the State Machine functional block is put into override by the LonMaker tool. Only the output network variables named in the Name I/O tab will be listed in the `Digital Output Overrides` list. See *Override* in Chapter 1 for more information.

Saving and Loading a Supervisory Application Design File

Supervisory application designs must be saved in design files. A *supervisory application design file* (.DSG extension) includes all the information required to define a schedule and state machine design, and includes information such as input/output names that is not downloaded to the SCH-10 module or stored in the LNS database. You can create and save many design files so that you can use the LonPoint Schedule Maker simulator to simulate design alternatives. Once you have validated a schedule design using the simulator, you can download the design to an SCH-10 module as described in *Downloading a Supervisory Application Design*. There may be only one open design at a time.

To start a new design, or to save or load a design file, select the Load/Save tab. The following buttons are used to create, load, and save design files:



To create a new design, click the New Design File button. This clears all entries on the Schedule Maker tabs so that you can create a new design.

To save a design, follow these steps:

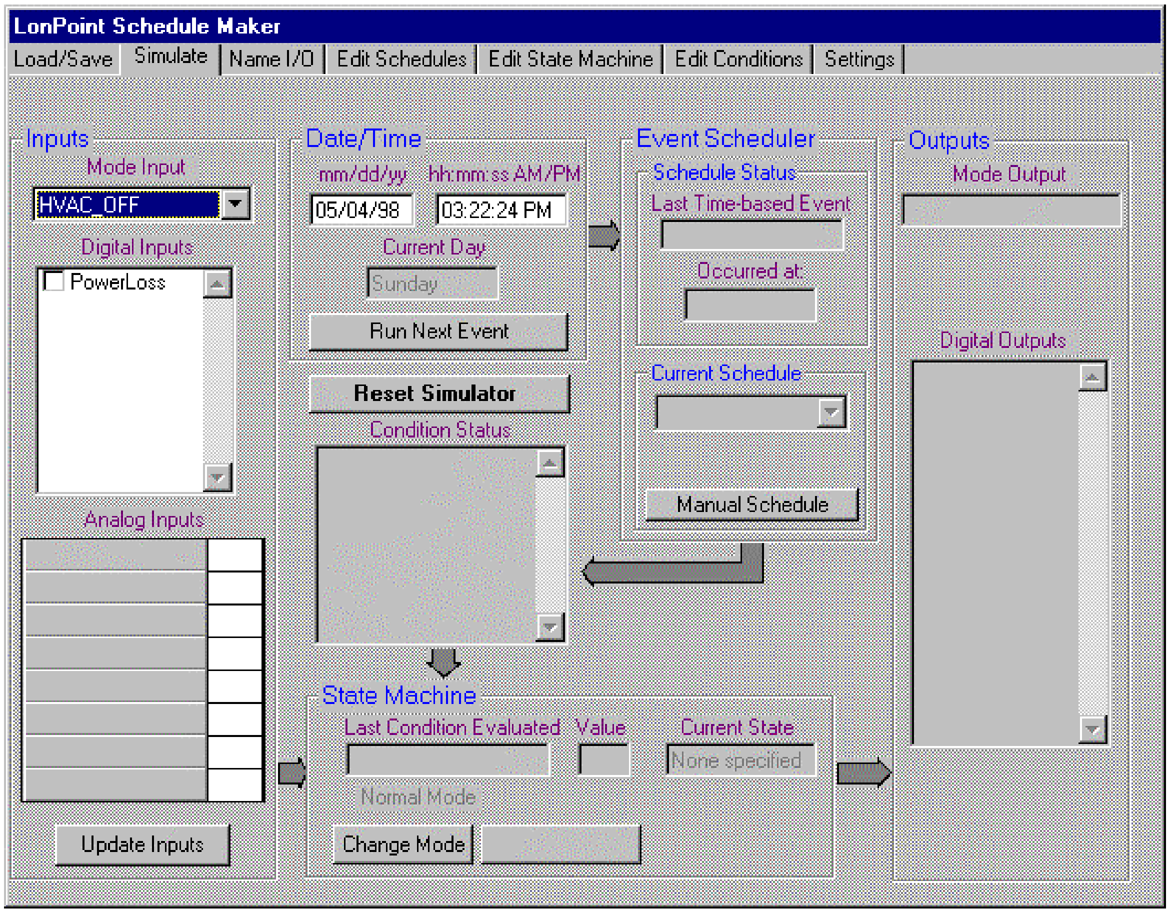
1. Click the Save Design File button.
2. Locate a drive and folder for the schedule design file.
3. Enter a file name. You do not have to type the .DSG extension.
4. Click Save. The design is saved in the selected file.

To load an existing design file, follow these steps:

1. Click the Load Design File button.
2. Locate the drive and folder that contains the schedule design file.
3. Click the file you want to load.
4. Click Open. The design saved in the file is loaded to the LonPoint Schedule Maker tabs.

Simulating a Supervisory Application Design

The LonPoint Schedule Maker simulator allows you to verify a supervisory application design before downloading the design to an SCH-10 module. Designs should always be simulated before downloading to verify that the SCH-10 module will function as expected. You will use the Simulate tab, shown in the following figure, to interact with the simulator.



The Simulate tab shows a graphical representation of the data flow within the SCH-10 application. The following sections describe how to use this tab to simulate a design. This tab displays the following information:

Inputs	Controls the simulated input values for the mode, digital, and analog input network variables of the State Machine functional block. The simulator ignores any inputs to these fields until you have clicked the Update Inputs button. See *Simulating Inputs* later in this chapter for more information.
Date/Time	Displays the simulated date, time, and day of the week. You can skip ahead to the next event using the Run Next Event button as described in *Running to the Next Event* later in this chapter.
Event Scheduler	Displays the status of the schedules defined in the Edit Schedules tab. The `Last Time-based Event` field displays the last event sent from the event scheduler and the `Occurred at:` field displays the time at which the event was sent.

	The `Current Schedule` field indicates what daily schedule is currently being used. You can manually set the schedule by clicking on the `Manual Schedule` button as described in *Overriding the Event Schedule* later in this chapter.
Condition Status	Displays the status of the events defined in the Edit Schedules tab. For example, if you have defined evtOccupied ON and evtOccupied OFF events in your schedule, and evtOccupied is selected in the Condition Status list, the last evtOccupied event was an evtOccupied ON event. If evtOccupied is cleared, the last evtOccupied event was an evtOccupied OFF event.
State Machine	Displays the current state of the state machine. The current state is displayed in the `Current State` field. When debugging, additional information is displayed showing the value of each condition as it is being evaluated by the state machine. Debugging is described in *Using Debug Mode* later in this chapter.
Outputs	Displays the simulated output values for the mode network variable output and the 16 digital output network variables (any or all of which can be named in the Name I/O tab). These values may change each time the state changes.

Resetting the Simulator

You can reset the simulator at any time to simulate resetting the SCH-10 module. You must always reset the simulator prior to using any of the other simulator features. To reset the simulator, click the Reset Simulator button on the Simulate tab. The Current State at the bottom of the tab should be the initial state that you selected for the state machine on the Edit State Machine tab.

Using Debug Mode

You can single step the evaluation of conditions by the state machine. This is useful for debugging a state machine. To single step evaluation, follow these steps:

1. Click the Change Mode button. A new Eval Next Cond button is displayed to the right of the Change Mode button; the last condition evaluated by the state machine is displayed in the `Last Condition Evaluated` field; and the value of the condition when it was evaluated is displayed in the `Value` field.

2. Press the Run to Next Event or Update Inputs button.

3. Click the Eval Next Cond button. The next condition specified for the current state is evaluated, and the results are displayed in the simulator tab. If there are more conditions to be evaluated, the Change Mode button is disabled and the Eval Next Cond button is still displayed. If there are no more conditions to be evaluated, the Eval Next Cond button is hidden and the Change Mode button is reenabled.

Simulating Inputs

You can control the simulated input values for the mode, digital, and analog input network variables of the State Machine functional block. These inputs are listed under the Inputs heading on the Simulate tab. Only the analog and digital network variables named in the Name I/O tab are listed.

To update the simulated Mode input, follow these steps:

1. Select a value in the Mode Input field.
2. Update any other inputs then click the Update Inputs button.

To update the simulated digital inputs, follow these steps:

1. Select the box to the left of the digital input name to simulate an On input; clear the box to simulate an Off input.
2. Update any other inputs then click the Update Inputs button.

To update the simulated analog inputs, follow these steps:

1. Click the box to the right of the analog input name and enter the new value.
2. Press the Enter key to enter the value.
3. Update any other inputs then click the Update Inputs button.

To simulate a power loss, follow these steps:

1. Select the box to the left of the PowerLoss field in the Digital Inputs list to simulate a power loss input; clear the box to indicate no power loss.
2. Update any other inputs then click the Update Inputs button.

The simulator ignores any inputs to the input fields until you have clicked the Update Inputs button so that you can make several changes before running the state machine.

Running to the Next Event

You can force the simulator to skip ahead in time to the next scheduled event. Scheduled events are defined on the Edit Schedules tabs. To run to the next event, click the Run Next Event button. The date and time will advance to the next scheduled event, and the simulator tab will be updated with the current state of the simulated Event Scheduler and State Machine functional blocks. If the current schedule has no events defined, the simulator will run until 12:01 AM of the next day.

Manually Entering Time and Date

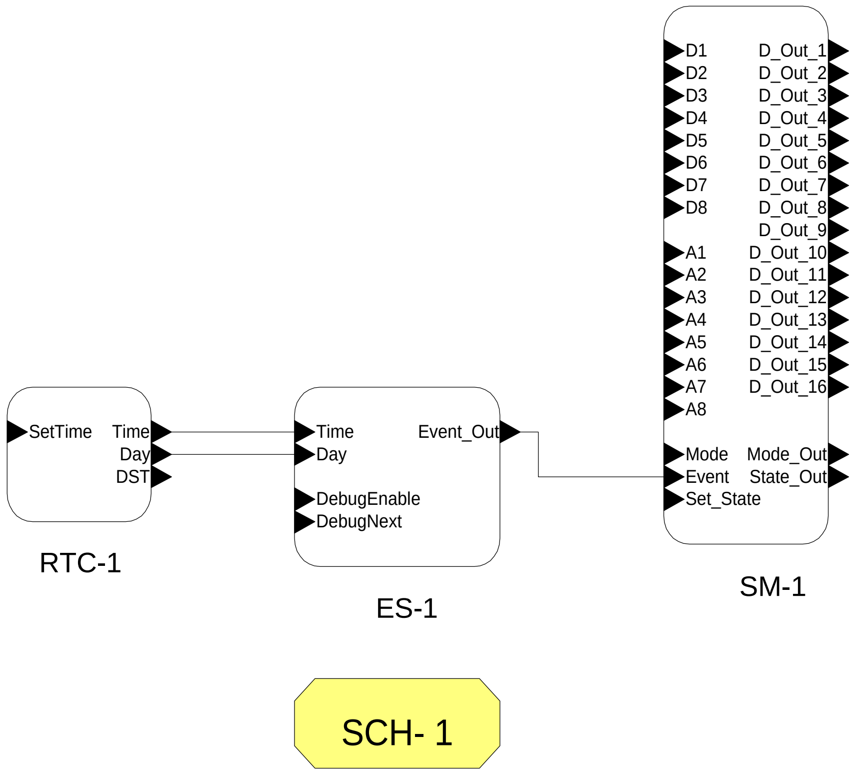
You can manually change either the time or date in the simulator in order to verify operations at future times/dates. Whenever either field is manually changed, the simulator will automatically perform a reset by backing up twenty-four hours for the new time/date and fast forwarding to the new time/date while simulating all events in the last twenty-four hours in order to get the state machine in a state that is current relative to the new date and time.

Overriding the Event Schedule

The `Current Schedule` field indicates what daily schedule is currently being used. You can manually set the schedule by clicking on the `Manual Schedule` button and selecting the desired schedule from the current schedule list. This allows you see the behavior of all daily schedules without waiting for every daily schedule to become active based on the simulated day or date.

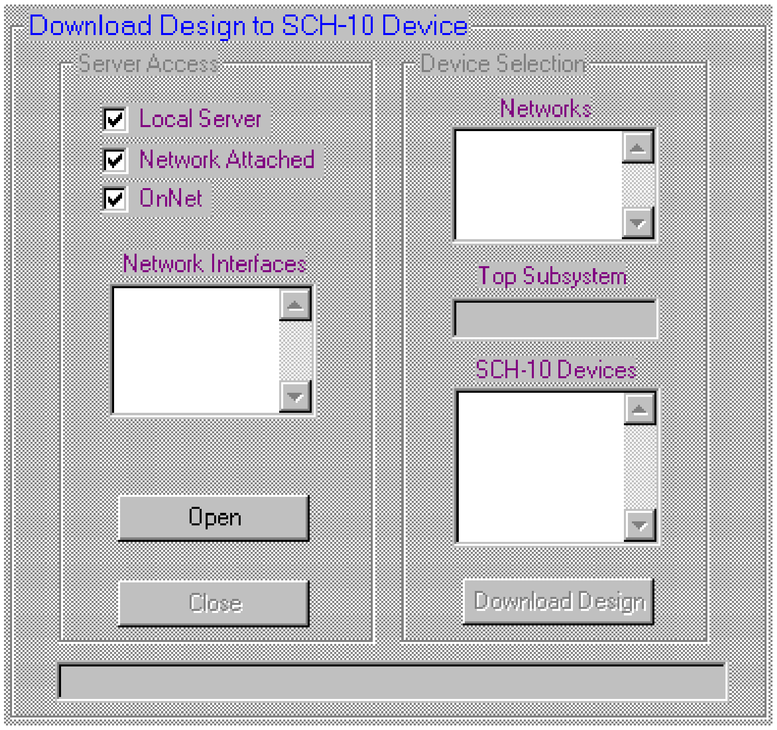
Adding the Scheduler Functional Blocks

Once you have successfully simulated a design, you must install the SCH-10 device and functional blocks using the LonMaker tool. You can install the device and functional blocks individually and connect them as shown in the following figure. At least one of these shapes must be in your top-level subsystem, or you can put these shapes in any subsystem and add a Node Object functional block for the SCH-10 device to your top-level subsystem. This will ensure that the LonPoint Schedule Maker utility finds an entry in the top-level subsystem when you download the schedule design.



Downloading a Supervisory Application Design

The supervisory application design is downloaded to the LNS Server, and optionally an SCH-10 module, using the following Load/Save tab.



To download the design into an LNS Server, and optionally a SCH-10 module, follow these steps:

1. Select `Local Server` if you are running on the same PC with the LNS Server; uncheck `Local Server` if you are running on a remote PC.

2. Select `Network Attached` if your PC is physically attached to the network; uncheck `Network Attached` if your PC is not physically attached.

3. Select `OnNet` if your PC is attached, and you want the design to be immediately downloaded to the SCH-10 module. Clear `OnNet` to store the design only in the LNS database. The design will be downloaded to the device the next time the LNS Server is OnNet.

4. Click the `Open` button. The available LNS network interfaces will be listed in the Network Interfaces list.

5. Select a network interface in the Network Interfaces list to show all available networks in the `Networks` list.

6. Select the network containing the SCH-10 device to be configured from the `Network` list. After a delay, the LonPoint Schedule Maker utility determines the top-level subsystem of the selected network and displays it in the `Top Subsystem` field. All the available SCH-10 devices in this subsystem are listed in the SCH-10 Devices list. The LonPoint Schedule

Maker utility can only configure SCH-10 modules that have device or functional block shapes in this top-level subsystem.

7. Select the SCH-10 device to be configured in the SCH-10 Devices list.
8. Click the `Download Design` button to configure the selected device. Any previous schedule design for the device will be overwritten. The download may take one to five minutes depending on schedule size and network load. If you are OnNet, you can speed up downloading by turning off monitoring in the LonMaker tool. A status message at the bottom of the Load/Save tab will display the download status. When the status message is cleared, the download is complete.

Once you have finished loading, saving, and downloading designs into a network, you should close the network database by clicking the `Close` button. You may then open another database if you wish.

Click the `Exit` button in this tab to exit the LonPoint Schedule Maker application.

Setting the SCH-10 Time

Prior to using the SCH-10 module for the first time, you must ensure that it has the correct date and time. To set the time, follow these steps:

1. Open a drawing containing the SCH-10 device using the LonMaker tool.
2. Right-click the Real-Time Clock functional block and select Browse from the shortcut menu.
3. Right-click anywhere in the row containing the SetTime network variable and select Details from the shortcut menu.
4. For each field, click the plus sign next to the field, click twice on the value (i.e. slower than a double-click), and enter the correct value.
5. Click OK to update the time.

Changing the Mode and Mode_Out Types

By default, the Mode and Mode_Out network variables of the State Machine functional block are the `SNVT_hvac_mode` type. You can change these types to generate different types of mode inputs or mode outputs. However, the types must be set to `SNVT_hvac_mode` when downloading a design as described in the previous section.

To change the type of the Mode or Mode_Out network variables, follow these steps:

1. Download the design to the SCH-10 module as described in *Downloading a Supervisory Application Design*.
2. Open the LonMaker subsystem containing the State Machine functional block.

3. Right-click the State Machine functional block and select Browse from the shortcut menu to open the LonMaker Browser.
4. Right-click the Mode or Mode_Out network variables and select Change Type from the shortcut menu. The Change Network Variable Type dialog appears.
5. Select a new network variable type from the Type List. The Standard Network Variable Type option allows you to select the type from the SNVTs detailed in the SNVT Master List help file. If it is not checked, you can choose from user-defined types in the type catalog.
6. Click OK to close the Change Network Variable Type dialog.
7. Close the LonMaker Browser.

After changing the type of the Mode input or Mode_Out output, you must change the type back to SNVT_hvac_mode before downloading a new schedule design to the SCH-10 module. To change the type back, follow steps 2 through 6 of the above procedure and select SNVT_hvac_mode in step 5.

Testing a Schedule Design

The Event Scheduler functional block includes three network variables to simplify debugging a supervisory application design that has been downloaded to an SCH-10 module. The network variables are the DebugEnable, DebugNext, and Time input network variables. The DebugNext input network variable operates like the Run Next Event button in the simulator, that is it allows you to fast forward the Event Scheduler to the next event. The Time input network variable allows you to monitor the simulated time maintained by the Event Scheduler functional block while debugging is enabled. To test a schedule design using these network variables, follow these steps:

1. Download the schedule to the SCH-10 module as described in *Downloading a Supervisory Application Design*.
2. Open the LonMaker subsystem containing the Event Scheduler functional block.
3. Right-click the Event Scheduler functional block and select Browse from the shortcut menu to open the LonMaker Browser.
4. Right-click the Time input network variable row and select Monitor from the shortcut menu. This allows you to monitor the simulated time maintained by the Event Scheduler functional block.
5. Select the DebugEnable input network variable and set its value to 100 1, corresponding to an input of 100% and On. This stops the real-time clock, allows you to monitor the simulated time, and enables the DebugNext input. After setting DebugEnable, you should see the monitored time stop.
6. Select the DebugNext input network variable and set its value to 100 1. The Time network variable will advance to the next event, and the Event output from the Event Scheduler functional block will be set to indicate the new

event. If connected, the State Machine functional block will respond to the updated input.

Note: *When Debug Enable is On* ***and*** *DebugNext is updated and On, the Event Scheduler emits the next event. While the EventScheduler is looking up the next event, it also is responding to the network variable polls at a slowed rate. If you select the Stop Automatic Display option of the LonMaker Browser Message dialog box, you can stop message flow. DebugNext will update, the EventScheduler will stop at the next event, and network variable polls will be responded to normally.*

11

The Schedule Keeper Utility

This chapter describes how to use the LonPoint Schedule Keeper utility to modify a schedule created using the Schedule Maker utility.

The Schedule Keeper Utility

The LonPoint Schedule Keeper utility is used by system operators to perform limited on-site modification to a supervisory application after it has been downloaded into an SCH-10 device. This utility allows the operator to change the times when scheduled events will be triggered and modify which days on the special schedules will be used. It does not allow the operator to create new schedules or new events. Schedules and events must be created using the LonPoint Schedule Maker utility described in Chapter 10.

Schedule Keeper can be started in *super user* mode as described in the next section. Super user mode allows you to create constraints on the end user functionality of Schedule Keeper. This allows the schedule creator to limit the changes that the end user can make to the schedule using Schedule Keeper.

Note: It is recommended that all changes to the SCH-10 functional blocks be made with the Schedule Maker and Schedule Keeper applications, not with the LonMaker Browser. If the LonMaker Browser must be used, reset the device after making changes. If the SCH-10 device is not reset, the SCH-10 device may not operate correctly.

Starting the Schedule Keeper Utility

Schedule Keeper is typically called from an HMI application such as an operator interface created by Wonderware In Touch. To simplify this type of usage, Schedule Keeper is started from the Windows command line using a launch file created by the Schedule Maker.

When the Schedule Maker application downloads a schedule into an SCH-10 device, the launch file is created in the LONWORKS SchApp\Launch folder. The name for this file has the following format:

*designName***.***networkName***.***systemName***.***subsystemName***.***deviceName***.***lau*

There are two flags which may be set when opening Schedule Keeper. The /L flag allows you to enter a specific Schedule Maker launch file as an argument. The /S flag causes Schedule Keeper to open in *super user* mode. Super user mode allows you to enter the *Schedule Constraints* and *Parameter Constraints* dialogs described below. If you open Schedule Keeper without the /S flag, you will only be able to access the *Schedules*, *Daily Defaults*, *Overrides*, and *System Parameters* dialogs.

To start Schedule Keeper directly in non super user mode, double-click the launch file associated with the schedule you want to view or edit.

To create a shortcut that allows you to open Schedule Keeper with one or more arguments, follow these steps:

1. Right-click the launch file in the LONWORKS SchApp\Launch folder and drag it to the desired location.

2. Release the mouse button and choose *Create Shortcut(s) Here* from the shortcut menu.

3. Right click the shortcut and select *Properties*.

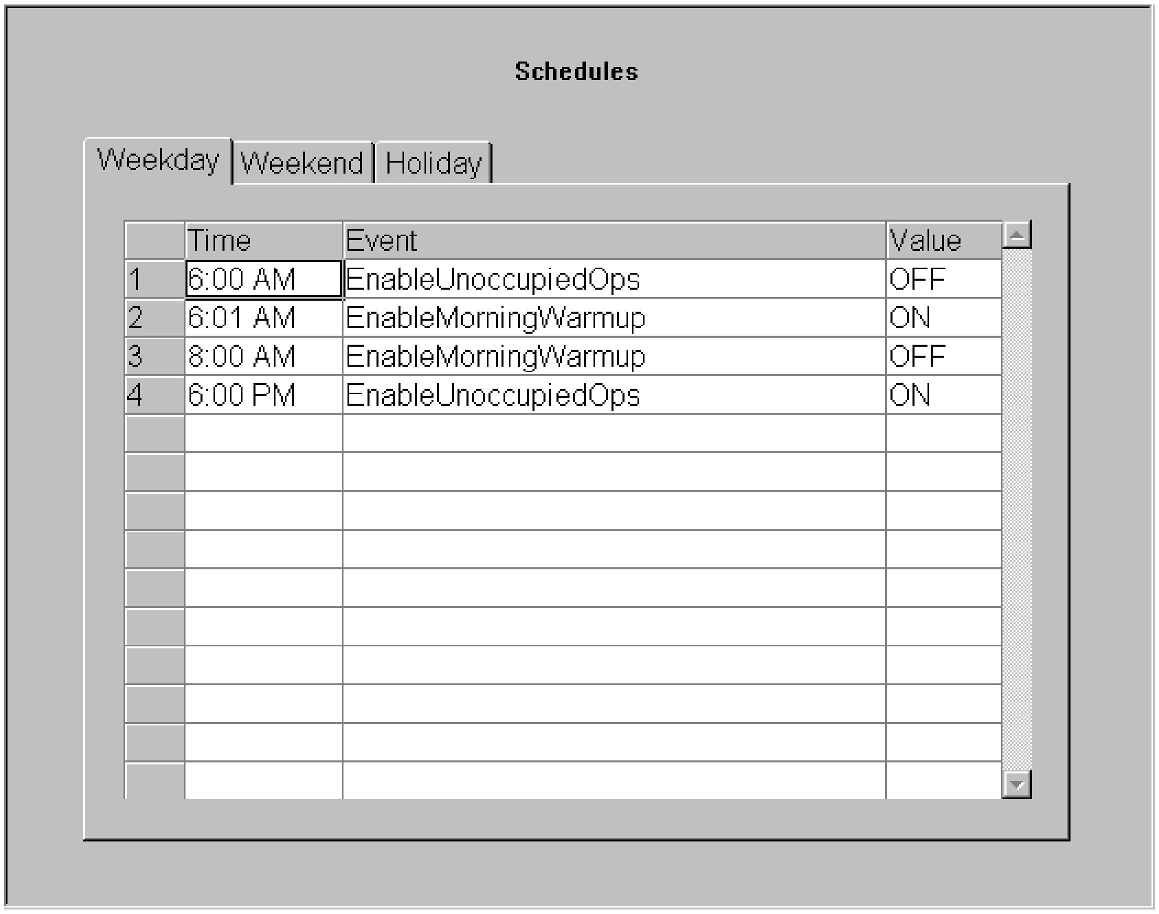
4. Click the *Shortcut* tab and enter the desired command line in the `Target` field. For example, to create a shortcut that starts Schedule Keeper for the specified schedule in super user mode, enter the following command line:

```
C:\LonWorks\Apps\Echelon\LonPoint\schkeepr.exe /S /L
"C:\LonWorks\SchApp\Launch\Fullsm1.NewNet.NewNet.Subsystem
1.SCH- 1.lau"
```

When Schedule Keeper starts, a window opens with the `Schedules`, `Daily Defaults`, `Overrides`, and `Save Changes` buttons and optionally the `System Parameters`, `Schedule Constraints`, and `Parameter Constraints` buttons. Click one of these buttons to begin modifying the schedule as described below.

Modifying the Daily Schedules

Click the `Schedule` button to open the following window:



	Time	Event	Value
1	6:00 AM	EnableUnoccupiedOps	OFF
2	6:01 AM	EnableMorningWarmup	ON
3	8:00 AM	EnableMorningWarmup	OFF
4	6:00 PM	EnableUnoccupiedOps	ON

This window contains one tab for each daily schedule created using Schedule Maker. In the example above, there are three daily schedules: `Weekday`, `Weekend`, and `Holiday`. Each daily schedule consists of a list of events. Each event has an Off and an On state. The meaning of these events will be provided by the schedule designer.

The example above shows a typical weekday schedule for a building. The Weekday schedule has four events. At midnight, the `EnableUnoccupiedOps` event is turned on. This event could involve shutting down heating and light and turning on an alarm system. At 6:00 AM, this event is turned off. At 6:01, the `EnableMorningWarmup` event begins. This event could involve turning on heaters and any other machinery that requires a warmup period. At 8:00 AM, this event is shut off, putting the building into its normal running state.

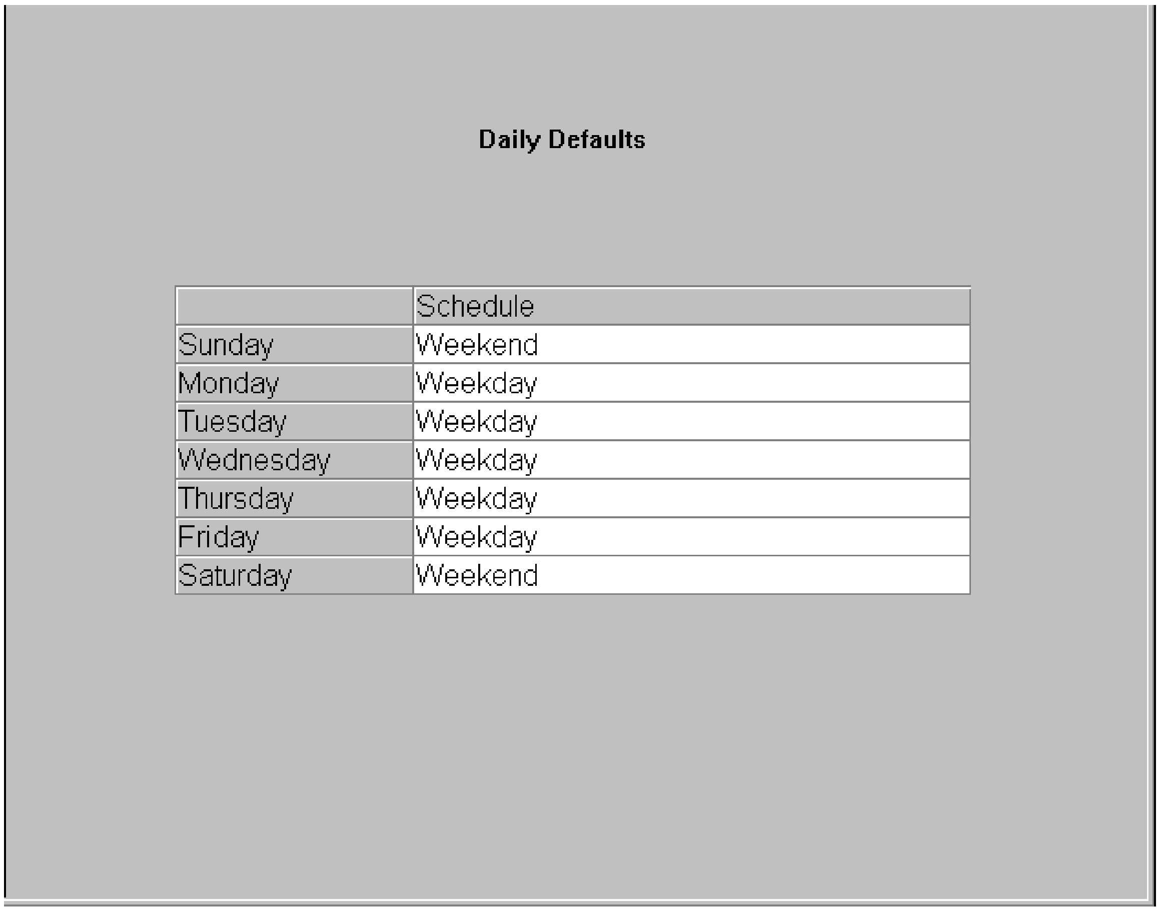
This window allows you to change the time an event is triggered, what event is triggered, and whether that event is set to On or Off. Click one of the times in the `Time` column to enter a new time for the event. If an invalid time is entered, the time will revert to its previous value. If the times are changed in such a way that the order of these events is no longer sequential (i.e. an event occurring at 4:00 PM is listed before an event at 2:00 PM), Schedule Keeper will reorder the events the next time this dialog is opened.

Click one of the events in the `Event` column to choose an event from a list of all events available on this SCH-10 device. Click a state in the `Value` column to choose `ON` or `OFF` from a menu.

This dialog does not allow you to change the number of events. In the example above, there will always be four events in the `Weekday` schedule. Schedule Keeper can be used the change what they are and when they occur. Use Schedule Maker, described in Chapter 10, to change the number of events.

Modifying the Daily Default Schedule

Click the Daily Defaults button to open the following window:



Daily Defaults

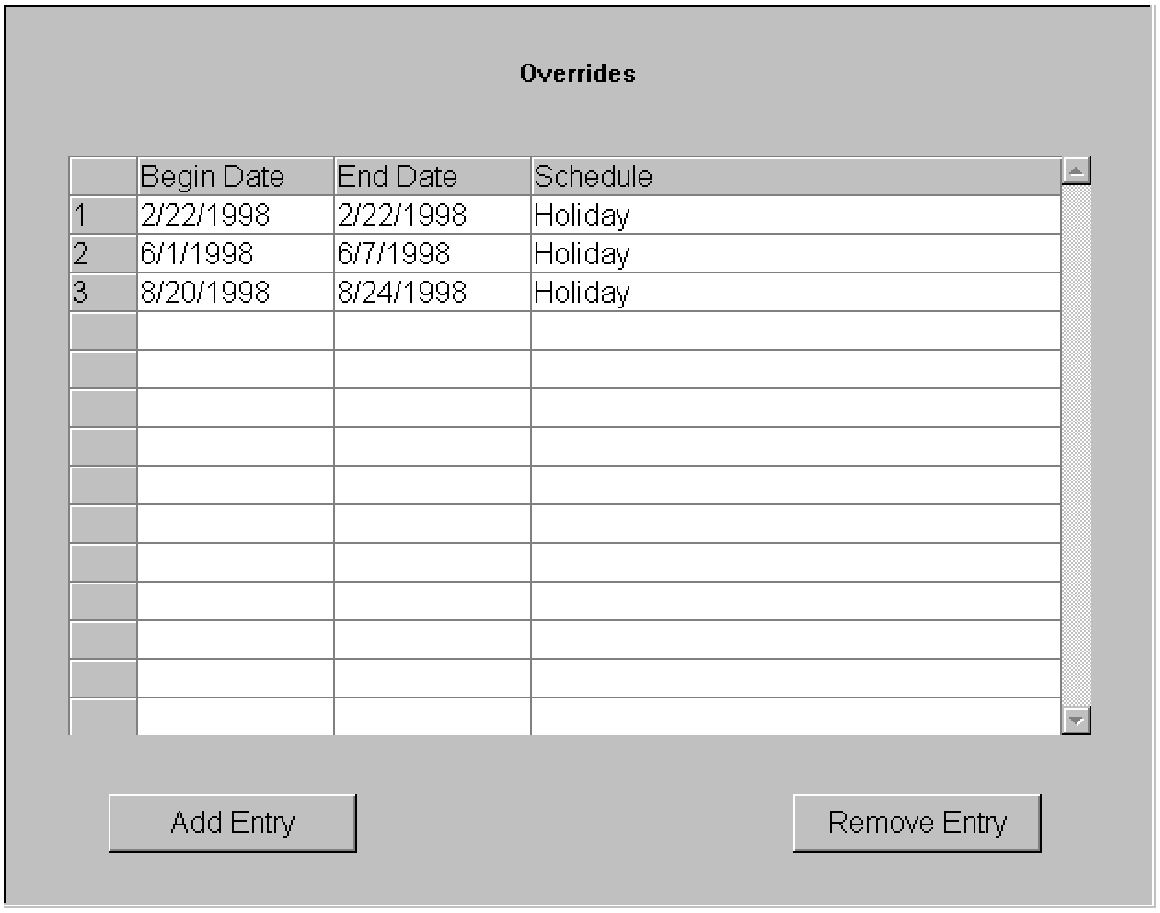
	Schedule
Sunday	Weekend
Monday	Weekday
Tuesday	Weekday
Wednesday	Weekday
Thursday	Weekday
Friday	Weekday
Saturday	Weekend

This window allows you to determine which schedule will be used on each day of the week by default (see *Modifying Schedule Overrides*, in the next section, for information on overriding the default schedule). Click an entry in the Schedule column to select a new daily schedule from a list of all daily schedules available on this SCH-10 device.

In the example above, the Weekday schedule is the default from Monday to Friday, and the Weekend schedule is the default on Saturday and Sunday.

Modifying Schedule Overrides

Click the `Overrides` button to open the following window:



	Begin Date	End Date	Schedule
1	2/22/1998	2/22/1998	Holiday
2	6/1/1998	6/7/1998	Holiday
3	8/20/1998	8/24/1998	Holiday

This window allows you to view, add, remove, and modify dates during which the daily default schedules (see *Modifying the Daily Default Schedule*, earlier in this chapter) will be overridden. In the example above, the Holiday schedule is configured to be used on December 22, June 1 to June 7, and August 20 to August 24th.

Click a value in the `Begin Date` or `End Date` columns to change the beginning or ending date for a schedule override period. This date must be in the format specified by the PC's operating system. You can find out what the current format is by entering something clearly illegal (i.e. "?"); this will open a dialog that contains the correct format. Do not enter overlapping periods on this table.

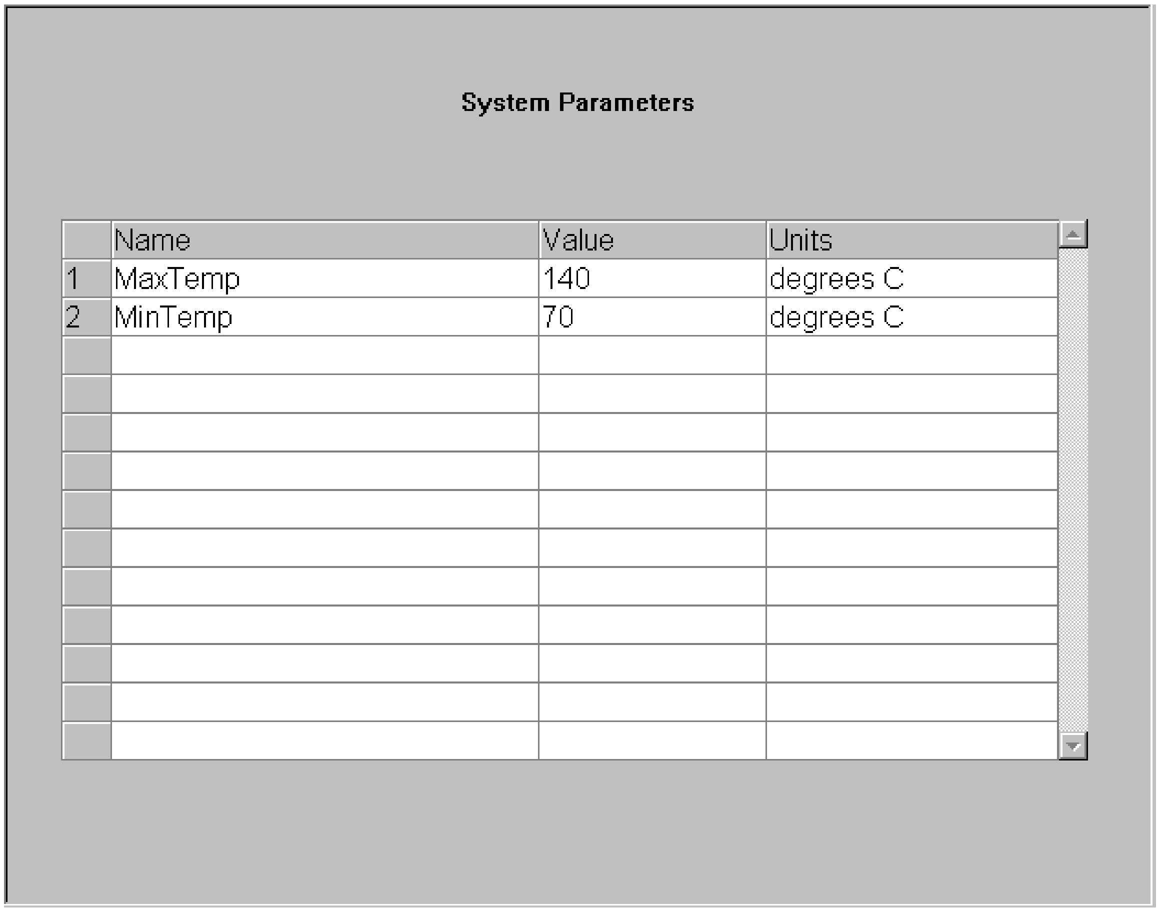
Click a value in the Schedule column to select a daily schedule from a pull-down list of all daily schedules available on this SCH-10 device. The details of the daily schedules can be viewed using the `Schedules` button (see *Modifying Daily Schedules*, earlier in this chapter).

Click `Add Entry` to add a period during which the daily Default schedule will be overridden. This will cause a new line to be added to the column with default start and end dates of 1/1/2038 and nothing in the Schedule column. Modify the dates and select a schedule as described above.

Click `Remove Entry` to delete an override schedule. A dialog opens asking you which entry you would like removed. Enter the number displayed at the left end of the row containing the schedule override period to be removed and click `OK`. The specified row will be removed and all rows beneath it will move up. Alternately, you can select a row or group of rows by clicking on the row number(s) and clicking `Remove Entry`.

Modifying System Parameters

The `System Parameters` button will only appear if one or more *System Parameters* were defined in the `Name I/O` tab of the *Schedule Maker*. Click `System Parameters` to open the following window:



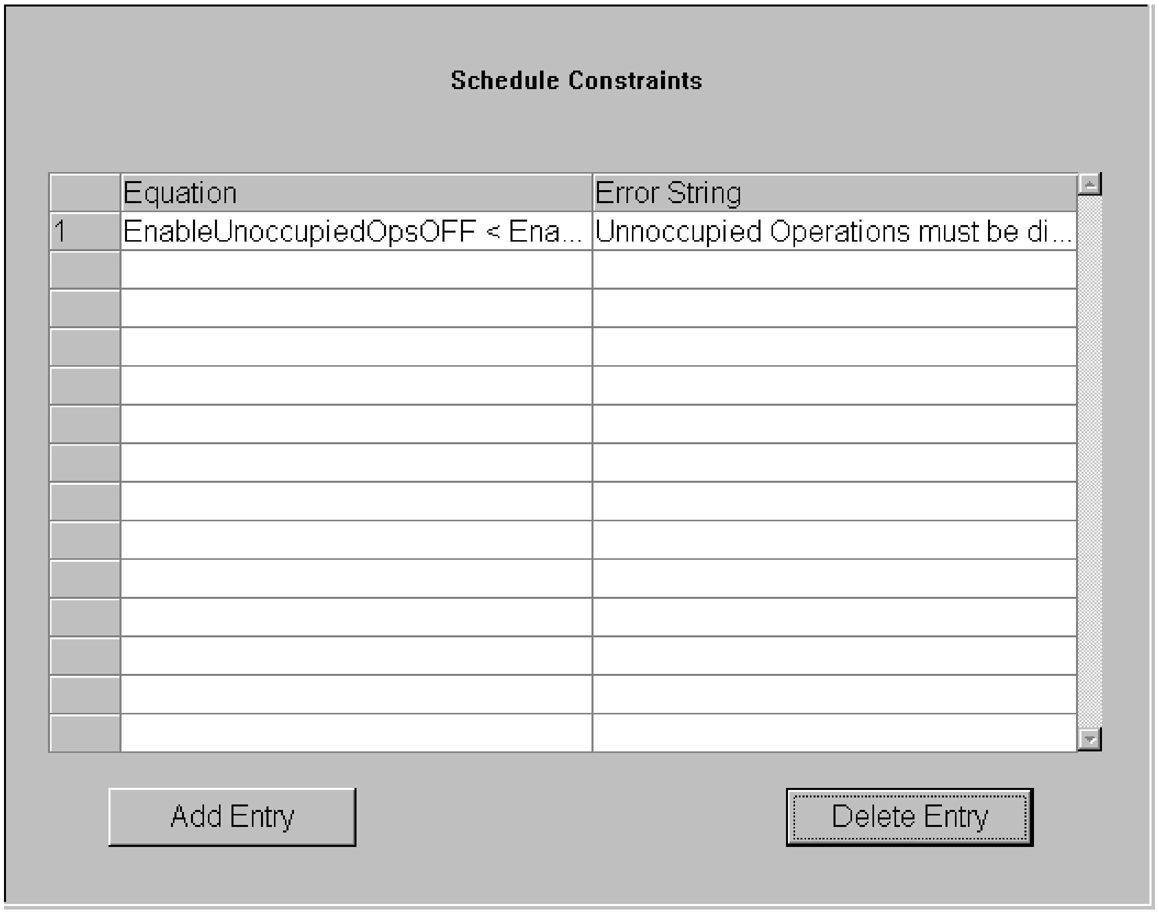
	Name	Value	Units
1	MaxTemp	140	degrees C
2	MinTemp	70	degrees C

This window allows you to view and modify the system parameters. The meaning of these parameters will be supplied by the schedule designer. Only the parameter values may be changed. Parameters cannot be added or deleted, nor can the names be changed. To change a parameter value, click the entry in the Value column for the parameter. A cursor will appear in the value you clicked which will allow you to modify the parameter value. Parameters may be set to any floating point value. Valid values for parameters may be limited using the `Parameter Constraints` dialog (available in super user mode only).

The Units column will only have values in it if parameter unit formatting has been set up as described in *Converting to US Units*, later in this chapter. Otherwise this column will be blank.

Adding Schedule Constraints

The `Schedule Constraints` button will only be available if Schedule Keeper was opened in super user mode. Click the `Schedule Constraints` button to open the following window:



	Equation	Error String
1	EnableUnoccupiedOpsOFF < Ena...	Unnoccupied Operations must be di...

Use this window to create, view, modify, and delete constraints for the values set in the `Schedules` dialog and error messages which will be displayed if these constraints are violated.

In the example above, the constraint equation is `EnableUnoccupiedOpsOFF < EnableMorningWarmpuON` and the error string is "Unoccupied Operations must be disabled before Morning Warmup can begin." If the operator attempts to enter a schedule in the `Schedules` window that violates this constraint, this error string will be displayed.

To add a schedule constraint, click `Add Entry` and fill in the `Equation` and `Error String` columns for the new row. To delete a schedule constraint, select the constraint to be deleted and click `Delete Entry`.

Schedule Constraint Equations

Schedule constraint equations are created to specify site-specific limitations in the way that Schedule Keeper users may modify a Schedule. These equations are then evaluated every time that the user exits the “Schedules” modification screen or moves from one Schedule tab to another. If the equation evaluates to FALSE when schedule constraints are evaluated, the constraint error string will be displayed and the modification will not be accepted.

Legal equations include:

The comparison equation – if you want to compare one event time to another, all of the standard comparison operations are available. Examples include:

```
EnableUnoccupiedOpsOFF < EnableMorningWarmupON
```

The `EnableUnoccupiedOpsOFF` event must occur before the `EnableMorningWarmupON` event.

```
EnableUnoccupiedOpsOFF + 00:10 > EnableMorningWarmupON
```

The `EnableUnnocupiedOpsOFF` event must occur either after or less than 10 minutes before the `EnableMorningWarmupON` event.

```
EnableUnoccupiedOpsOFF >= EnableMorningWarmupON
```

The `EnableUnoccupiedOpsOFF` event must occur at the same time or after `EnableMorningWarmupON` event.

```
EnableUnoccupiedOpsOFF <= EnableMorningWarmupON + 00:20
```

The `EnableUnoccupiedOpsOFF` event must occur before or no more than 20 minutes after the `EnableMorningWarmupON` event.

```
EnableUnoccupiedOpsOFF = EnableMorningWarmupON
```

The `EnableUnoccupiedOpsOFF` event must occur at the same time as the `EnableMorningWarmupON` event.

```
EnableUnoccupiedOpsOFF !=EnableMorningWarmupON
```

The `EnableUnoccupiedOpsOFF` event must not occur at the same time as `EnableMorningWarmupON` event.

The existence equation – If there is an event that must appear in every schedule, you should specify it with the existence equation. This equation consists of the event name and only the event name.

Logical equations - Simple equations may be logically strung together with the “&” and “|” operators, signifying logical AND and OR, respectively. In general, this should be avoided, as the same effect can usually be achieved by writing more constraint equations, and each one can have a more specific error string.

The Schedule Constraint Language

By default the units used in these equations are the ones native to the relevant network variable types. See *Converting to US Units*, later in this chapter, for information on using non-SI units.

The following notation will be used in the examples and grammar:

Operands

e is the time at which the event e occurs and is in the range 0:00 < e< 23:59

For example, if e is the event MorningWarmup ON @ 6:00 AM, then e is 6:00.

t is a time variable in the range 0:00 < t < 23:59. The time separator character is specific to the locale of the PC (e.g., ":" for US users and "." for French).

Operators

Relational Operators

$t_1 = t_2$	t_1 must be the same time as t_2
$t_1 != t_2$	t_1 must not be the same time as t_2
$t_1 < t_2$	t_1 must occur before t_2
$t_1 <= t_2$	t_1 may **not** occur after t_2
$t_1 > t_2$	t_1 must occur after t_2
$t_1 >= t_2$	t_2 may **not** occur after t_1

Arithmetic Operators

$t_1 + t_1$	The sum of t_1 and t_2. If this sum is greater than 24:00, 24:00 will be subtracted from the result.

Boolean Operators

exp_1 & exp_1	If exp_1 and exp_1 are both true, the expression will evaluate as true. If either exp_1 or exp_1 are false, the expression will evaluate as false.
$exp_1 \mid exp_1$	If either exp_1 or exp_1 are true, the expression will evaluate as true. If both exp_1 and exp_1 are false, the expression will evaluate as false.

Existence Expression

e_1	e_1 must exist in all Schedules (the existence expression may not be combined with any other construct in a Schedule Constraint Equation).

Examples:

Example 1:

$e_1 < 6:00$ event e_1 must happen before 6am

$e_1+0:10 <= e_2$ event e_1 must happen at least 10 minutes before event e_2

Example 2

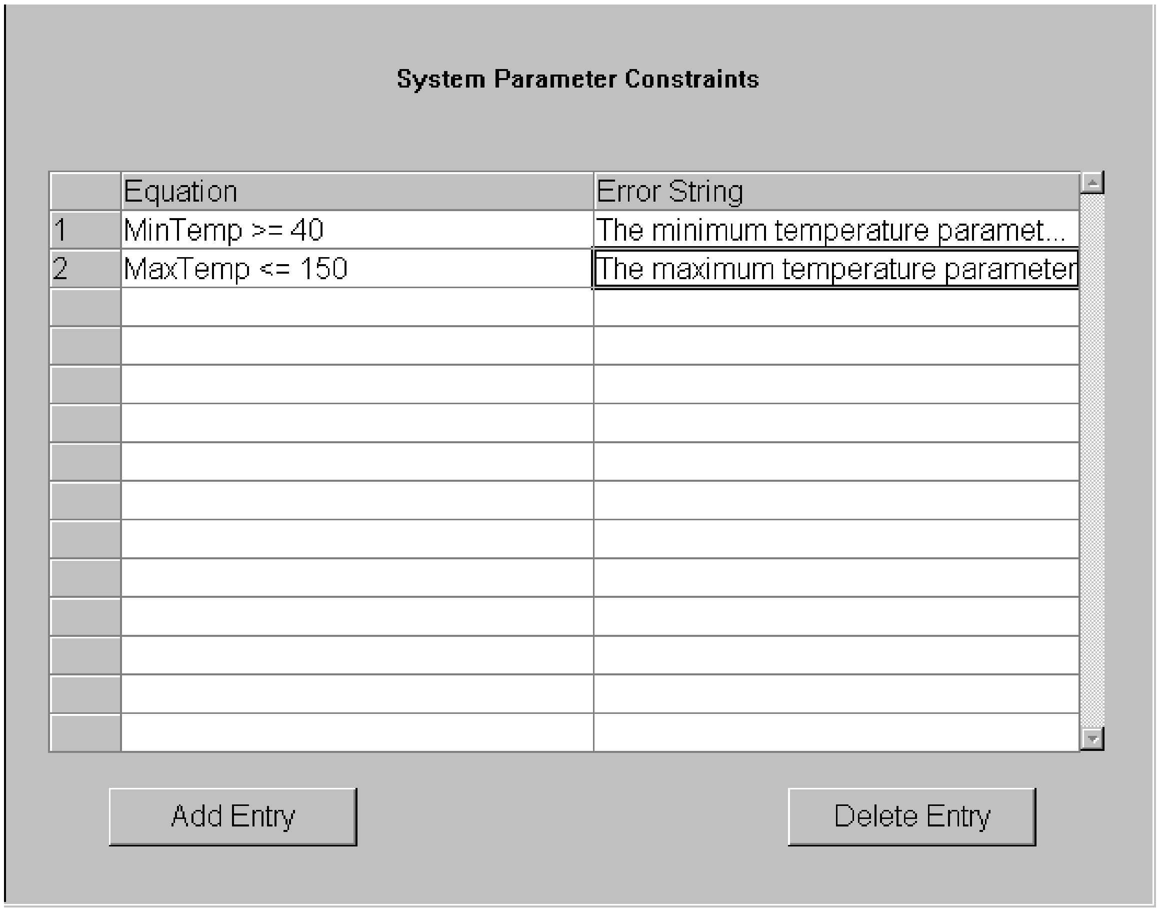
$e_1 < 6{:}00$ event e_1 must happen before 6 AM

$6{:}00 <= e_1$ & $e_1 <= 7{:}00$ event e_1 must happen between 6 and 7 AM

$e_1 < e_2$ & $e_2 < e_3$ & $e_3 < 8{:}00$ event e_1 must precede e_2 which must precede e_3. All must happen before 8am

Adding Parameter Constraints

The `Parameter Constraints` button will only be available if Schedule Keeper was opened in super user mode and one or more system parameters were defined in *Schedule Maker's* `Name I/O` tab. Click the `Parameter Constraints` button to open the following window:



	Equation	Error String
1	MinTemp >= 40	The minimum temperature paramet...
2	MaxTemp <= 150	The maximum temperature parameter

Use this window to create, view, modify, and delete constraints for the values set in the `System Parameters` window and error messages which will be displayed if these constraints are violated. Only parameters which were defined in the *Schedule Maker's* `Name I/O` tab can be limited.

In the example above, two parameter constraints have been set. The first states that the `MinTemp` parameter must be greater than or equal to 40, and the second states that the `MaxTemp` parameter must be less than or equal to 150.

To add a parameter constraint, click `Add Entry` and fill in the `Equation` and `Error String` columns for the new row. To delete a schedule constraint, select the constraint to be deleted and click `Delete Entry`.

System Parameter Constraint Equations

System parameter constraint equations are used to create site-specific limitations in the way that Schedule Keeper users may modify system parameters.. If the equation evaluates to FALSE when schedule constraints are evaluated, the constraint error string will be displayed and the modification will not be accepted.

Legal equations include:

The comparison equation – if you want to compare one event time to another, all of the standard comparison operations are available. Examples include:

```
MaxTemp < 30
```

The `MaxTemp` parameter must be less than 30.

```
MaxTemp > MinTemp + 50
```

The `MaxTemp` parameter must exceed the `MinTemp` parameter by more than 50.

```
MaxTemp >= MinTemp
```

The `Mextemp` parameter must be greater than or equal to the `MinTemp` parameter.

```
MaxTemp != MinTmep
```

The `MaxTemp` parameter must not to equal to the `MinTemp` parameter.

Logical equations - Simple equations may be logically strung together with the “&” and “|” operators, signifying logical AND and OR, respectively. In general, this should be avoided, as the same effect can usually be achieved by writing more constraint equations, and each one can have a more specific error string.

System Parameter Constraint Language

This section specifies the syntax for the system parameter constraint language and provides some examples of its use. The language is essentially the same as the time constraint language.

By default the units used in these equations are the ones native to the relevant network variable types. See *Converting to US Units*, later in this chapter, for information on using non-SI units.

The following notation will be used in the examples and grammar:

Operands

c is the value of the named constant c.
For example, space1TempThreshold might be 22.1 degrees C

v is a floating point variable

Operators

Relational Operators and their semantics

$v_1 = v_2$	v_1 must be the same as v_2
$v_1 != v_2$	v_1 must not be the same as v_2
$v_1 < v_2$	v_1 must be less than v_2
$v_1 <= v_2$	v_1 must be less than or equal to v_2
$v_1 > v_2$	v_1 must be greater than v_2
$v_1 >= v_2$	v_1 must be greater than or equal to v_2

Arithmetic Operators

$v_1 + v_2$	the floating point sum of v_1 and v2

Boolean Operators

exp_1 & exp_1	If exp_1 and exp_1 are both true, the expression will evaluate as true. If either exp_1 or exp_1 are false, the expression will evaluate as false.
exp_1 | exp_1	If either exp_1 or exp_1 are true, the expression will evaluate as true. If both exp_1 and exp_1 are false, the expression will evaluate as false.

Examples:

Example 1

$c_1 < 25.4$ the value of c_1 must be less than 25.4

$c_1+10 <= c_2$ the value of c_1 must be 10 less than the value of c_2

Example 2

$c_1 < 25.4$ the value of c_1 must be less than 25.4

$20.5 <= c_1$ & $c_1 <= 26.5$ the value of constant c_1 must lie between 20.5 and 26.5

$c_1 < c_2$ & $c_2 < c_3$ & $c_3 < 29.2$ the value of c_1 must be less than the value of c_2 which must less than the value of c_3. All must be less than 29.2

Saving and Exiting

To save the changes made using Schedule Keeper, click the `Save Changes` button or select `Save` from the `File` menu. This will download the changes to the SCH-10 device, if attached, and return Schedule Keeper to the start-up window. To exit Schedule Keeper without saving the changes, select `Exit` from the `File` menu.

Converting to US Units

By default, the values used for comparisons in System Parameter Constraint equations are the native values used by the network variables. For example, a network variable of type `SNVT_temp_f` would use degrees Celcius as its units.

To change the format type of the network variables used in these equations, follow these steps:

1. Open the launch file which is to use different unit formats with a text editor such as Notepad.
2. Each System Parameter has the following format:

```
SystemParameter=<name>,<index to CP array>[,<format name>]
```

For Example:

```
SystemParameter=MaxTemp,3
```

3. Add the format names to the system parameters which are to have a different format. For example, to have the parameter `MaxTemp` of SNVT type `SNVT_temp_f` above formatted in degrees Farenheit, you would modify the string to read as follows:

```
SystemParameter=MaxTemp,3,SNVT_temp_f
```

The format specifier `SNVT_temp_f` will find the current default format for `SNVT_temp_f`, which will be `SNVT_temp_f#US` if you installed US units format files, or `SNVT_temp_f#SI` if you installed System Internationale units format files.

12

The Type Translator Functional Block: Application and Plug-in

This chapter describes how to configure a Type Translator functional block using the LonPoint Plug-in.

The Type Translator Functional Block

The Type Translator functional block translates network data from one network variable type to another network variable type. This is useful for connecting network variables on devices from different vendors that use types that are not compatible with each other or with the other LonPoint functional blocks. The Type Translator allows you to set up scaling and mapping for the translation. The following figure and tables summarize the inputs and outputs of the Type Translator functional block:

Data_In
Data_Out
TRANS- 1

Input Network Variables

Default name	Default type	Description
Data_In	None (changeable)	The input network variable to be processed by the Type Translator.

Output Network Variables

Default name	Default type	Description
Data_Out	None (changeable)	The output network variable which has been processed by the Type Translator.

The Type Translator functional block has one input and one output network variable, both with changeable types. The type of these network variables may be any network variable type that is a floating-point, eight- or sixteen-bit fixed-point, enumeration (enum), or SNVT_switch. The types for both network variables must be set before this functional block can be connected.

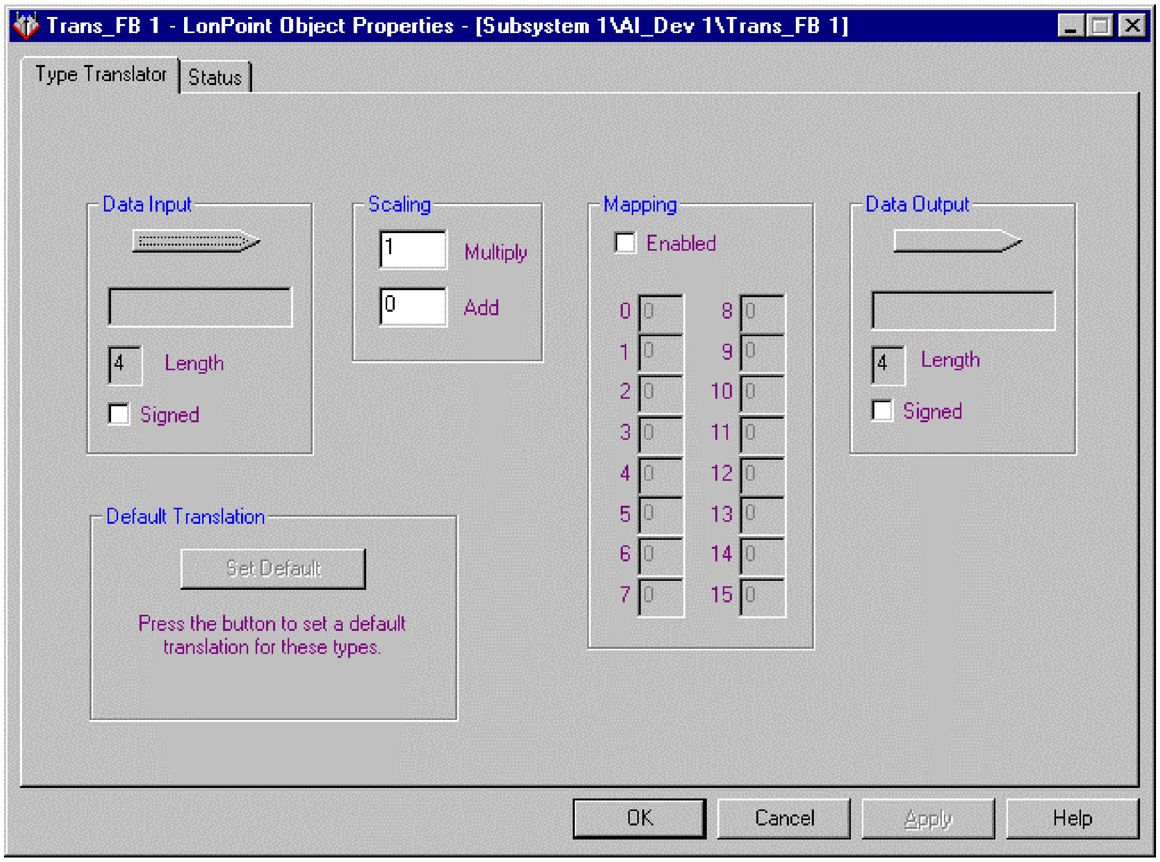
It is recommended that the types of the Data_In and Data_Out network variables only be changed using this plug-in. Changing the type of these network variables with another tool, such as the LonMaker Browser, can cause unexpected behavior. This is because the type translator application requires configuration properties associated with these network variables to be set correctly when the network variable types are changed, and the LonMaker Browser and other tools do not make these changes.

Configuring a Type Translator Using the LonPoint Plug-in

Right-click a Type Translator functional block shape and select *Configure* from the shortcut menu to open the Type Translator window of the LonPoint plug-in. This window contains two tabs, Type Translator and Status.

Type Translator

The Type Translator tab appears as follows:



Data Input: Length 4; Signed ☐

Scaling: Multiply 1; Add 0

Mapping: Enabled ☐

0	0	8	0
1	0	9	0
2	0	10	0
3	0	11	0
4	0	12	0
5	0	13	0
6	0	14	0
7	0	15	0

Data Output: Length 4; Signed ☐

Default Translation: Press the button to set a default translation for these types.

The Data Input and Data Output fields determine the type, length, and signed attribute of the input and output network variables, respectively. The first time a Type Translator functional block is configured, the fields that show the current types will be blank, and you must select valid types before the functional block can be used. Click the Data Input or Data Output network variable buttons to open the network variable type dialog and select the network variable type of the input or output.

The network variable type dialog allows you to select the network variable type from the list of available network variable types. To view and select user network variable types, deselect the *Standard Network Variable Type* option and select the appropriate user type file. Only floating point, fixed point, enumerations, and `SNVT_switch` types are recognized by the type translator. Click the `Set Default`, `OK`, or `Apply` button to set the default translation for the specified types.

The type translator tab contains the following fields:

Length	Indicates the network variable length in bytes. This field is read-only.
Signed	Determines whether the data is treated as a signed value. The Signed option for the output network

	variable type also controls whether the data in the mapping table is interpreted as signed or unsigned.
Scaling	Sets scaling for network variable values. The value read from the input network variable is multiplied by the `Multiply` field value and then added to the `Add` field value. When an input or output type is changed, the scaling values revert to their defaults. Once the network variable types of the input and output network variables have been set, changing the type of either of them resets this property to its defaults. The default for the Add field is 0; the default for the Multiply field depends on the scaling factors of the input and output network variable types. The type translator plug-in does not display any units. Any values entered must correspond to the native units of the corresponding network variable type (normally SI units), if there are any. The Scaling values are associated with the `Data_In` network variable.
Mapping	Maps one value to another value. To use mapping, you must select the `Enabled` option. If this option is selected, each input value (0-15) will be mapped to the output value in the associated field. If the input value is outside of the range 0-15, then the mapping corresponding to the nearest value is used. Fractional input values will be truncated to compute the index (i.e. 4.99 will be truncated to 4; if you wish to have values rounded to the nearest integer value, use the Scaling option to add .49 to each value). The map table entries are entered and displayed as signed byte values (-128 to 127). However, if the `Signed` setting of Data Output is cleared, these values will be interpreted as unsigned byte values (0 to 255), so the values –128 to –1 internally map to the values 128 to 255. Once the network variable types of the input and output network variables have been set, changing the type of either of them disables mapping. The type translator plug-in does not display any units. Any values entered must correspond to the native units of the corresponding network variable type (normally SI units), if there are any. The Mapping values are associated with the `Data_Out` network variable
Set Default	When this option is set, `Scaling` and `Mapping` values are automatically selected to convert the `Data_In` network variable type to the `Data_Out` network variable type. This is done automatically every time you change a type, so normally you will

not have to set the `Scaling` or `Mapping` values yourself.

Status

This tab allows you to view and change the status of a Type Translator functional block. See *Status* in Chapter 2 for more information.

13

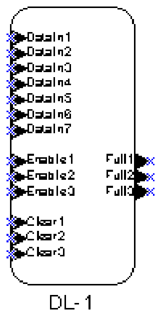
The Data Logger Functional Block: Application, Plug-in, and Utility

This chapter describes how to configure a Data Logger functional block using the LonPoint Plug-in and how to use the LonPoint Data Logger Utility to extract the logged data into a text file and export the text file to a trending application.

The Data Logger Functional Block

The Data Logger functional block allows you to control the DL-10 device's data logging ability. The DL-10 hardware platform is identical to the SCH-10 hardware platform, but the two devices run different applications. You can convert an SCH-10 device into a DL-10 device by downloading a DL-10 application into it, and vice versa.

The DL-10 application filters and logs data into up to three logs in a DL-10 device. The LonPoint Data Logger Utility allows you to upload these logs from the DL-10 device to a comma delimited text file and export the file to a trending application such as Wonderware InTouch. See *The LonPoint Data Logger Utility*, later in this chapter, for more information. The following figure and tables summarize the inputs and outputs of the Data Logger functional block:



Input Network Variables

Default name	Default type	Description
DataIn(1-7)	`SNVT_str_asc` (changeable)	One of seven data input network variables. Multiple network variables may be connected to each input. Any of the seven inputs may be stored in any one of the three logs.
Enable(1-3)	`SNVT_switch`	One of three Enable network variables corresponding to the three logs. If the Enable network variable for a log is set to Off, the log will not record data. If the Enable network variable for a log is set to On, the log will function as normal.
Clear(1-3)	`SNVT_switch`	One of three Clear network variables corresponding to the three logs. If the Clear network variable for a log is set to On, the log will be cleared.

Output Network Variables

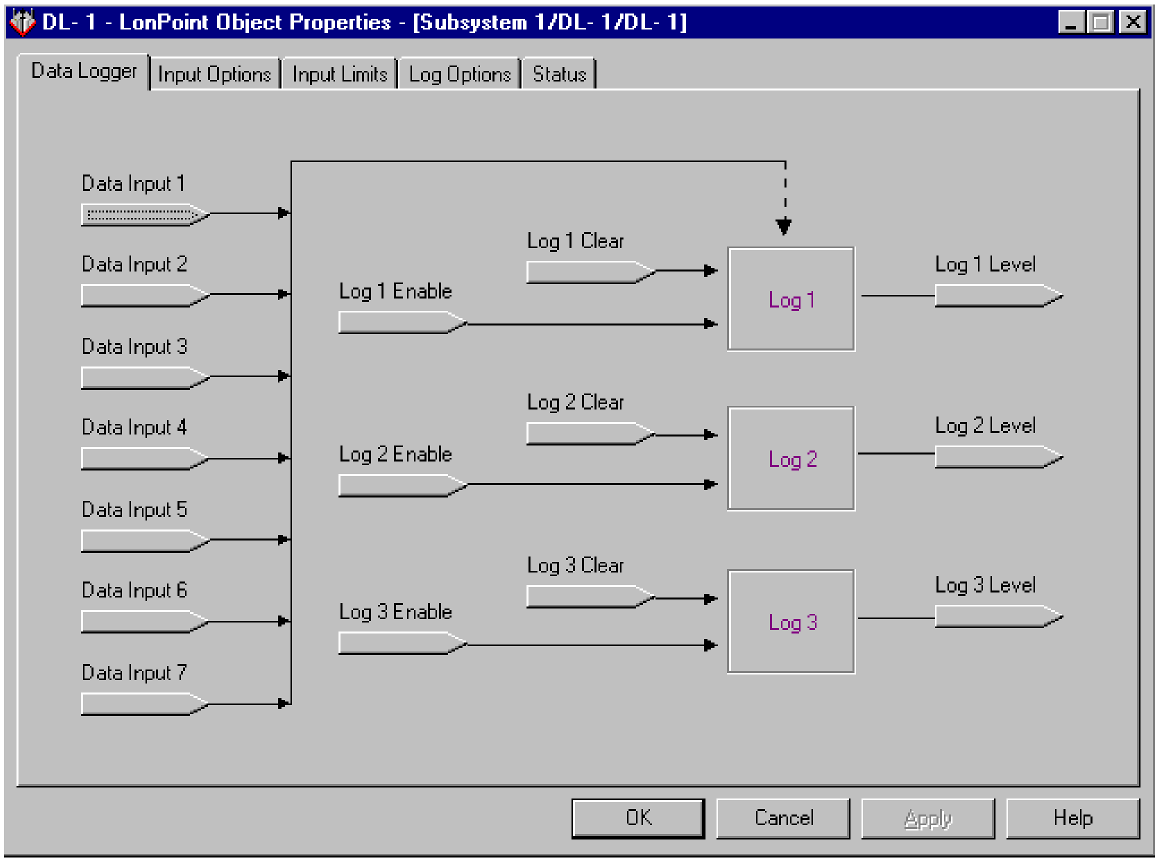
Default name	Default type	Description
Full(1-3)	`SNVT_lev_cont`	One of three Full network variables. This network variable contains a percentage value indicating how full the corresponding log is.

Configuring the Data Logger with the LonPoint Plug-in

Right-click a Data Logger functional block and select *Configure* from the shortcut menu to open the LonPoint Plug-in. The Data Logger window of the LonPoint Plug-in contains the following tabs: `Data Logger`, `Input Options`, `Input Limits`, `Log Options`, and `Status`.

Data Logger

This tab appears as follows:

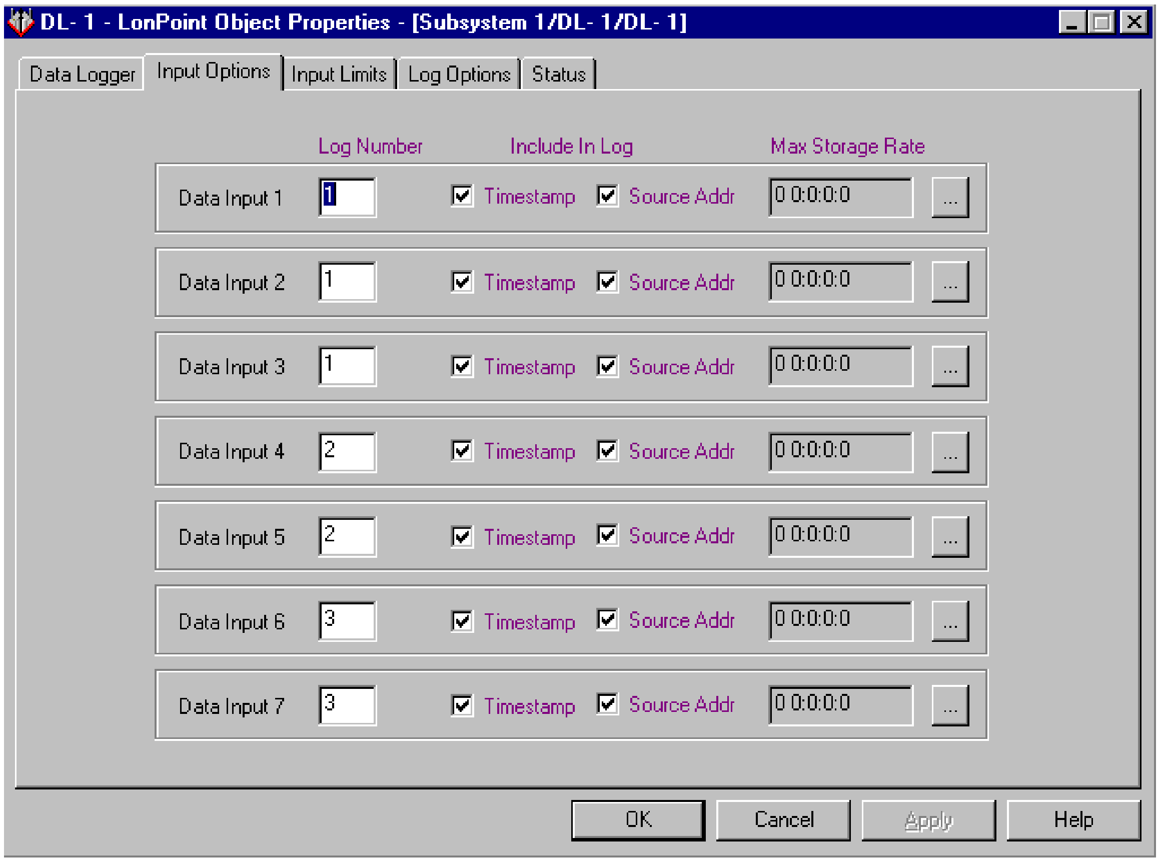


This tab displays the flow of data through the DL-10 application. You can change the network variable types of the 7 Data Inputs as described in

Changing Network Variable Types in Chapter 1. There are no restrictions on the network variable types allowed.

Input Options

This tab appears as follows:



DL- 1 - LonPoint Object Properties - [Subsystem 1/DL- 1/DL- 1]

Data Logger | Input Options | Input Limits | Log Options | Status

	Log Number	Include In Log		Max Storage Rate
Data Input 1	1	✓ Timestamp	✓ Source Addr	0 0:0:0:0
Data Input 2	1	✓ Timestamp	✓ Source Addr	0 0:0:0:0
Data Input 3	1	✓ Timestamp	✓ Source Addr	0 0:0:0:0
Data Input 4	2	✓ Timestamp	✓ Source Addr	0 0:0:0:0
Data Input 5	2	✓ Timestamp	✓ Source Addr	0 0:0:0:0
Data Input 6	3	✓ Timestamp	✓ Source Addr	0 0:0:0:0
Data Input 7	3	✓ Timestamp	✓ Source Addr	0 0:0:0:0

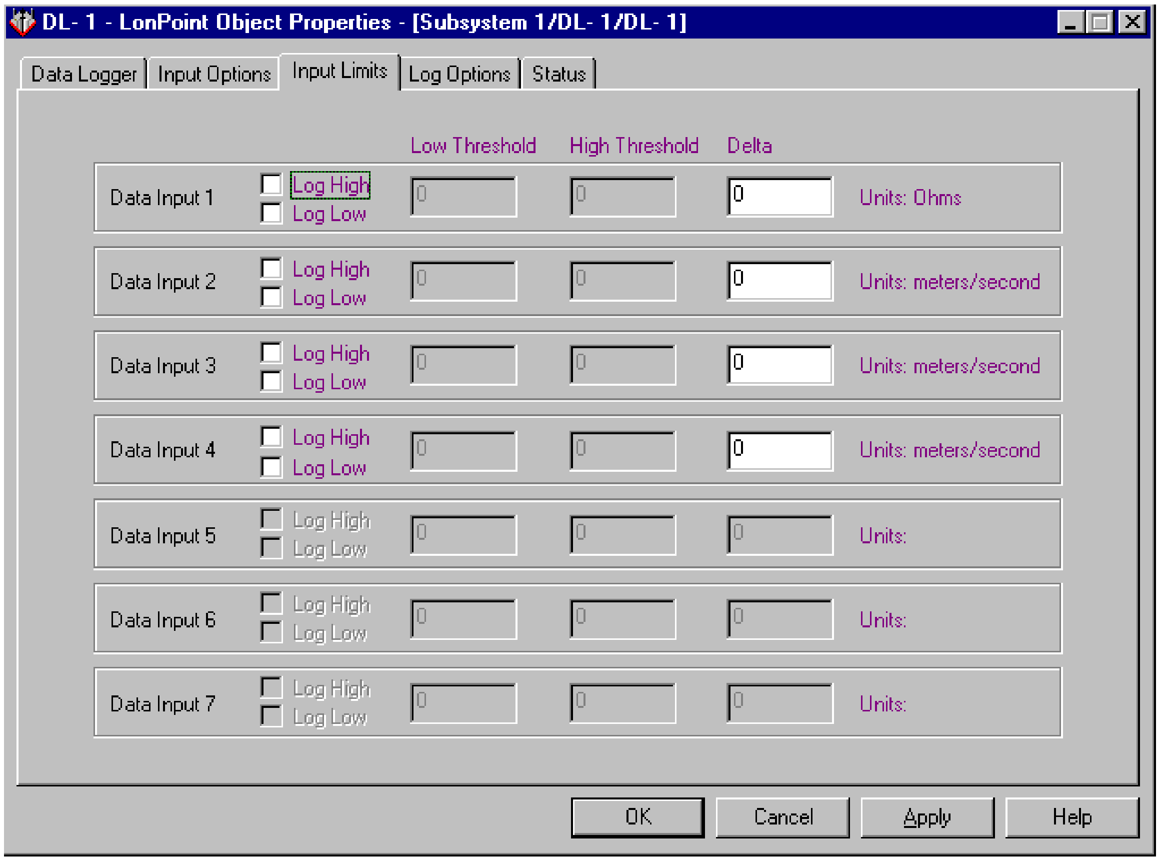
This tab contains the following fields:

Log Number	Specifies which log the data received on the corresponding Data Input will be sent to. The log may perform additional filtering as described in the `Input Limits` tab. This value must be an integer from 0 to 3. Setting this value to 0 disables logging for the corresponding input.
Timestamp	Specifies that a timestamp will be included with every piece of data logged from the corresponding Data Input.
Source Addr	Specifies that the Subnet/Node address of the device that the data was received from will be included with every piece of data logged from the corresponding data input. See *Tagnames*, later in this chapter.
Max Storage Rate	Specifies the minimum amount of time between logging data. This allows you to take data samples at

a specified interval. Click the button to the right of the corresponding Data Input to change the maximum storage rate for that input.

Input Limits

This tab appears as follows:



DL- 1 - LonPoint Object Properties - [Subsystem 1/DL- 1/DL- 1]

Data Logger | Input Options | Input Limits | Log Options | Status

		Low Threshold	High Threshold	Delta	
Data Input 1	☐ Log High ☐ Log Low	0	0	0	Units: Ohms
Data Input 2	☐ Log High ☐ Log Low	0	0	0	Units: meters/second
Data Input 3	☐ Log High ☐ Log Low	0	0	0	Units: meters/second
Data Input 4	☐ Log High ☐ Log Low	0	0	0	Units: meters/second
Data Input 5	☐ Log High ☐ Log Low	0	0	0	Units:
Data Input 6	☐ Log High ☐ Log Low	0	0	0	Units:
Data Input 7	☐ Log High ☐ Log Low	0	0	0	Units:

This tab allows you to set up filtering for numeric data. You can set separate filtering options for each of the 7 Data Inputs network variables. Only floating point, fixed point, or enumerated network variables can be filtered. If any other network variable type is specified for an input, the corresponding filter area will be grayed out.

If you change any Data input network variable to a type that cannot be filtered, you should then select this tab and click `Apply` to ensure that filtering is disabled for that network variable.

For each of the 7 Data Input network variables, you may set the following fields:

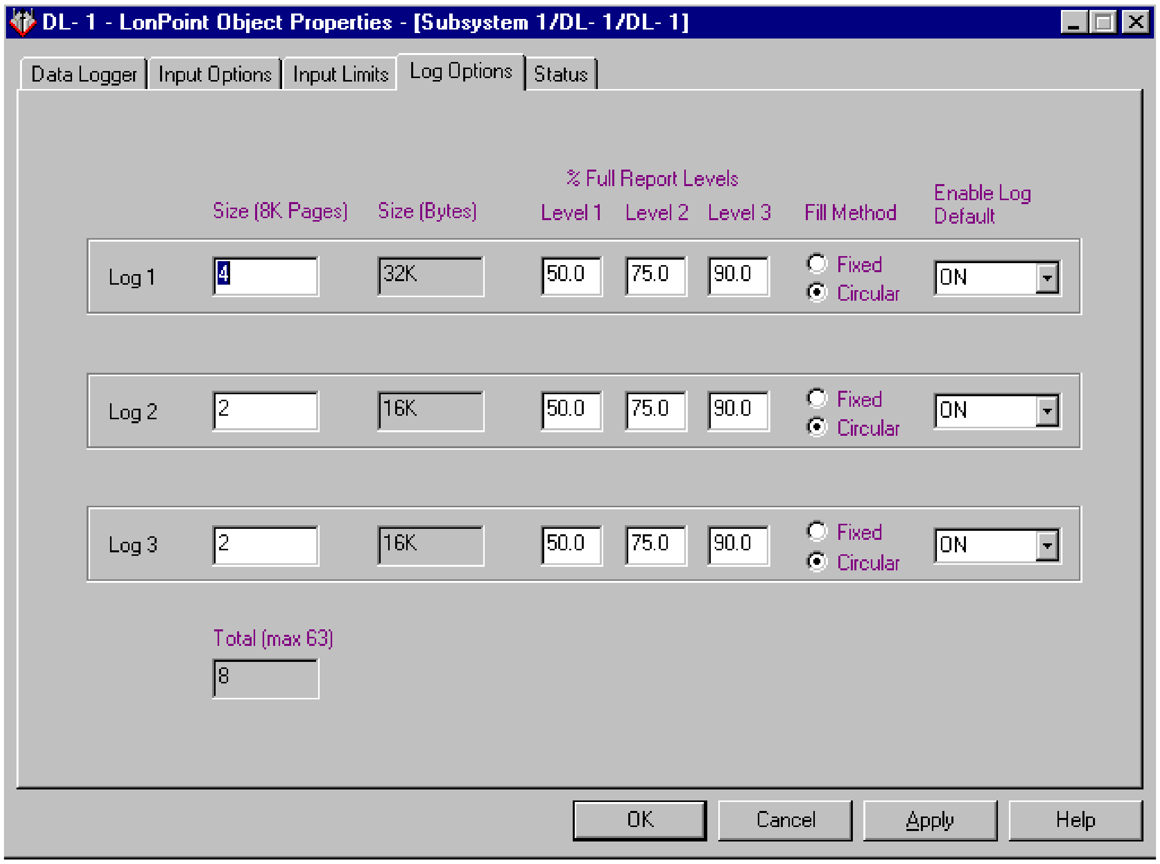
Log Low	Causes incoming data to be filtered against the *Low Threshold* field before being logged.
Log High	Causes incoming data to be filtered against the *High Threshold* field before being logged.

Low Threshold	This threshold will only be used if the `Log Low` option is selected. Only data that is *less than or equal* to the value in this field will be logged. Upon crossing this threshold data is logged once more after the value crosses back from the threshold. This way the point at which a value returned from an out of limits condition will be recorded.
High Threshold	This threshold will only be used if the `Log High` option is selected. Only data that is *greater than or equal to* the value in this field will be logged. Upon crossing this threshold data is logged once more after the value crosses back from the threshold. This way the point at which a value returned from an out of limits condition will be recorded.
Delta	Data is logged when the difference between the last point entered into the log is greater than or equal to the value entered.

For example, if you want to log data points between 75.0 and 150.0, select the Log High and Log Low options, set the *Low Threshold* field to 150.0, and the *High Threshold* field to 75.0.

Log Options

This tab appears as follows:



DL- 1 - LonPoint Object Properties - [Subsystem 1/DL- 1/DL- 1]

	Size (8K Pages)	Size (Bytes)	% Full Report Levels Level 1	Level 2	Level 3	Fill Method	Enable Log Default
Log 1	4	32K	50.0	75.0	90.0	Fixed [✓ Circular]	ON
Log 2	2	16K	50.0	75.0	90.0	Fixed [✓ Circular]	ON
Log 3	2	16K	50.0	75.0	90.0	Fixed [✓ Circular]	ON

Total (max 63): 8

This tab allows you to determine several options that relate to the logs in the DL-10 device. This tab contains the following fields:

Size (8K Pages)	Specifies the number of 8K pages used by this log. The three logs can have a maximum of 63 pages between them. The total number of pages is shown in the Total field.
Size (Bytes)	Displays the size of the log in bytes. This field is read-only and reflects the number of pages allocated to the log in the `Size (8K Pages)` field.

The data log memory is divided into 8-byte records, so there are 64,512 available records. Each record contains a one-byte value that describes the contents of the record and up to 7 bytes of data. The number of 8 byte records used to log a network variable value depends on the size of the network variable being logged and whether the Source Addr or Timestamp options are set for the corresponding `Data` network variable in the `Input Options` tab. The size of the points in each record are as follows:

Network variable index (1 byte): Identifies which of the input network variables to the data logger received the update. Every logged point contains a network variable index

Source address (2 bytes): Identifies the subnet/node address of the device the logged data was received from. This information is included with each record if the `Source Addr` option is selected for the corresponding `Data` network variable.

Time stamp (7 bytes): Identifies the time and date when this record was logged. This is derived from the DL-10 device's Real Time Clock functional block (see *Setting the DL-10's Real Time Clock*, later in this chapter). This information is included with each record if the `Timestamp` option is selected for the corresponding `Data` network variable.

Network variable value (The number of bytes depends on the network variable type, may be from 1 to 31 bytes): The network variable length in bytes can be acquired from the *SNVT Master List* (if the network variable is a SNVT) or from the `Attributes` tab of the *Network Variable Properties* dialog accessed through the LonMaker Browser (see Chapter 9 of the *LonMaker for Windows User's Guide*). Every logged point contains a network variable value.

The following table describes the number of records which are used to log a point for any network variable length:

Network Variable Length (bytes)	**No Source Address or Timestanp (8 byte records)**	**Source Address Only (8 byte records)**	**Timestamp Only (8 byte records)**	**Source Address and Timestamp (8 byte records)**
1	1	1	2	2
2	1	1	2	2
3	1	1	2	2
4	1	1	2	2
5	1	2	2	3
6	1	2	2	3
7	2	2	3	3
8	2	2	3	3
9	2	2	3	3
10	2	2	3	3
11	2	2	3	3
12	2	3	3	4
13	2	3	3	4
14	3	3	4	4
15	3	3	4	4
16	3	3	4	4
17	3	3	4	4
18	3	3	4	4
19	3	4	4	5
20	3	4	4	5
21	4	4	5	5
22	4	4	5	5
23	4	4	5	5
24	4	4	5	5
25	4	4	5	5
26	4	5	5	6
27	4	5	5	6
28	5	5	6	6
29	5	5	6	6
30	5	5	6	6
31	5	5	6	6

The maximum number of samples that can be logged can be calculated using this table.

It is possible to log network variables of different sizes into the same data log. If this is done, it may not be possible to determine exactly the number of samples that may be logged, but you can determine the worst case by calculating the number for the largest network variable size.

%Full Report Levels	Specifies when the three Full network variables will be sent on the network. When a new level is reached, the Full network variable corresponding to the log will be sent on the network. This only has an effect if Bound monitoring is being used. See *Monitoring Options*, later in this chapter, for more information.
Fill Method	Specifies the behavior of the corresponding log if its capacity is reached. The Fixed option causes new data to be lost. The Circular option causes the log to throw out the oldest logged data to make room for the new data.
Enable Log Default	Specifies whether the corresponding log will be Enabled or not if its Enable network variable is not connected or has not received any updates since the last reset.

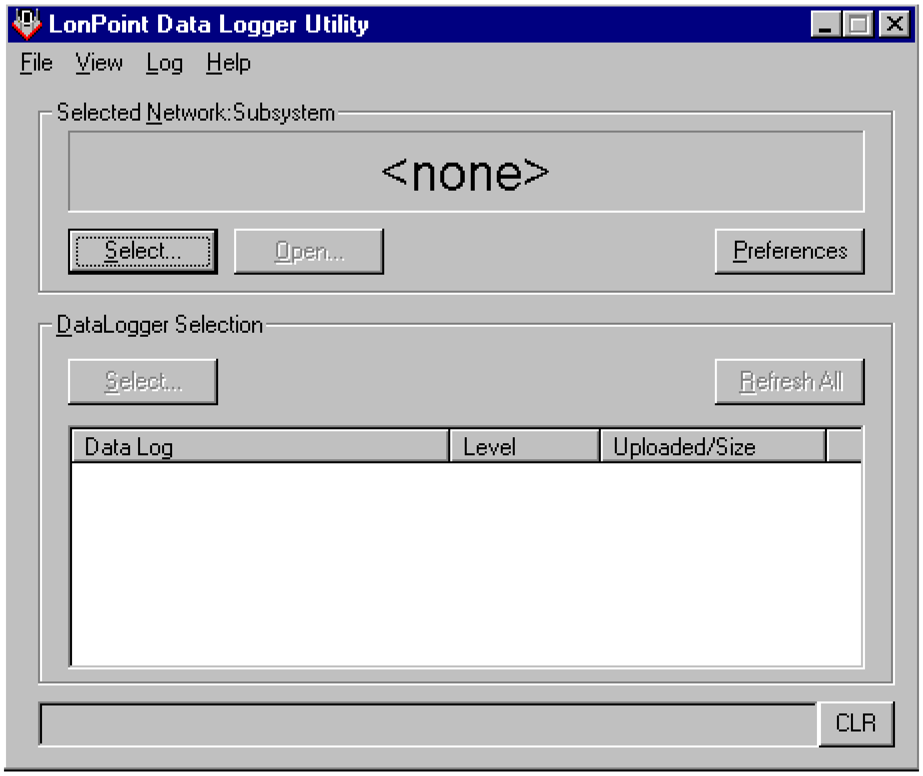
Status

This tab allows you to view and change the status of a Data Logger functional block. See *Status* in Chapter 2 for more information.

The LonPoint Data Logger Utility

The LonPoint Data Logger Utility is an LNS application that allows you to upload data logs from a DL-10 device into a binary log file (.BIN extension) , and to export the binary log file to a comma separated value (CSV) text file (.CSV extension). See *CSV File Format*, later in this chapter, for more information on CSV files.

To start the Data Logger Utility, click the Windows Start button and select *LonPoint Data Logger Utility* from the Echelon LonPoint Software program menu. This will display the following window:



This window allows you to open an existing network and subsystem, select one or more of any LonPoint DL-10 Data Loggers that are part of the system, and enter a mode where the data logs are monitored for their output levels.

You may also manually perform operations such as uploading the data log, exporting the uploaded binary log file, and clearing the data log in a DL-10 device.

This window contains the following fields and buttons:

Network: Select	Click this button to select the network containing the DL-10 device from a list of existing networks.
Open	Click here to open the selected network and go onnet. You can optionally select a network interface and subsystem. Once the network is opened a list of available DL-10 Data Loggers will be shown under `Data Logger Selection`.
Preferences	You may select this before or after opening the network. This brings up the preferences property tabs (see *Setting Preferences*, later in this chapter, for more information).
DataLogger: Select	Click this button to open a DL-10 Data Logger selection dialog. One or more DL-10 devices may be selected. The utility will establish communications

	with the DL-10 devices and the selected logs will be displayed in the Data Log list area.
Data Log list	This list shows each of the data logs for all selected DL-10s; typically 3 logs per DL-10. The Level field displays what the log level is, the Uploaded/Size field shows how many bytes of data have been uploaded and how large the intermediate binary file is. To select a data log, click its *Data Log* field. To set options for a data log, double click its *Data Log* field to open a `Log Preferences` dialog. To manually control a data log, right-click its *Data Log* field and select a command from the shortcut menu (see *Log Menu Commands*, later in this chapter).
Refresh All	Click this button to poll and update all log level fields in the `Data Log list`.
CLR	Click this button to clear the status bar at the bottom of the window. The status bar displays any network errors encountered during the session.

Menu Commands

The following commands are available in the menu bar of the Data Logger Utility:

File Menu

Exit	Exits the utility.

View Menu

Show NI Select	Opens the network interface selection dialog when a network is selected using the network *Select…* button. If this option is not selected, the default network interface will be selected.
Show Subsystem Select	Opens the subsystem selection dialog when a network is selected using the *Select* button. If this option is not selected, the DL-10 devices from all subsystems in the network will be displayed when the network is opened using the *Select* button.

Log Menu

Upload Now	Uploads the selected data log to a binary log file (.BIN extension). This file is created in the same folder that the Data Logger Utility was run from, with a naming format of `LwDLogLN.bin`, where `L` is `0` for the first DL-10 device selected, and `1` for the second, and so on; and `N` is `1` for Log 1, `2` for Log 2, etc.
Export	Exports the binary log file for the selected data log. Once a data log has been uploaded, the binary log file

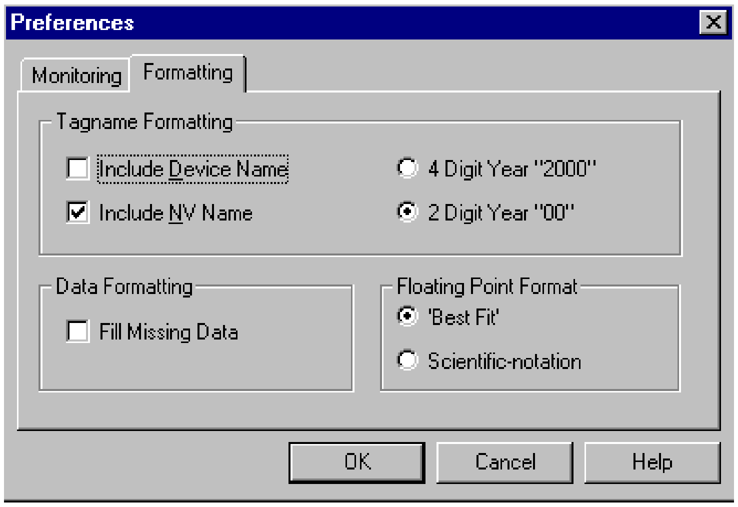
can be exported into a CSV text file. When a binary log file is exported it is stored in the same folder as the binary log file with the same base file name and a .CSV extension.

Once the binary log file is uploaded you may manually export it as many times as you wish, changing the formatting preferences (see *Formatting Preferences*, later in this chapter) in between export steps if needed.

Clear Log	Clears the selected data log in the DL-10 device using the Data Logger functional block's Clear network variable.
Clear Uploaded	Clears (empties) the binary file log for the selected data log.
Run Executable	Runs the command line specified by the `Execute Application` field of the *Log Preferences* tab for the selected log.
Preferences	Opens the Log Preferences dialog for the selected data log.

Formatting Preferences

To modify formatting preferences, click the `Preferences` button and select the `Formatting` tab. This tab appears as follows:



Tagname Formatting
- [ ] Include Device Name
- [✓] Include NV Name
- ○ 4 Digit Year "2000"
- ● 2 Digit Year "00"

Data Formatting
- [ ] Fill Missing Data

Floating Point Format
- ● 'Best Fit'
- ○ Scientific-notation

This tab controls the format of the Data Logger CSV text file. The format is discussed under *CSV File Format*, later in this chapter. The `Formatting` tab contains the following fields:

Include Device Name	Includes the device name as part of the tagname. If a LONMARK object is the source of the data then that object's name will also be included in the tagname.

Include NV Name	Adds the Network Variable name, if available, to the end of the tagname (see *Tagnames*, later in this chapter).
4 Digit / 2 Digit Year	Selects one of two year formats for timestamped data (see *Timestamp* in the *Input Options* tab of the Data Logger functional block).
Fill Missing Data	If selected, the previous value of any columns without updated values will be included in each data line.
Floating Point Format	Select `Best Fit` for a 5 digit plus decimal point display. Select `Scientific-notation` for a full mantissa plus exponent display.

Monitoring Preferences

The Data Logger Utility can obtain Network Variable updates from the Data Logger functional block's Full network variables using either *Polled* or *Bound* monitoring. Which one you should use depends on how you are using this applicaton.

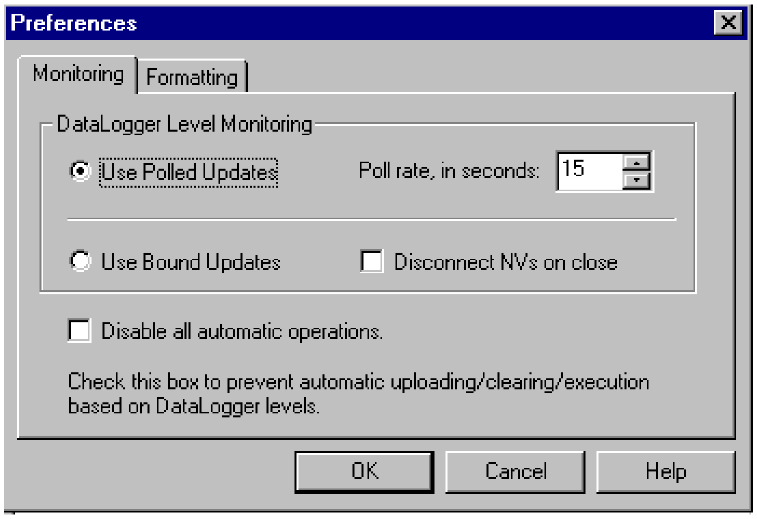
Polling the level output network variables in the DL-10 device results in a quicker start-up and shut-down of the Data Logger Utility in many cases. If you are running the Data Logger Utility as a scheduled job where you want to start it up, process any log files as required, and shut it down, then polling is the best choice. If the utility is left up and running is will periodically poll the `Full` network variables.

Bound network variables result in less network traffic if you plan on continuously running the Data Logger Utility as a background task. The levels are only updated in the utility when they cross the configurable Full % values set in the *Log Options* tab of the Data Logger functional block, and these updates occur immediately rather than at a poll interval. Bound network variables can also be used with remote hosts that are periodically attached via an SLTA-10 network interface. In this case, the binding should be left in place so a network variable update event from a DL-10 device will cause the SLTA-10 network interface to dial-out to the remote host.

The bound network variable case can add time to the utility's start-up since the binding must take place at that time if it is not already in place. If you do not select the *Disconnect NVs on close* option then start-up and shut-down times will be shorter since the connections are left in place.

When bound network variables are used, these connections can be re-used for each selected Network:Subsystem combination. When monitoring a new set of DL-10 devices from a given Network:Subsystem combination, network variables must be re-bound. In this case the previous connections are removed by the utility and a new set is created.

To modify monitoring preferences, click the *Preferences* button and select the *Monitoring* tab. This tab appears as follows:


Preferences

Monitoring | Formatting

DataLogger Level Monitoring

(●) Use Polled Updates   Poll rate, in seconds: 15

( ) Use Bound Updates   [ ] Disconnect NVs on close

[ ] Disable all automatic operations.

Check this box to prevent automatic uploading/clearing/execution based on DataLogger levels.

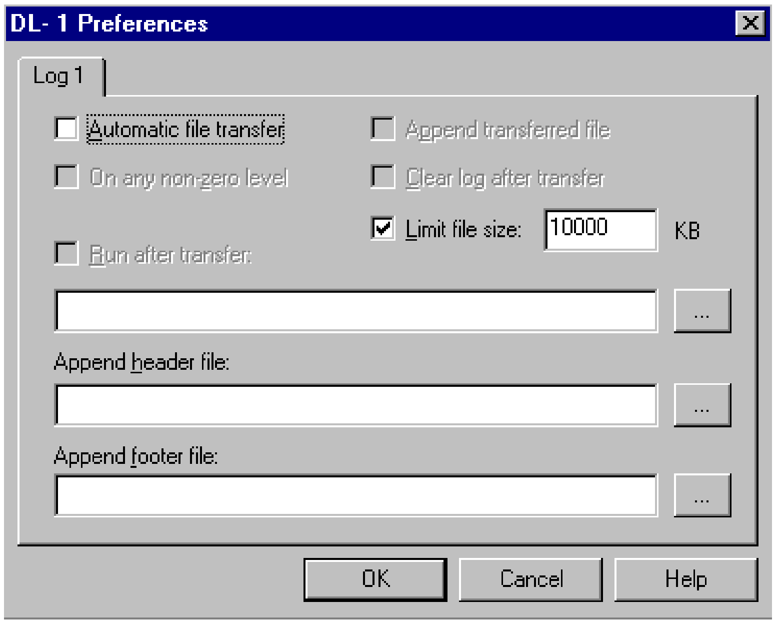
This tab contains the following fields:

Use Polled Updates	Causes the Data Logger's `Full` network variables to be polled.
Poll Rate	The rate, in seconds, that the utility will poll the network variables if the Use Polled Updates option is selected.
Use Bound Updates	Causes the Data Logger's `Full` network variables to be bound to the utility. In this case network traffic to the utility only occurs when the output level network variables cross any of the configured threshold values, or when the log is cleared.
Disconnect NVs on close	Disconnects network variables when the utility is shut down, or when the network is closed. If not selected, connections are is left in place.
Disable all automatic operations	Prevents any automatic file operations from occurring such as clearing, uploading, or command line execution based on Data Logger Full network variable levels. See *Log Preferences* in the next section for more information.

Log Preferences

Each individual data log has a level output network variable which is monitored by the utility. The utility may be configured to automatically perform a number of tasks when this level output value reaches configurable percentage full values.

To view these preferences, select a Log from the `Data Log list` and select *Preferences* from the `Log` menu. This will open the following dialog:


DL- 1 Preferences

Log 1

- [ ] Automatic file transfer
- [ ] Append transferred file
- [ ] On any non-zero level
- [ ] Clear log after transfer
- [x] Limit file size: 10000 KB
- [ ] Run after transfer:

Append header file:

Append footer file:

This dialog contains the following fields:

Automatic File Transfer	Causes the data log file to be uploaded to a binary log file and exported to a CSV text file. To prevent redundant automatic file transfers, a data log will not be updated after a previous upload until the data log is cleared by the utility, the utility is shut down, or the network is closed.
On any non-zero level	Starts automatic file transfer whenever the `Full` output value is non-zero instead of checking against the configurable percentage full values.
Append Transferred File	Appends uploaded data logs to the binary log file rather than creating a new binary log file. This only affects automatic file transfers. Manually transferred binary log files are always overwritten.
Clear log after transfer	Clears the Data Logger's data log after a successful automatic upload.
Limit file size	Limits the size of the binary log file to the specified size. This can be used to limit the actual transfer, and can be used to limit the file size for appended files.
Run after transfer	Executes the specified command line after the binary log file has been automatically uploaded and exported to a CSV text file. The specified command line typically contains the full path for an executable, plus any command line arguments. Quotes may be included as needed. The "%1" token (with or without

quotes) may be used to represent the full pathname of the generated CSV text file.

For example, select this option and enter the following command line to run the InTouch Historical Data Merge Utility on the output CSV text file (the first set of quotes are required because the path has spaces in it):

```
"C:\Program
Files\FactorySuite\InTouch\hdmerge.exe"
"%1"
```

Or, select this option and enter the following command line to start Excel with the output CSV text file. The /e option skips the Excel splash screen:

```
"C:\Program Files\Microsoft
Office\Office\excel .exe" /e "%1"
```

Append header file	Appends the specified file to the beginning of every CSV text file generated. Enter the full pathname of the file to be appended.
Append footer file	Appends the specified file to the end of every CSV text file generated. Enter the full pathname of the file to be appended.

CSV Text File Format

The LonPoint Data Logger Utility translates a DL-10 binary log file into a Comma Separated Values (CSV) text file for exporting to trending applications. The comma delimiter is configurable and is based on the Windows Regional Setting / List Separator property.

There are several factors which affect the layout of this file. A Data Logger CSV text file consists of a *tagname line* followed by a series of *data lines*.

Tagname Line

The first line in a CSV text file is the *tagname* line. This consists of textual column descriptors. The first two fields are `$Date` followed by `$Time`. Following these fields are the tagnames for the subsequent data columns.

A data column and a unique tagname is included for each unique data source identified in the binary log file. A unique data source is identified for each of the seven `Data` network variable inputs to the Data Logger functional block. In addition, unique data sources are identified for multiple network variables received on the same `Data` input if you have enabled logging of source addresses (see the *Input Options* tab of the Data Logger plug-in) and the network variable updates come from different source addresses. Multiple updates received on the same `Data` network variable from the same device will not be identified as unique data sources because they have the same source address.

The data source information will be included in the tagname for that network variable if the `Source Addr` option is selected in the Data Logger plug-in's `Input Options` tab. If the source network variable belongs to a LONMARK object then the object name is included, otherwise the device name is included. If the `Include NV Name` option is selected in *Formatting Preferences* (described earlier in this chapter), the source network variable name is included.

If the `Source Addr` option is cleared on the Data Logger plug-in's `Input Options` tab, the DL-10 device's `Data` input network variable's name will be used.

If a list separator is found anywhere in a tagname, it is substituted with an underscore.

Tagnames are built with object names separated by a colon ":". The tagname is formatted using one of the following:

LonMarkObjectName

LonMarkObjectName:OutputNVName

DeviceName:LonMarkObjectName:OutputNVName

DeviceName

DeviceName:OutputNVName

DL10Input:NVName

The order in which these unique data columns appear in the CSV text file is determined by the order in which the unique data sources first appear in the log file. For example, the first data column will be the first data source found in the log file.

Data Lines

The first two fields of any data line are the *Date* and *Time* fields. If there are no timestamped data points at all in the log file then an artificial date and time are presented as values ascending by one second per line, starting at 01/01/90 00:00:01.

A new data line is included for each of the following:

- Each unique timestamp in the data log.
- Each new data point for the same data source in the data log.

For example, if two unique data points appear with the same timestamp then that line will contain a single timestamp and two columns containing data. If two unique data points appear with different timestamps then two separate lines will be generated, one with one data point in one column, and another with the other data point in another column.

If the `Fill Missing Data` option is selected in the Data Logger Utility's *Formatting Preferences* page, a data point is included in every column of the data line. If you choose this option then previous data points will be used to fill out each column. If this is not the choice then the missing data is shown as an empty field. For example, the following data lines were generated with the `Fill Missing Data` option turned off:

```
08/04/98,10:44:30,2.380000e+002,1.527428e+003
08/04/98,10:44:30,2.390000e+002,
08/04/98,10:44:30,2.400000e+002,
```

The following data lines were generated with this option is turned on:

```
08/04/98,10:44:30,2.380000e+002,1.527428e+003
08/04/98,10:44:30,2.390000e+002,1.527428e+003
08/04/98,10:44:30,2.400000e+002,1.527428e+003
```

where the first line represents new data for the last column, and the second and third lines contain “old” data for the last column.

Data Point Formats

Data points are formatted based on the following rules:

If the network variable size is 1 byte then the data is treated as a signed integer.

If the network variable size is 2 bytes and the network variable type is `SNVT_switch`, the data is treated as such and displayed as `NN.N B` where `B` is ‘0’ or ‘1’. For any other 2 byte network variable type; the data is treated as a signed 2-byte integer.

If the network variable size is 4 bytes then the data is treated as a floating point value.

If the network variable size is 29 bytes and the network variable is of the `SNVT_alarm` type, the data is treated as such.

If the network variable size is 31 bytes and the network variable is of the `SNVT_str_asc` type, the data is displayed as a string.

Command Line Switches

When invoking the Data Logger Utility there are several command line switches that help automate the start up process:

/net Network Name	Opens the specified network.
/ifc Interface Name	Uses the specified network interface.
/sub Subsystem Name	Opens the specified subsystem.
/log Data Logger Name	Accesses the specified Data Logger. Multiple switches can be used to select more than one Data Logger.
/idle seconds	Causes the utility to shut down after seconds of idle activity.
/cmd Command File	Opens the specified file as a source of command line switches. This is a text file, and can have newlines for readability.

Setting the DL-10's Real Time Clock

The LonPoint DL-10 employs a battery backed Real Time Clock functional block. This object is the source for the optional time stamp that is included with data points in the DL-10's data logs.

There is no configuration plug-in for the Real Time Clock functional block. To set the time and date on the Real Time Clock, follow these steps:

1. Open the LonMaker network containing the DL-10 whose Real Time Clock you want to set.
2. If the DL-10 device does not have a Real Time Clock functional block shape in the LonMaker drawing, create one as described in *Creating a Functional Block* in Chapter 4 of the *LonMaker for Windows User's Guide*. You don't need to have this shape on the drawing in order to use the Data Logger functional block, but you will need it to set the Real Time Clock.
3. Right-click the Real Time Clock functional block shape and select `Browse` from the menu. This will start the *LonMaker Browser* (see Chapter 9 of the *LonMaker for Windows User's Guide*).
4. In the Browser click the `SetTime` network variable.
5. In the `Value` field on the Browser's button bar enter the date and time in the same format as it is shown:

   `YYYY/MM/DD HH:MM:SS`
6. Click the `Set Value` button to update the Real Time Clock to the date and time you entered.

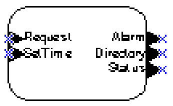
14

The Node Object Functional Block: Application and Plug-in

This chapter describes how to configure a Node Object functional block using the LonPoint Plug-in.

The Node Object Functional Block

Every LonPoint device contains one Node Object functional block. This functional block is used by LONMARK aware network tools such as the LonMaker tool to manage the device. The following figure and tables summarize the inputs and outputs of the Node Object functional block:



NO- 1

Input Network Variables

Default name	**Default type**	**Description**
Request	`SNVT_obj_request`	Allows the device to be placed in one of several modes. Use the LonMaker *Manage* command to change the device's mode.
SetTime	`SNVT_time_stamp`	Allows you to set the time to the device.

Output Network Variables

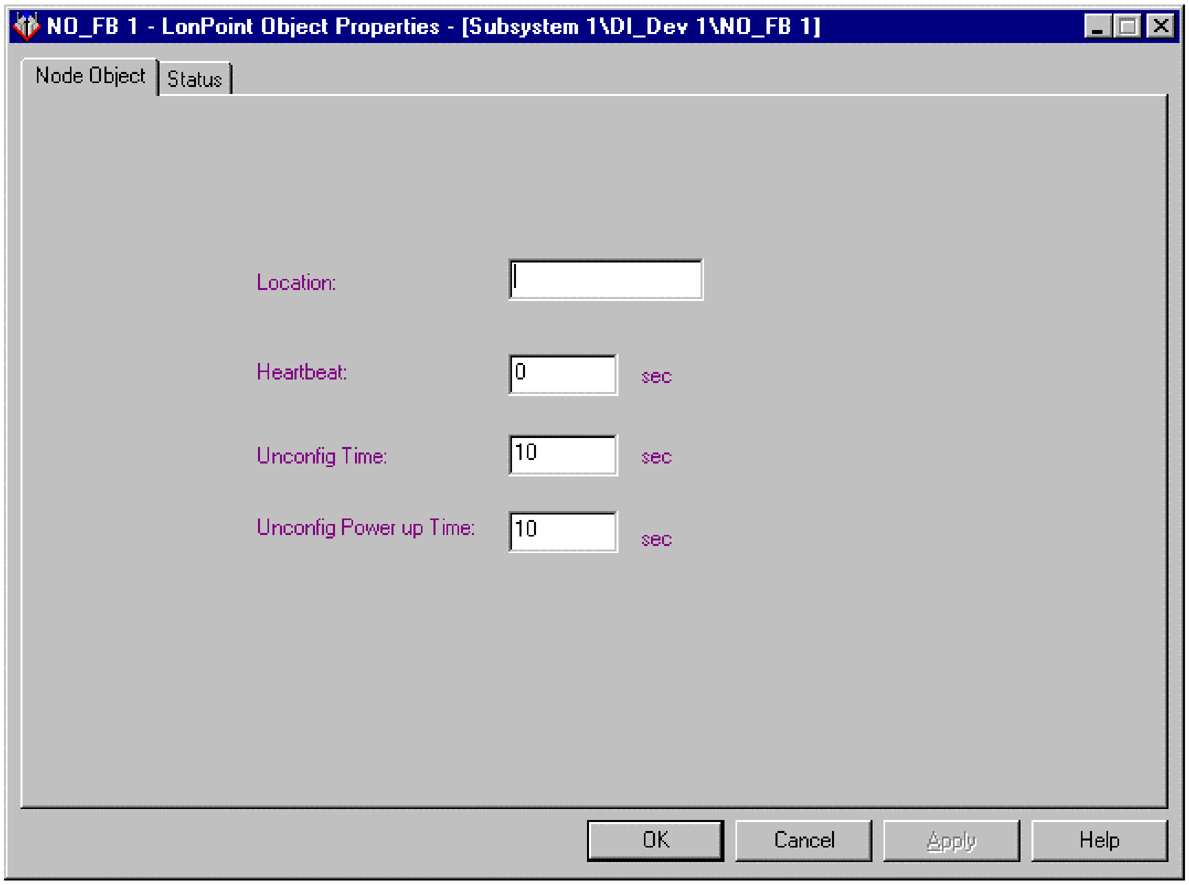
Default name	**Default type**	**Description**
Status	`SNVT_status`	Indicates the status of the device.
Alarm	`SNVT_alarm`	Indicates the alarm status of the device. LonPoint devices do not generate alarms.
Directory	`SNVT_address`	Indicates the address of the file directory of the device. This is used by LONMARK aware network tools for reading and writing configuration properties.

Configuring a Node Object with the LonPoint Plug-in

Right-click a Node Object functional block shape and select *Configure* from the shortcut menu to open the Node Object window of the LonPoint plug-in. This window contains two tabs: `Node Object` and `Status`.

Node Object Window: Node Object

This tab appears as follows:


NO_FB 1 - LonPoint Object Properties - [Subsystem 1\DI_Dev 1\NO_FB 1]
Node Object
Status
OK
Cancel
Apply
Help

Location:

Heartbeat: 0 sec

Unconfig Time: 10 sec

Unconfig Power up Time: 10 sec

This tab allows you to set the following options:

Location	The location string for this device. This property can be used to document the device's location within the plant so it can be easily found. This field may contain up to 30 characters.
Heartbeat	Determines the rate at which the `Status` network variable sends out a heartbeat. Set this value to 0 to disable heartbeat output. The default value is 0 seconds. See *Heartbeats*, in Chapter 1, for more information.
Unconfig Time	The length of time, in seconds, the service button must be pushed to put the device in the unconfigured state. The default is 10 seconds. Set this value to 0 to disable the ability to put the device in the unconfigured state by holding down the service pin. The changes to this configuration property do not take effect until after a device reset.

Unconfig Power up Time	Length of time the service button must be pushed to put the device in the unconfigured state at power-up. The default is 10 seconds. Set this value to 0 to disable the ability to put the device in the unconfigured state by holding down the service pin at power-up. The changes to this configuration property do not take effect until after a device reset.

Status

This tab allows you to view and change the status of a Node Object functional block. See *Status* in Chapter 2 for more information.