



i.LON[®] 100 e3 Plug-in Supplement



078-0315-01A

Echelon, LONWORKS, LonTalk, Neuron, LONMARK, LonScanner and the Echelon logo are trademarks of Echelon Corporation registered in the United States and other countries. LonMaker is a trademark of Echelon Corporation.

Other brand and product names are trademarks or registered trademarks of their respective holders.

Neuron Chips and other OEM Products were not designed for use in equipment or systems which involve danger to human health or safety or a risk of property damage and Echelon assumes no responsibility or liability for use of the Neuron Chips in such applications.

Parts manufactured by vendors other than Echelon and referenced in this document have been described for illustrative purposes only, and may not have been tested by Echelon. It is the responsibility of the customer to determine the suitability of these parts for each application.

ECHELON MAKES AND YOU RECEIVE NO WARRANTIES OR CONDITIONS, EXPRESS, IMPLIED, STATUTORY OR IN ANY COMMUNICATION WITH YOU, AND ECHELON SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Echelon Corporation.

Printed in the United States of America.
Copyright © 2002-2005 Echelon Corporation.

Echelon Corporation
www.echelon.com

Welcome

This document describes how to configure the *i.LON 100* Internet Server's applications with the *i.LON 100* Configuration Plug-In. The *i.LON 100 e3 User's Guide* describes how to configure these applications with the *i.LON 100* Web pages. Echelon recommends that you use the Web pages to configure the *i.LON 100* server whenever possible.

Related Documentation

The documentation for the *i.LON 100* server is provided as online Adobe Acrobat PDF files and Windows Help files. The *i.LON 100* documentation consists of the following manuals:

- *i.LON 100 e3 User's Guide* — Describes how to configure the *i.LON 100* server with the *i.LON 100* Web pages.
- *i.LON 100 e3 Hardware Guide* — Describes how to assemble and wire the *i.LON 100* server hardware.
- *i.LON 100 e3 Programmer's Reference* — Describes how to configure the *i.LON 100* using XML files and SOAP calls. This allows you to create your own applications you can use to configure the *i.LON 100* server.
- *IP-852 Channel User's Guide* — Describes how to configure an IP-852 channel with the Echelon LONWORKS[®]/IP Configuration Server. You will need this information if you plan to use the *i.LON 100* server as an IP-852 router.

The following additional documents may be useful if you are using certain features of the *i.LON 100*. You can download these documents from Echelon's Web site at www.echelon.com.

- *OpenLDV[™] Programmer's Guide, xDriver Supplement* — Describes how the xDriver software can be used by an LNS or OpenLDV application to manage communications with multiple LONWORKS networks over a TCP/IP network. The xDriver software is used to communicate with the *i.LON 100* server when it is functioning as a Remote Network Interface (RNI).
- *LNS[®] Programmer's Guide* — Describes how to write LNS applications that can take advantage of the communication provided by the *i.LON 100* server.
- *LonMaker[®] User's Guide* — Describes how to use the LonMaker tool, which can be used to install the *i.LON 100* server in a LONWORKS network.

Hardware Requirements

Hardware requirements and recommendations for the computer running the *i.LON 100* Configuration Plug-In are listed below:

- Pentium II 600 MHz or faster
- 128-MB RAM minimum

- 70 MB free hard disk space
- CD-ROM drive
- Super VGA (800 × 600) or higher-resolution display with 256 colors
- Mouse or compatible pointing device

Software Requirements

Software requirements for computers running the *i.LON 100* Configuration Plug-In or the *i.LON 100* configuration Web pages are listed below:

- Microsoft® Windows® XP, Windows 2000, or Windows Server 2003. Echelon recommends that you install the latest service pack available from Microsoft for your version of Windows. Screen resolution 1024x768 with large or small fonts; 800x600 with small fonts only.
- LonMaker 3.1, Service Pack 2 or better and/or LNS 3.1, Service Pack 8, Update 1 or better.
- Internet Explorer 6 or higher.
- Terminal emulator, such as Windows HyperTerminal

Table of Contents

Welcome	i
Related Documentation	i
Hardware Requirements	i
Software Requirements	ii
Table of Contents	ii
Introduction	1
Introduction	2
<i>i.LON 100</i> Internet Server Applications	2
Configuring the <i>i.LON 100</i> Applications	5
Configuration Plug-In Overview	6
Getting Started With the Configuration Plug-In	6
Using the Configuration Plug-in With Other LNS Tools	6
<i>i.LON 100</i> Plug-in Limitations	7
Using the <i>i.LON 100</i> Plug-in	7
The <i>i.LON 100</i> Shapes Stencil	8
Adding an <i>i.LON 100</i> Shape to a LonMaker Drawing	8
Adding <i>i.LON 100</i> Functional Blocks to a LonMaker Drawing	9
Starting the <i>i.LON 100</i> Plug-in	9
Organization of the <i>i.LON 100</i> Plug-in	10
Applications Tab	10
Resynchronizing the <i>i.LON 100</i> Plug-in With a Network	11
Setting <i>i.LON 100</i> Plug-in Options	13
Alarming	17
Alarming Overview	18
Alarm Generator	18

Alarm Notifier.....	19
Alarming with the <i>i</i> .LON 100 Configuration Plug-In	20
Providing an Alarm Notification	22
Scenario - Configuring an Alarm System	26
Designing the LonMaker Network Drawing.....	26
Creating a Constant Data Point	27
Configuring the Alarm Generator Functional Block.....	28
Configuring the Alarm Notifier Functional Blocks	30
Simulating an Alarm	34
Detecting Heartbeat Failures.....	35
Latching, Acknowledging, and Clearing Alarms	35
Scheduling	37
Scheduling Overview	38
Creating a Schedule	38
Planning Out Your Schedule	39
Example of Planning Out a Schedule	39
The Scheduler Functional Block.....	39
The Calendar Functional Block.....	40
Using the Same Exceptions for Multiple <i>i</i> .LON 100 Servers.....	41
The Real Time Clock Functional Block Shape	42
Creating a Schedule	42
Adding a Scheduler Functional Block to a Network.....	43
Defining Outputs	44
Defining Presets.....	45
Creating Daily Schedules	47
Creating Exceptions.....	51
Creating a One-time Override Schedule	56
Recovering from a Power Outage	58
Logging Data	59
Data Logger Overview	60
Using a Data Logger with an Alarm Notifier	63
Using Pulse Counter Inputs	65
Pulse Counter Overview	66
Using the Pulse Counter	67
Using Digital Inputs/Digital Outputs	69
Overview	70
Using a Digital Input	71
Using a Digital Output.....	72
Troubleshooting	75
Troubleshooting.....	76

1

Introduction

This chapter provides an introduction to the *i.LON 100* server and its applications, as well as a roadmap to follow when reading this document and learning how to use the *i.LON 100* Configuration Plug-In.

Introduction

The *i.LON 100* Internet Server is a low-cost, high-performance network interface and network device that connects LONWORKS, M-Bus and MODBUS devices to corporate IP networks or the Internet. The *i.LON 100* server features a built-in Web server that allows Web access to all the data managed by the *i.LON 100* server, as well as built-in applications for alarming, scheduling and data logging. Additionally, it includes a Web Binder for bridging multiple LONWORKS domains, and it provides a SOAP/XML Web services interface for use by custom Web pages and for integration with enterprise applications. The *i.LON* server operates on 100 – 240 VAC and can be ordered with an optional built-in 56K V.90 analog modem.

The *i.LON 100* server also includes built-in I/O for reading pulse meters and switching local loads. All data points and built-in I/O is accessible through either the LONWORKS or Web interfaces. The *i.LON 100* server can be used as a Remote Network Interface (RNI), allowing remote access to the entire LONWORKS network using LNS or OpenLDV based tools, including the LonMaker Integration Tool. It can also be used with the LonScanner™ Protocol Analyzer to capture, analyze, characterize, and display ANSI/CEA-709.1 network packets either locally or remotely via the Internet. The *i.LON 100* server operates on 100 – 240 VAC and can be ordered with an optional built-in 56K V.90 analog modem. Models are available for TP/FT-10 channels and PL-20 channels.

i.LON 100 Internet Server Applications

This document describes how to configure the following *i.LON 100* applications with the Configuration Plug-In. For descriptions of the other applications provided with the *i.LON 100* server, see the *i.LON 100 e3 User's Guide*.

- Alarming – The *i.LON 100* server can trigger alarms based on inputs from the devices on the network it is attached to, or from its built-in hardware inputs. In response to an alarm condition, the *i.LON 100* server can be configured to in turn update any network variable in the LONWORKS network, update Web pages, log the conditions to one or more data logs, or send out emails notifying recipients of the alarms and the conditions that triggered them. Alarms can be configured to shut off automatically when certain conditions are met or they can be configured to require manual clearance via a Web page.
- Data Logging – The *i.LON 100* server can log data points from devices on a LONWORKS network. These logs can be downloaded using the Internet File Transfer Protocol (FTP), retrieved using a SOAP/XML Web service, or displayed with the *i.LON 100* Web pages.
- Scheduling – The *i.LON 100* server can be used to create daily and weekly schedules, as well as exception schedules and override schedules. These schedules can drive the inputs to any LONWORKS device.
- Digital Input/Output – The *i.LON 100* server contains 2 built-in digital inputs and 2 built-in digital relay outputs. These can be used to provide LONWORKS functionality to non-LONWORKS devices.

- Pulse Metering – The *i.LON 100* server contains 2 built-in pulse metering inputs. You can configure the *i.LON 100* server to count pulses or to measure the pulse frequency from pulse output devices.

Table 1.1 describes the remaining chapters of this document in more detail.

Table 1.1 Document Roadmap

Chapter	Description
Chapter 2, <i>Configuring the i.LON 100 Applications</i>	This chapter provides an overview of the tools you can use to configure the <i>i.LON 100</i> server: <i>i.LON 100</i> Web pages and the <i>i.LON 100</i> Configuration Plug-In, as well as tips to follow to ensure that the configuration of the <i>i.LON 100</i> server remains synchronized between the two tools. Echelon recommends using the Web pages to configure the <i>i.LON 100</i> server whenever possible.
Chapter 3, <i>Alarming</i>	This chapter describes how to configure the <i>i.LON 100</i> alarming applications with the Plug-In. This includes the Alarm Generator functional block, which you can use to generate alarms based on monitored conditions, and the Alarm Notifier functional block, which you can use to send emails and update data points based on alarm conditions.
Chapter 4, <i>Scheduling</i>	This chapter describes how to configure the <i>i.LON 100</i> scheduling applications with the Plug-In. You can use the scheduling applications to schedule daily, weekly and monthly updates to the data points on your network.
Chapter 5, <i>Logging Data</i>	This chapter describes how to configure the Data Logger application with the Plug-In. You can use Data Loggers to save network data in logs files, and how to view the logs.
Chapter 6, <i>Using Pulse Counter Inputs</i>	This chapter describes how to configure the <i>i.LON 100</i> server's Pulse Counter with the Plug-In. You can use the Pulse Counter to measure electrical energy, volume, rate or flow, or power.
Chapter 7, <i>Using Digital Inputs/Digital Outputs</i>	This chapter describes how to configure the <i>i.LON 100</i> server's two digital inputs and two digital outputs with the Configuration Plug-In.
Appendix A, <i>Troubleshooting</i>	This appendix describes how to diagnose and resolve problems that may occur when you use the Configuration Plug-In.

2

Configuring the *i.LON* 100 Applications

This chapter describes how you can configure the *i.LON* 100 applications with either the *i.LON* 100 Web pages or the *i.LON* 100 Configuration Plug-In.

Configuration Plug-In Overview

You can use the Configuration Plug-In (with the LonMaker tool or with a non-LonMaker LNS application), or the *i.LON 100* Web pages to configure the *i.LON 100* server. These use cases are described below:

- *Using the i.LON 100 with the LonMaker Tool and the Configuration Plug-In.* When using the *i.LON 100* server with the LonMaker tool, the *i.LON 100* Configuration Plug-in will be launched when you configure the *i.LON 100* device shape or any *i.LON 100* functional block shape. The *i.LON 100* configuration property keeps track of the functional blocks that have been dropped in the LonMaker drawing, and will only allow you to configure the applications corresponding to those functional blocks that exist in the drawing.
- *Using the i.LON 100 with a non-LonMaker LNS Tool and the Configuration Plug-In.* When using the *i.LON 100* with an LNS tool other than LonMaker, you can launch the *i.LON 100* Configuration Plug-in by whatever method plug-ins are launched by that tool. The *i.LON 100* Configuration Plug-in will list all *i.LON 100* applications; you are responsible for configuring the correct applications.
- *Using the i.LON 100 Web Pages.* You can also configure the *i.LON 100* server with the *i.LON 100* Web pages. This does not require the use of LonMaker or LNS. The *i.LON 100* Web pages can be used to view and modify (but not create) data points, and to configure any of the *i.LON 100* applications. Note that you cannot add the *i.LON 100* server to an LNS network with the *i.LON 100* Web pages. An LNS tool such as the LonMaker tool must be used to add the *i.LON 100* to an LNS network and create any necessary network variable connections. **Echelon recommends using the Web pages to configure the *i.LON 100* server whenever possible.**

This chapter describes how to access the *i.LON 100* server with the Configuration Plug-In. The remaining chapters of this document describe how to use the Plug-In to configure the various applications of the *i.LON 100* server. The *i.LON 100 e3 User's Guide* describes how to configure these applications with the Web pages.

Getting Started With the Configuration Plug-In

You can use the *i.LON 100* Configuration Plug-in to create and configure data points on the *i.LON 100*, and to configure the *i.LON 100* scheduling, alarming, data logging, digital input and output, pulse counting, type translation, and analog function processing applications. You can start the plug-in from any LNS application that supports LNS plug-ins, such as the LonMaker Integration Tool.

This section provides background information you will need when using the Configuration Plug-in, and describes how to begin configuring your *i.LON 100* server with the Configuration Plug-In.

Using the Configuration Plug-in With Other LNS Tools

The *i.LON 100* Configuration Plug-in is designed to work with any LNS tool that supports plug-ins. When the plug-in is used with the LonMaker tool, it is aware

of what functional block shapes have been added to the LonMaker network and will automatically add them when a resynchronization is performed. See the *Resynchronizing the i.LON 100 Plug-in With a NETWORK* section later in this chapter for more information.

When the *i.LON 100 Configuration Plug-in* used with a non-LonMaker tool, the plug-in is unaware of functional blocks added to the network and these functional blocks must be created in the plug-in manually as described in *Applications* section later in this chapter.

If you use both the LonMaker tool and a non-LonMaker tool to configure your network, the plug-in will be aware of all functional block shapes in the LonMaker drawing, but it will not be aware of functional blocks created in another tool. One of the resynchronization options is **Delete Application Instances Not Defined in LonMaker**. If this option is set and a resynchronization is performed, all *i.LON 100* functional blocks that do not have an associated functional block shape in the LonMaker network drawing will be deleted during the resynchronization, and their configuration will be lost. This option is not set by default.

i.LON 100 Plug-in Limitations

The following *i.LON 100* functional blocks cannot be configured using the *i.LON 100 Configuration Plug-in*:

- *Node Object* — The **Node Object** functional block is used by installation tools such as the LonMaker tool to manage the other functional blocks on the *i.LON 100* server. You typically do not need to configure or bind the **Node Object** functional block. For more information on the uses of the **Node Object** functional block, see the Node Object functional profile, available from the LONMARK Web site (www.lonmark.org).
- *Real Time Clock* — The **Real Time Clock** functional block sends the current time as kept by the *i.LON 100* server. You typically do not need to configure this functional block. You can right-click the functional block and select **Configure** to open the **Setup – Time** Web page, which you can use to adjust the *i.LON 100* server's time settings.
- *Web Server* — The **Web Server** functional block is used to create local data points that are bound to other devices. You can use these data points to monitor and control a LONWORKS network via a Web page. Right-click the **Web Server** functional block and select **Configure** to open the Web page you can use to configure the Web Binder application. This is described in Chapter 10 of this document.

Using the i.LON 100 Plug-in

To begin using the *i.LON 100 Configuration Plug-in*, follow these steps:

1. Use an LNS installation tool such as the LonMaker tool to add an *i.LON 100* server to a LONWORKS network and commission it. If you are using the LonMaker tool, see the *Adding an i.LON 100 Shape to a LonMaker Drawing* section below for more information. If you are using another LNS tool, see the documentation for the tool.
2. Optionally create one or more *i.LON 100* functional blocks with the LNS installation tool. Typically, you add functional blocks corresponding to the *i.LON 100* applications you want to use. If you are using the

LonMaker tool, see the *Adding i.LON 100 Functional Blocks to a LonMaker Drawing* section below for more information. If you are using another LNS tool, see the tool's documentation for more information.

3. Configure the *i.LON 100* server, or the *i.LON 100* functional blocks. For example, to configure an *i.LON 100* functional block using the LonMaker tool, right-click the functional block in the LonMaker drawing and then click **Configure** on the shortcut menu. If you are using another LNS tool, see the documentation for the tool. See the *Starting the i.LON 100 Plug-in* section for more information.

The *i.LON 100* Shapes Stencil

The ***i.LON 100 Shapes*** Visio stencil is installed with the *i.LON 100* software. This stencil contains LonMaker shapes for the *i.LON 100* device and the *i.LON 100* functional blocks. This stencil should open automatically when a LonMaker network is opened. If this stencil does not open, follow these steps to open it:

1. In Visio, select the **File** menu, hover your mouse cursor over **Stencils**, and then select **Open Stencil** from the drop-down menu. The **Open Stencil** dialog opens.
2. Browse to LonWorks\LonMaker\Visio.
3. Select the iLON100.vss stencil.

If you are using the LonMaker tool, the new version 3.0 stencil will overwrite the old stencil. You cannot use the version 3.0 *i.LON 100* shape to create and commission a version 1.0 or 1.1 *i.LON 100*; attempting to do so will result in an error.

Adding an *i.LON 100* Shape to a LonMaker Drawing

You can add an *i.LON 100* server to a LonMaker drawing. This allows you to place the *i.LON 100* functional blocks in the drawing and connect the input and output network variables on those functional blocks to other devices in the LONWORKS network. This in turn allows you to use the data logging, scheduling, and alarming applications on the *i.LON 100* server to monitor and control the network. See the *LonMaker User's Guide* for more information about the LonMaker tool.

To add an *i.LON 100* shape to a LonMaker drawing, follow these steps:

1. Start the LonMaker tool.
2. Open or create a LonMaker network.
3. If this is the first time that you are opening the LonMaker network since installing the *i.LON 100* software, register the **Echelon *i.LON 100* Configuration Plug-in** and the **Echelon *i.LON 100* Web Server Plug-in** with the **Plug-in Registration** window. See the *LonMaker User's Guide* for more information on this step.
4. Drag the ***i.LON 100 FTT*** shape (for the free topology model) or ***i.LON 100 PL*** shape (for the power line model) from the *i.LON 100* Shapes stencil to the LonMaker drawing. The New Device Wizard appears.
5. Step through the **New Device Wizard**. If your *i.LON 100* server is connected to the network, you can commission it just like any other LONWORKS device. If you create any new network variables on any of the

i.LON 100 functional blocks, you must commission the *i.LON 100* device to add the new network variables to the *i.LON 100* hardware.

NOTE: If you have trouble commissioning the device on a power line channel due to inability to communicate with the device, you should increase the LonMaker **Transmit Timer** to at least 512 ms; to do this, open the **LonMaker** menu, click **Network Properties**, select the **Timing** tab, and then set this property.

Adding *i.LON 100* Functional Blocks to a LonMaker Drawing

Once you have added the *i.LON 100* shape to the LonMaker drawing as described above, you can add *i.LON 100* functional blocks. The functional blocks supported by the *i.LON 100* server are contained in the *i.LON 100* Shapes stencil. Drag an *i.LON 100* functional block shape to the LonMaker drawing in order to add it to the network design. See the *LonMaker User's Guide* for more information on adding functional blocks and creating network variable shapes.

Starting the *i.LON 100* Plug-in

To start the *i.LON 100* configuration plug-in, follow these steps:

1. Select an *i.LON 100* device or functional block as required by your LNS installation tool and then start the plug-in. For example, if you are using the LonMaker tool, select an *i.LON 100* device or functional block shape and then click **Configure** on the shortcut menu. The Specify *i.LON 100* Configuration Data Source dialog opens.



Figure 3.2 Specify *i.LON 100* Configuration Data Source Dialog

2. Enter the IP address of the *i.LON 100* server in the TCP/IP Address field. If you have configured the *i.LON 100* server to use a port other than 80, you must specify the port with the UP address. For example, to specify port 8080 with the IP address shown in Figure 3.2, you would enter

172.25.130.18:8080.

Select the **Use SOAP Authentication** check box if you want the SOAP messages sent to and from your *i.LON* 100 server to be authenticated. The Username and Password both default to *ilon*.

3. Click **OK**.

NOTE: The *i.LON* 100 server must be connected to both the TCP/IP and the LONWORKS network for you to configure it with the *i.LON* 100 Configuration Plug-in. The *i.LON* 100 server should always be in sync with the LONWORKS application that opens it (*i.e.* in the LonMaker tool, you should be connected and OnNet).

The *i.LON* 100 server must be commissioned and online and the LonMaker tool must be attached to the network with the *i.LON* 100 in order to use the *i.LON* 100 Configuration Plug-in.

Organization of the i.LON 100 Plug-in

The *i.LON* 100 Configuration Plug-in is organized into two main tabs, the **Data Points** tab and the **Applications** tab. You can use the **Data Points** tab to create and configure local, external, and constant data points. See Chapter 4 of the *i.LON 100 e3 User's Guide* for more information on data points.

You can use the **Applications** tab to create and configure *i.LON* 100 functional blocks, as described in the following section.

Applications Tab

The **Applications** tab of the *i.LON* 100 Configuration Plug-in contains the **Functional Blocks** list, as shown in Figure 3.3.

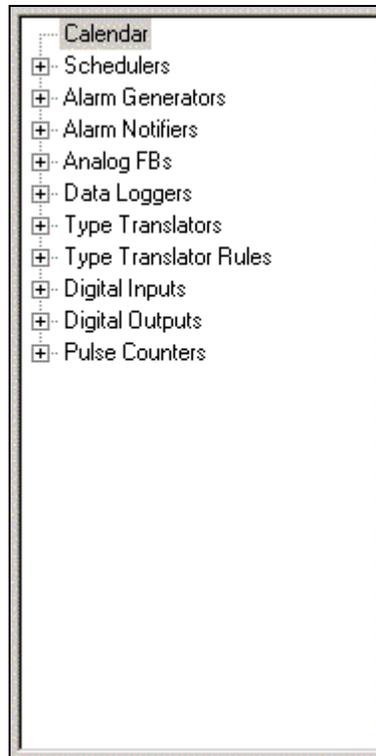


Figure 3.3 Applications Pane

Click a '+' button to show a list of all instances of a specified application. Select an application to configure it, as described in the remainder of this document (note that the Type Translator and Analog Function Block applications are described in Chapter 11 of the *i.LON 100 e3 User's Guide*).

The applications pane will display all functional blocks that have been added to the *i.LON 100* server. In addition, if the *i.LON 100* Configuration Plug-in in was started via a request to configure a functional block, this functional block will be shown in the applications pane. If you are using the LonMaker tool, you can perform a resynchronization to have all functional blocks in the LonMaker drawing added to this pane. See the *Resynchronizing the i.LON 100 Plug-in With a Network* later in this chapter for more information.

You can add a functional block to the list by right clicking the application and then clicking **Add** on the shortcut menu. Copy a functional block (and its configuration) by right-clicking the instance and selecting **Duplicate** from the short-cut menu. Remove a functional block by right clicking the instance and selecting **Delete** from the short-cut menu.

Resynchronizing the i.LON 100 Plug-in With a Network

When you add or remove an *i.LON 100* functional block using an LNS installation tool, the changes will not be recognized by the *i.LON 100* server until it is resynchronized. Resynchronization has the following effects:

- Adds and removes functional blocks to make the *i.LON 100* **Functional Blocks** list consistent with the functional block shapes in the LonMaker drawing. This step will not be performed if the network was configured with an LNS tool other than the LonMaker tool.

- Updates the type of all local and external data points to be consistent with the LONWORKS network. It will not update data points if only the format is different. If the type of a data point changes, all preset information, including the default value, will be lost.
- Adds data point references to the functional blocks. For example, if you add input network variables to the **Data Logger** functional block, the corresponding data points will be added to the **Data Logger** configuration page of the *i.LON 100* Plug-in. This step will not be performed if the network was configured with an LNS tool other than the LonMaker tool.
- If you removed any network variables from the *i.LON 100* server, they will optionally be reported. You can remove the associated data points from the *i.LON 100* using the **Delete** button on the **Data Points: Properties** tab.
- If any functional blocks refer to data points that are no longer defined (*i.e.* the corresponding network variable has been removed), they will be reported. To have these data points automatically removed, set **Remove Invalid Data Points From the *i.LON 100*** as described below.

To resynchronize the *i.LON 100* server with a LonMaker drawing, follow these steps:

1. Click the **Tools** menu and then select **Resync**. Note that in version 1.0 of the *i.LON* software, the resynchronization feature was accessible through a button on the lower part of the dialog.
2. Optionally click **Options** to open the Resync Options dialog.

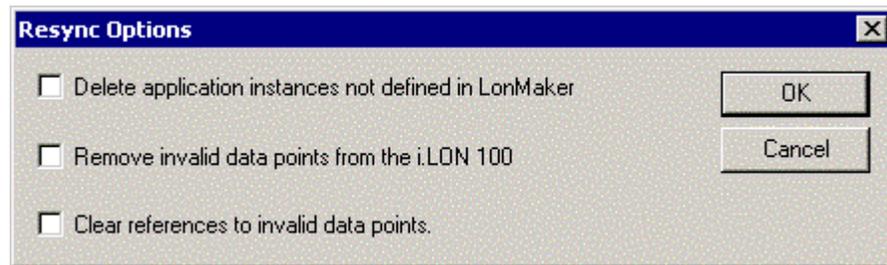


Figure 3.4 Resync Options Dialog

3. Set the options on the Resync Options dialog, and click **OK** to continue. Table 3.1 describes these options.

Table 3.1 Resync Options Dialog Properties

Options	Description
Delete Application Instances Not Defined in LonMaker	If you are using the <i>i.LON 100</i> Configuration Plug-in with the LonMaker tool, select this check box to have the resynchronization delete all functional blocks that do not have corresponding functional block shapes in the LonMaker network. If there are <i>i.LON 100</i> functional blocks that do not have corresponding LonMaker functional block shapes, resynchronizing with this option set will cause those functional blocks to be removed, along with all configuration. This check box is selected by default.
Remove Invalid Data Points From the <i>i.LON 100</i>	Select this check box to have any invalid data points (<i>i.e.</i> data points that can no longer access their associated network variable) removed as part of the resynchronization. This check box is not selected by default.
Clear References to Invalid Data Points	Select this check box to have any references to invalid data points removed as part of the resynchronization. This check box is not selected by default.

4. Click **Go** to begin the resynchronization. The dialog lists the changes that will be made to the *i.LON 100* server's configuration data. Review the changes.
5. Click **Apply** to accept the changes and update the *i.LON 100* server and the *i.LON 100* Configuration Plug-in.

Click **Continue** to accept the changes and update the Configuration Plug-in, but not the *i.LON 100* server. The *i.LON 100* server will be updated when you click the **OK** or **Apply** button in the *i.LON 100* Configuration Plug-in.

Click **Cancel** to reject the changes.

NOTE: If you subsequently delete the network using a tool other than the LonMaker tool after performing a resynchronization with the *i.LON 100* Configuration Plug-in, you must manually delete the network folder and all subfolders. The resynchronization process creates a folder (**\iLonXml**) below the network folder and a resynchronization log that are not removed when the network is deleted (this folder is also created by the backup and upgrade procedures described later in this chapter). The LonMaker tool will ask you whether you want to remove these folders as part of the network deletion process. If you do not remove these files, you will not be able to create a new network of the same name as the deleted one.

Setting *i.LON 100* Plug-in Options

To set *i.LON 100* Configuration Plug-in options, open the **Tools** menu and select **Options**. The *i.LON 100* Configuration Utility Options dialog opens.

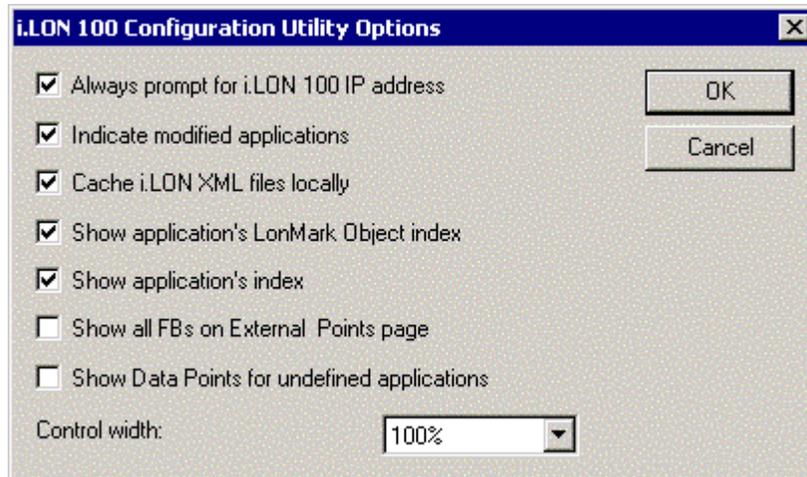


Figure 3.5 *i.LON 100 Configuration Utility Options Dialog*

Set the options on the dialog, and click **OK** to save your changes. Table 3.2 describes the options.

Table 3.2 *i.LON 100 Configuration Utility Options Dialog Properties*

Option	Description
Always Prompt for <i>i.LON 100</i> IP Address	Select this check box to always have the Specify <i>i.LON 100</i> Configuration Data Source dialog open when you start the <i>i.LON 100</i> Configuration Plug-in. If the <i>i.LON 100</i> is not commissioned or you are not OnNet, this dialog will appear, regardless of how this option is set.
Indicate Modified Applications	Select this check box to have an asterisk appear next to applications in the Applications list when they have been modified but the changes have not been applied (<i>i.e.</i> you have not clicked the Apply or OK buttons).
Cache <i>i.LON 100</i> XML Files Locally	Select this check box to have <i>i.LON 100</i> XML files cached locally. These files will be stored in the <i>iLON100Xml</i> folder in the LonWorks database folder for the LonWorks database that contains the <i>i.LON 100</i> server.
Show Application's LonMark Index	Select this check box to have the functional block numbers appear for each functional block in the Applications list.
Show Application's Index	Select this check box to have the application indices for each functional block appear in the Applications list.

Option	Description
Show All FBs on External Points Page	Select this check box to see all functional blocks on all devices when creating an external data point when starting the <i>i.LON 100 Configuration Plug-in</i> from the LonMaker tool. If this option is cleared, only the functional blocks that have been added to the LonMaker drawing will be shown. If the <i>i.LON 100 Configuration Plug-in</i> is started from another LNS application, you will always see all functional blocks on all devices when creating an external data point. This option is cleared by default.
Show Data Points For Undefined Applications	Select this check box to have the Data Points:Properties tab show all NVL data points, regardless of whether the associated functional block has been created on the <i>i.LON 100</i> .
Control Width	Resizes the <i>i.LON 100 Configuration Plug-in</i> window. Any change to this option will not take effect until the <i>i.LON 100 Configuration Plug-in</i> is closed and restarted.

3

Alarming

This chapter describes how to configure the *i.LON 100* alarming applications with the Plug-In. This includes the Alarm Generator functional block, which you can use to generate alarms based on monitored conditions, and the Alarm Notifier functional block, which you can use to send emails and update data points based on alarm conditions.

Alarming Overview

The *i*.LON 100 server contains two applications that control alarming—the **Alarm Generator** and the **Alarm Notifier**. The *i*.LON 100 server can use the Alarm Generators to monitor up to 40 data points, and trigger alarms by setting the status of an input data point when specified alarm conditions occur. The Alarm Generator can optionally update a SNVT_alarm and/or SNVT_alarm_2 data point for LONMARK integration with other alarming devices.

The **Alarm Notifier** functional blocks monitor the status of data points, as well as conditions sent via SNVT_alarm and SNVT_alarm_2 data points. Alarm Notifiers can be configured to respond to an alarm by updating one or more data points, sending email messages, and recording the alarm into a log file. A single Alarm Notifier can be configured to monitor multiple inputs.

Alarm Generator

You can create up to 40 Alarm Generators per *i*.LON 100 server. An Alarm Generator generates alarms based on the values of any of the *i*.LON 100 data points. When you create an Alarm Generator, you will specify an *input data point*, and either a *compare data point* (if you want to monitor the value of the input data point against the value of another data point) or a *compare value* (if you want to monitor the value of the input data point against a constant value).

The Alarm Generator compares the values whenever the input data point or the compare data point is updated. You will select the function the Alarm Generator will use to make the comparison. If the result of the comparison is true, an alarm will be generated, and the status of the input data point will be updated to an alarm condition.

For example, you could select Greater Than as the comparison function. In this case, the Alarm Generator would generate an alarm when either data point is updated, and the value of the input data point is greater than the compare value. The Alarm Generator includes the following binary comparison functions: Less Than, Less Than or Equal, Greater Than or Equal, Equal, and Not Equal.

The Alarm Generator also includes an analog comparison function. When you select this comparison function, you will specify four offset limits for the Alarm Generator. The four offset limits allow you to generate alarms based on how much the value of the input data point exceeds, or is exceeded by, the value of the compare data point. The Alarm Generator generates an alarm when either data point is updated and the difference between their values exceeds any of the offset limits.

You will define a hysteresis level for each alarm-offset limit when you use the analog comparison function. After an alarm has been generated based on an offset limit, the value of the input data point must return to the hysteresis level defined for that offset limit before the alarm clears, and before another alarm can be generated based on that limit. As a result, the Alarm Generator will not generate an additional alarm each time the input data point is updated after it reaches an alarm condition, but before it has returned to a normal condition.

All of the comparison functions have additional features that will allow you to throttle alarm generation. You can specify an interval that must elapse between alarm generations for a single data point. You can also define an interval that must elapse after an alarm has returned to normal status before that alarm will

be cleared. These features prevent the Alarm Generator from triggering multiple alarms each time the input data point reaches an alarm condition.

You can optionally select up to two *alarm data points* for each Alarm Generator: one of type SNVT_alarm and one of type SNVT_alarm_2. The status of these data points will be updated to an alarm condition each time the Alarm Generator state changes (*i.e.* passive to active *or* active to passive).

Alarm Notifier

You can also create up to 40 Alarm Notifiers per *i.LON* 100 server. An Alarm Notifier logs alarm conditions and generates email messages and data point updates each time an alarm condition occurs. You can use an Alarm Notifier to provide notification of alarms generated by any device that produces a SNVT_alarm or SNVT_alarm_2 output, including the *i.LON* 100 server. For example, you can use an Alarm Notifier to provide alarm notification of alarms produced by the Alarm Generators on your *i.LON* 100 server.

You will specify a group of input data points for each Alarm Notifier. The Alarm Notifier will read the status of these data points each time they are updated to determine if the data point's status has been changed to a value that you have defined as an alarm condition. If it has been updated to such a value, an alarm notification will be made.

You can specify one or more output data points for each Alarm Notifier. These data points will be updated each time an alarm notification occurs. You can also specify an email profile for each Alarm Notifier. An email message will be sent to the addresses specified for that email profile each time an alarm notification occurs. You can specify the message text, subject heading, and attachment to be included with each email notification. Email profiles allow you to notify different people when different alarms occur. This is useful if different groups of people need to receive notifications about the various alarm conditions that can occur on your network.

Each Alarm Notifier generates a log file. It will add an entry to this log file each time it causes an alarm notification. You can find these log files in the `/root/AlarmLog` directory of the *i.LON* 100 server. These files are named `histlogX`, where X represents the index number assigned to the Alarm Notifier when it was created. An Alarm Notifier will not generate a log file until it has generated an alarm notification.

The *i.LON* 100 server does not limit how much alarm data can be logged. However, you should maintain at least 1024KB of free disk space on the *i.LON* server. You can view the amount of free disk space using the **Setup - System Info** Web page.

In addition, the Alarm Notifier generates a summary log that summarizes the log entries made by all Alarm Notifiers that were classified as active alarms. This file is called `sumlog0`, and can also be found in the `/root/AlarmLog` directory of your *i.LON* 100 server.

If you are using the *e3* software, Echelon recommends that you use the Web pages to configure your Alarm Notifiers, as the Alarm Notifier application instance you are configuring may not be compatible with the Configuration-Plug-In.

Alarming with the *i.LON 100* Configuration Plug-In

To configure an Alarm Generator to generate alarms with the *i.LON 100* Configuration Plug-in, follow these steps:

1. Open an existing LonMaker network or create a new LonMaker network as described in the *LonMaker User's Guide*.
2. Use an existing *i.LON 100* device in the LonMaker network, or create and commission a new *i.LON 100* device.
3. Define a data point to be monitored as described in Chapter 4 of the *i.LON 100 e3 User's Guide*. This is the *input data point*.
4. Define a data point to use as a comparison value. This is the *compare data point*. For example, you may define a constant data point to compare the input data point against.
5. Associate the **Alarm Generator** functional block shape with any of the available **Alarm Generator** functional blocks on the *i.LON 100* server, and then click **Next**.
6. Enter a name for the **Alarm Generator** functional block, and then click **Finish**.
7. Right-click the **Alarm Generator** functional block, and then select **Configure** from the shortcut menu. Enter the IP address or fully qualified host name of the *i.LON 100* server when prompted. The dialog shown in Figure 4.1 opens.

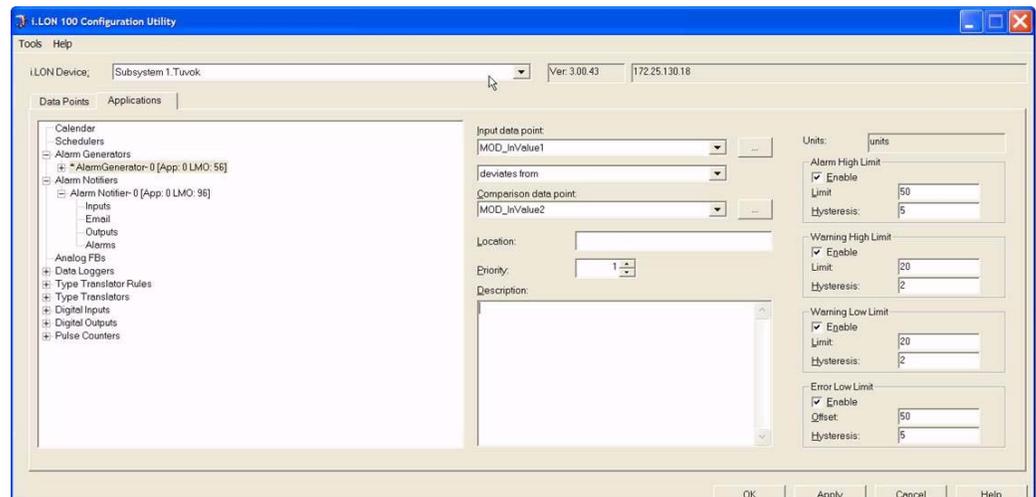


Figure 4.1 *i.LON 100* Configuration Utility – Alarm Generator

8. Select the input and comparison data points from the **Input Data Point** and **Comparison Data Point** lists.
9. Select a comparison function from the **Comparison** list between the input and comparison data points. The available comparison functions include the following:
 - Input value deviates from comparison value.
 - Input value is equal to comparison value.
 - Input value is greater than comparison value.
 - Input value is greater than or equal to comparison value.

- Input value is less than comparison value.
 - Input value is less than or equal to comparison value.
 - Input value is not equal to comparison value.
10. If you selected the **Deviates From** comparison function, set the **Alarm High Limit, Warning High Limit, Warning Low Limit, and Error Low Limit** fields on the right side of the dialog. These fields define the conditions the Alarm Generator will use to determine when an alarm should be generated.
- Select the applicable check box to enable each field. Then, fill in the **Limit, Offset** and **Hysteresis** properties for each field you enabled to define the alarm conditions for the data point. See Chapter 5 of the *i.LON 100 e3 User's Guide* for descriptions of these fields.
11. Set **Priority** to a value from 0 to 11, with 0 being the highest priority and 11 being the lowest priority. This value will be sent in the priority field of the SNVT_alarm or SNVT_alarm_2 output. Note that if you previously set the **Priority** field to **None** with the *i.LON 100* Web pages, the property will become incompatible with the Configuration Plug-In, meaning that the Configuration Plug-In will display an invalid value for the Priority field and you may not be able to change the field's value.
12. Enter a description of the criteria in **Description**. This allows other network integrators to easily understand the purpose of this **Alarm Generator** functional block.
13. Select **Output** from below the Alarm Generator from **Applications** list. The dialog shown in Figure 4.2 opens.

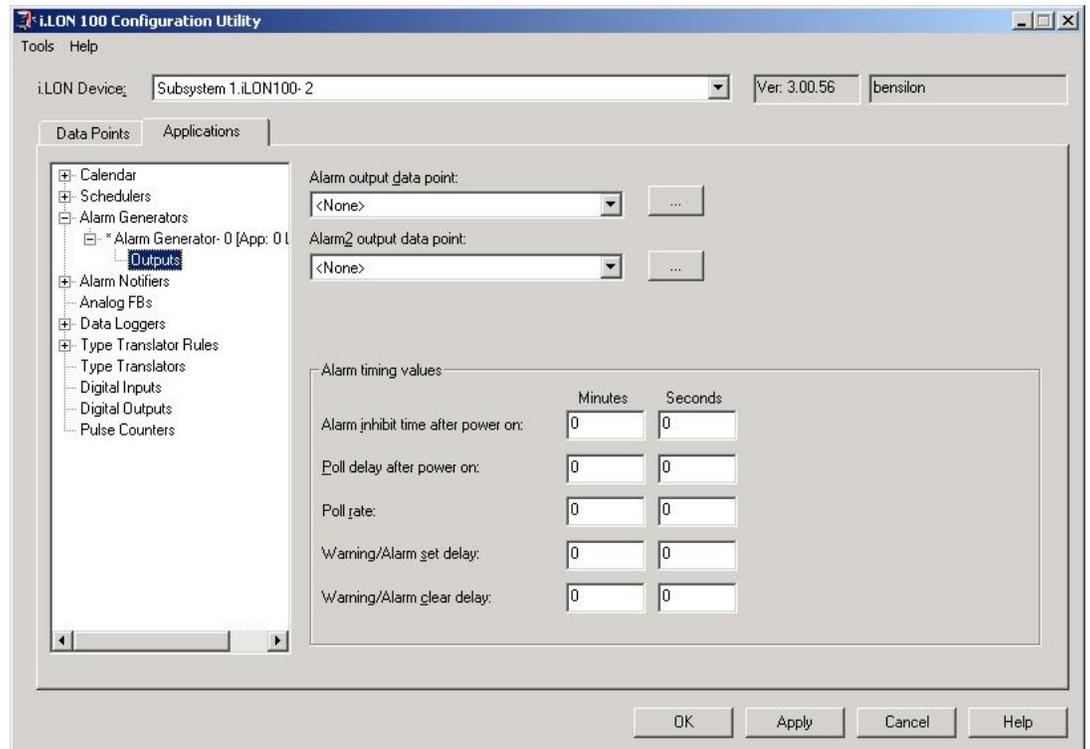


Figure 4.2 Alarm Generator Outputs Dialog

14. Select one or two output data points from the **Alarm Output Data Point** and **Alarm2 Output Data Point** drop-down lists. Use **Alarm Output Data Point** to select a SNVT_alarm output, and use **Alarm2 Output Data Point** to select a SNVT_alarm_2 output.
15. You can optionally specify the **Alarm Timing Values**, as well. Set **Poll Rate** to a positive value if you have specified an external data point for either the input value or the comparison value. Click **Help** for a description of these values.
16. Click **OK** to configure the *i.LON 100* device and close the plug-in, or click **Apply** to configure the *i.LON 100* device without closing the plug-in.

Providing an Alarm Notification

You can configure an **Alarm Notifier** functional block to provide a network or email notification of an alarm condition. The alarm condition may be generated by an **Alarm Generator** or by another LONWORKS device that produces SNVT_alarm or SNVT_alarm_2 alarm outputs, or it may be generated in response to any data point going offline.

If you are using the *e3* software, Echelon strongly recommends that you use the Web pages to configure your Alarm Notifiers, as the Alarm Notifier application instance you are configuring may not be compatible with the Configuration-Plug-In.

To configure an Alarm Notifier to provide alarm notifications with the Configuration Plug-In, follow these steps:

1. Use an LNS installation tool such as the LonMaker tool to open an existing network or create a new network as described in your installation tool documentation.
2. Use an existing *i.LON 100* server in your network, or create and commission a new *i.LON 100* device.
3. If you plan to provide email notifications with the Alarm Notifier, configure the *i.LON 100* device to send email as described in Chapter 2 of the *i.LON 100 e3 User's Guide*.
4. Add an **Alarm Notifier** functional block to your network. For example, using the LonMaker tool drag an **Alarm Notifier** functional block shape from the *i.LON 100* Shapes stencil to the drawing. The New Functional Block wizard appears. The next 3 steps assume you are using the LonMaker tool.

NOTE: If you are using the *e3* software and you attempt to configure an Alarm Notifier that has previously been configured with the Web pages, a dialog will appear to warn you that the Alarm Notifier may be incompatible with the Configuration Plug-In. Echelon recommends that you use the Web pages to configure your Alarm Notifiers whenever possible.

5. Associate the **Alarm Notifier** functional block shape with any of the available **Alarm Notifier** functional blocks on the *i.LON 100* device, and then click **Next**.
6. Enter a name for the **Alarm Notifier** functional block, and then click **Finish**.

- Right-click the **Alarm Notifier** functional block and then select **Configure** from the shortcut menu. Enter the IP address or fully qualified host name of the *i.LON 100* server when prompted. The dialog shown in Figure 4.3 appears.

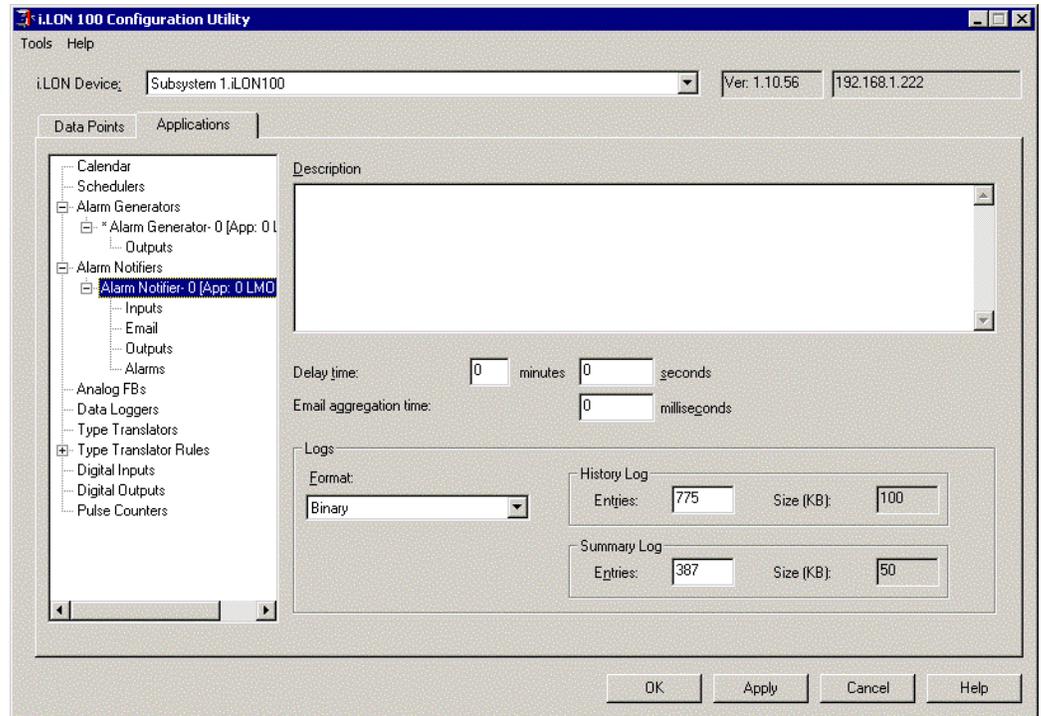


Figure 4.3 Alarm Notifier Configuration Dialog

- Enter a description of the Alarm Notifier in **Description**. This allows other network integrators to easily understand the purpose of this **Alarm Notifier** functional block.
- Optionally set global options for this Alarm Notifier. Click **Help** for a description of the available options.
- Click **Inputs** under the selected **Alarm Notifier** in the Applications list. The dialog shown in Figure 4.4 opens.

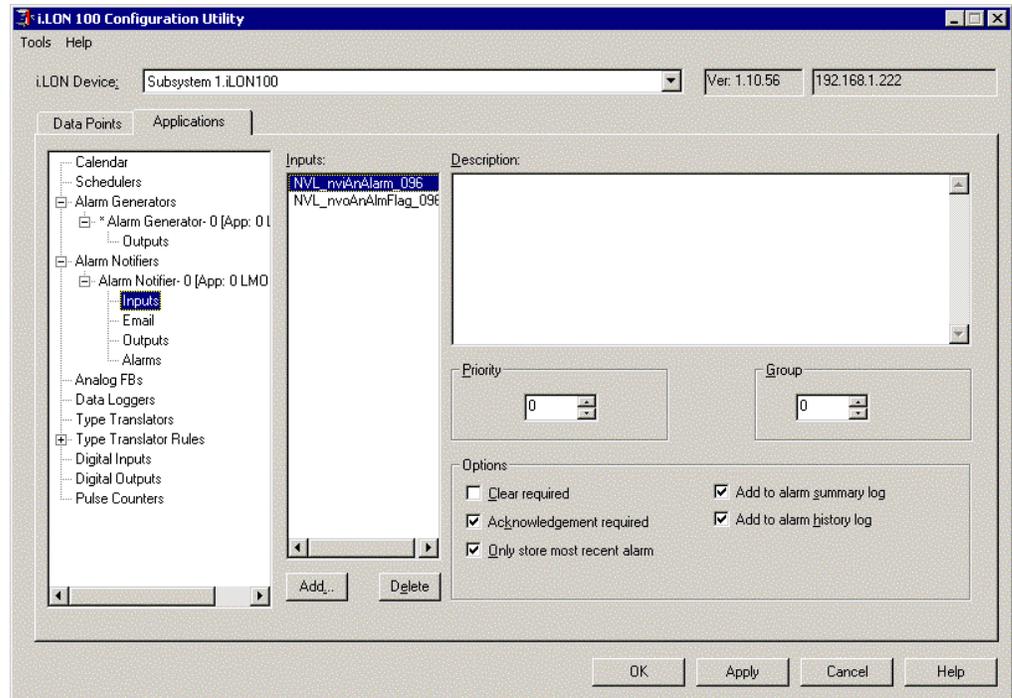


Figure 4.4 Alarm Notifier Inputs Dialog

11. Select the data points to be monitored by the Alarm Notifier in the **Inputs** list. The Alarm Notifier will generate notifications for these data points as defined on the **Email** and **Outputs** tabs. You can select any type of data point. If a data point is a SNVT_alarm or SNVT_alarm_2 type, the Alarm Notifier will generate alarm notifications based on the value of the data point. If the data point is of any other type, the Alarm Notifier will generate an alarm notification if the status of the data point changes to OFFLINE. This typically happens when the time between two data point inputs is greater than the **Max Recv Time** set in the Data Points Properties tab.
12. Set **Priority** to the priority of the alarm generated by this Alarm Notifier. This value can be from 0 (highest priority) to 255 (lowest priority). If alarm conditions are received on more than one input, the alarm with the highest priority will take precedence.
13. Set **Group** to an alarm group number. This value can be from 1 to 127. A value of 0 indicates the alarm has no group. You can use groups to classify alarms into installation-specific categories. You can assign any meaning to each of the group numbers. The group numbers have no effect on the function of the Alarm Notifier, but you can include the group number in email messages sent by the Alarm Notifier.
14. Optionally set input options for this Alarm Notifier. Click **Help** for a description of the available options.
15. To send email notifications, click **Email** under the selected **Alarm Notifier** in the Applications list. Select SNVT_switch data points to be used to trigger email notifications, and define an email message to be sent for each of these data points. When an alarm changes state (*i.e.* when it becomes active or when it becomes passive), the Alarm Notifier sends email messages that are associated with all data points in the

Enable Data Points list that are set to On. You can define the text for each email message, including substitution strings that specify details about the alarm such as alarm value, alarm limit, and alarm time. Click **Help** for more details on how to define email notifications.

16. To provide alarm notifications via data point outputs from the Alarm Notifier, select **Outputs** under the selected **Alarm Notifier** in the Applications list. Select SNVT_switch data points to be used to enable outputs from the Alarm Notifier. You can define output data points to be set for each enable data point with an On value. Each output data point has an *active value* and a *passive value*. When an alarm condition is received, the enabled output data points are set to the active value. When an alarm is cleared, the enabled output data points are set to the passive value. Click **Help** for more details on how to define output notifications.
17. To enable or disable specific alarm conditions for an Alarm Notifier, select **Alarms** under the selected **Alarm Notifier** in the Application list. This opens the dialog shown in Figure 4.5.

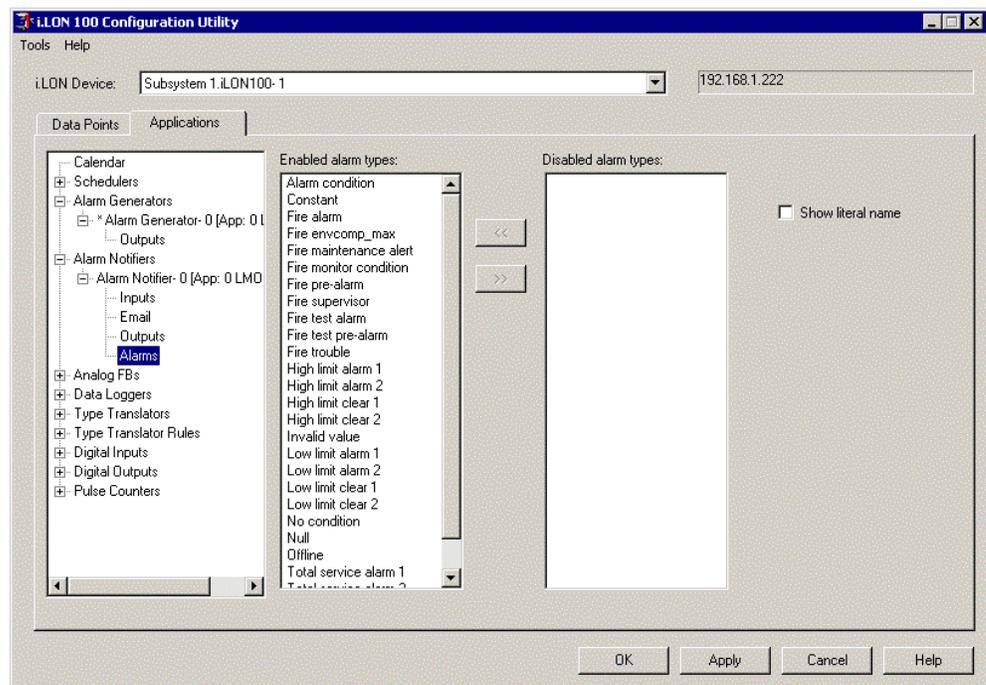


Figure 4.5 Alarms Dialog

Normally, an alarm is triggered whenever an alarm condition other than **No Condition** or **Null** is received (alarm conditions are received on **SNVT_alarm** or **SNVT_alarm_2** network variables—these are sent by the Alarm Generator or any other functional block with a **SNVT_alarm** or **SNVT_alarm_2** output).

To disable one or more alarm types, select them from the **Enabled Alarm Types** list and click the “>>” button, to move them to the **Disabled Alarm Types** list.

To see the enumerated names for alarm types used by **SNVT_alarm** and **SNVT_alarm_2**, set the **Show Literal Name** check box.

18. Click **OK** to configure the *i.LON 100* device and close the plug-in, or click **Apply** to configure the *i.LON 100* device without closing the plug-in.

Scenario - Configuring an Alarm System

The following example demonstrates how to design and configure a simple alarming application using the *i.LON 100* Configuration Plug-in. The example is specific to a particular scenario, but the concepts can be used for any alarming application. In this example, a water-piping system for a building in New York City uses an **Alarm Generator** functional block connected to a temperature sensor functional block on another device.

The Alarm Generator compares the data point value from the temperature sensor with a constant data point value set to a temperature near freezing. This constant value represents the low-limit 2 value for the temperature. When the temperature falls below four degrees above the *low-limit 2 value*, the generator updates a SNVT_alarm data point. This is the *low-limit 1* value. An **Alarm Notifier** functional block monitors this data point and when an alarm condition is set, it automatically turns on the heating mechanism to keep the pipes from freezing and sends an email to the building staff informing them of the alarm and the response. The alarm is logged in the *i.LON 100*'s internal alarm log. When the temperature returns to a safe level, the alarm is cancelled automatically.

If the temperature continues to drop below the low-limit 2 value, the Alarm Generator functional block triggers a different, more serious alarm condition. A separate Alarm Notifier functional block then sends a more urgent email message reports this second alarm condition. This Alarm Notifier is configured so the alarm is not automatically cleared. It must be manually cleared using a Web page hosted by the *i.LON 100*.

Designing the LonMaker Network Drawing

To use the LonMaker tool to create the example network described above, follow these steps:

1. Open an existing LonMaker network or create a new LonMaker network as described in the *LonMaker User's Guide*.
2. Use an existing *i.LON 100* device in the LonMaker network, or create and commission a new *i.LON 100* device.
3. Configure the *i.LON 100* device to send email as described in Chapter 2 of the *i.LON 100 e3 User's Guide*.
4. Drag an **AI-10** LonPoint device shape from the LonPoint Shapes stencil to the drawing. This LonPoint device will be physically attached to a temperature sensor. See the *LonPoint Hardware Guide* and *LonPoint User's Guide* for more information.
5. Drag an **Analog Input** functional block shape from the LonPoints Shapes stencil to the drawing. Associate it with one of the Analog Sensor functional blocks on the AI-10 device. Name this shape **Temperature Input**.
6. Drag a **DO-10** LonPoint shape from the LonPoint Shapes stencil to the drawing. This LonPoint will be physically attached to a heater control relay near the water pipes.

7. Drag a **Digital Output** shape from the LonPoint Shapes stencil to the drawing. Associate it with one of the **Digital Output** functional blocks on the DO-10 device. Name this shape **Heater**.
8. Drag an **Alarm Generator** functional block shape from the *i.LON 100* Shapes stencil to the drawing. Associate it with any of the available **Alarm Generator** functional blocks on the *i.LON 100* server. Name this shape **Alarm Generator**.
9. Drag two **Alarm Notifier** functional block shapes from the *i.LON 100* Shapes stencil. Associate them with two **Alarm Notifier** functional blocks on the *i.LON 100* server. Name these shapes **Alarm Notifier – Warning** and **Alarm Notifier – Emergency**.
10. Make the following network variable connections:
 - Connect the **Analog** output network variable on the **Temperature Input** functional block to the **nviAgInput** input network variable on the **Alarm Generator** functional block.
 - Connect the **nvoAgAlarm** output network variable on the **Alarm Generator** functional block to the **nviAnAlarm** input network variables on both the **Alarm Notifier – Warning** and **Alarm Notifier – Emergency** functional blocks.
 - Connect the **nvoAnAlarmFlag** output network variable on the **Alarm Notifier – Warning** functional block to the **Digital** input network variable on the **Heater** functional block.

When you are done, your LonMaker drawing should look something like Figure 4.6.

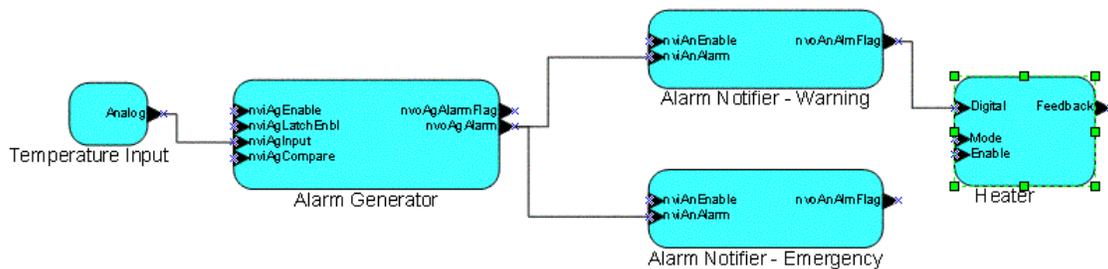


Figure 4.6 Sample LonMaker Drawing

Creating a Constant Data Point

In the scenario described above, the goal is to check the value of the **nviAgInput** network variable to determine whether or not the pipes are freezing. In order to do this, create a constant data point that contains the value for freezing as a value in degrees Celsius. To do this, follow these steps:

1. Right-click the *i.LON 100* device shape and select **Configure** from the shortcut menu. Enter the IP address or fully qualified host name of the *i.LON 100* server when prompted. The *i.LON 100* Configuration Plug-in opens.
2. Select the **Data Points** tab and the **Constants** sub-tab.
3. Click **Add** to create a new constant data point. The **Set Data Point Format** dialog opens.

- Choose **STANDARD.FMT** from the **Type Files** list, select **SNVT_temp_f#SI** from the **Type List**, and set **Data Point Name** to **NVC_freezing**, as shown in Figure 4.7.

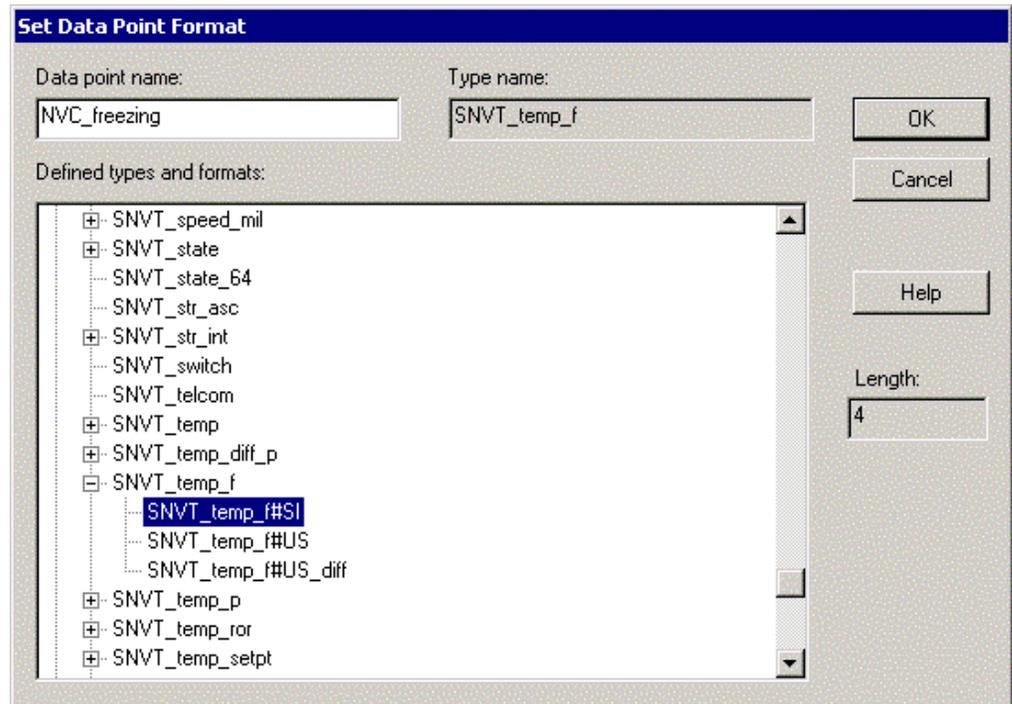


Figure 4.7 Set Data Point Format Dialog

- Click **OK**. The **NVC_freezing** data point appears in the **Points** list.
- Select the **Presets** tab.
- Select the **NVC_freezing** data point from the **Points** list.
- Click the **Default** field and enter **0**. Click **Apply**.

Configuring the Alarm Generator Functional Block

To configure the **Alarm Generator** functional block, follow these steps:

- Right-click the **Alarm Generator** functional block and select **Configure** from the shortcut menu. The *i.LON 100 Configuration Plug-in* opens with the **Alarm Generator** configuration dialog selected.
- Select the **Alarm Generator** window.
- Select **NVL_nviAgInput** from **Input Data Point** list, **Deviates From** from the second list, and **NVC_freezing** from the **Comparison** list. This causes the **High Limits**, **Deadband**, and **Low Limits** frames on the right side of the tab to become active.

NOTE: Be sure that the **NVL_nviAgInput** data point is associated with the **nviAgInput** network variable on the **Alarm Generator** functional block. You can verify this using the **Input Data Point** list. The data points that are on the selected **Alarm Generator** functional block are listed first under **Local Data Points**.

4. Set the **Enable Warning** and **Enable Error** check boxes in **Low Limits**. This causes an alarm to be generated when the value of the **Input Data Point** passes below a certain point.
5. Set **Offset 1** to **-4**. This causes an **AL_LOW_LMT_ALM_1** alarm to be generated when the **Input Data Point** (*i.e.* the temperature) passes below 4 degrees. The value in this field is subtracted from the **Comparison data point** to determine when the alarm is generated; **0 - (-4) = 4**.
6. Set **Offset 2** to 0. This causes an **AL_LOW_LMT_ALM_2** alarm to be generated when the **Input data point** passes below 0 degrees F (-18 degrees C).
7. Set the **Deadband** to 2. If one of the low limits is passed, the alarm condition will stay active until the **Input Data Point** is 2 degrees above the threshold. This prevents the heater from rapidly switching on and off as the temperature hovers around 4 degrees.
8. Set the **Priority** to 1.
9. Enter a description of the criteria in **Description**.

When you are finished, the **Alarm Generator** window appears as shown in Figure 4.8.

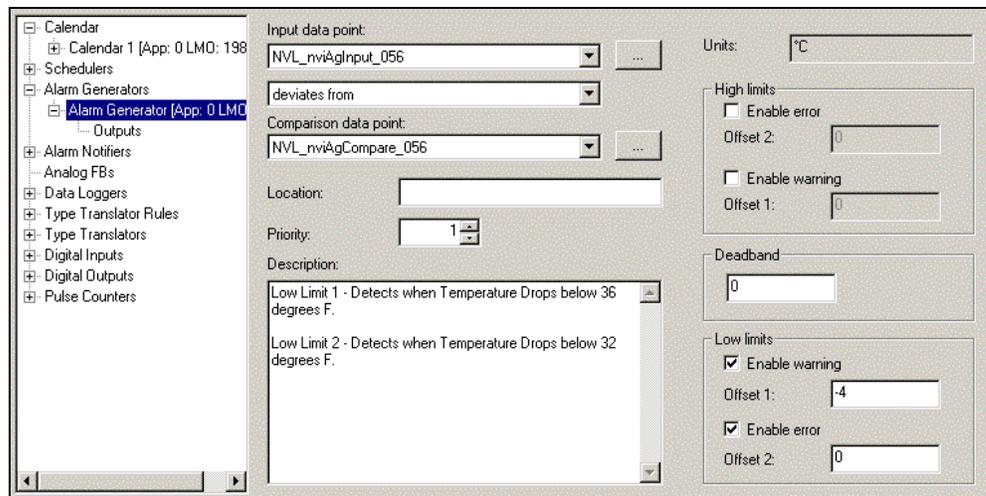


Figure 4.8 Alarm Generator Window

10. Select the **Output** window from below the **Alarm Generator**.
11. Set **Alarm Output Data Point** to **NVL_nvoAgAlarm**. This data point is associated with the **nvoAgAlarm** network variable on the **Alarm Generator** functional block (you should verify this as described in the note in step 3). The **Output** tab appears as shown in Figure 4.9.

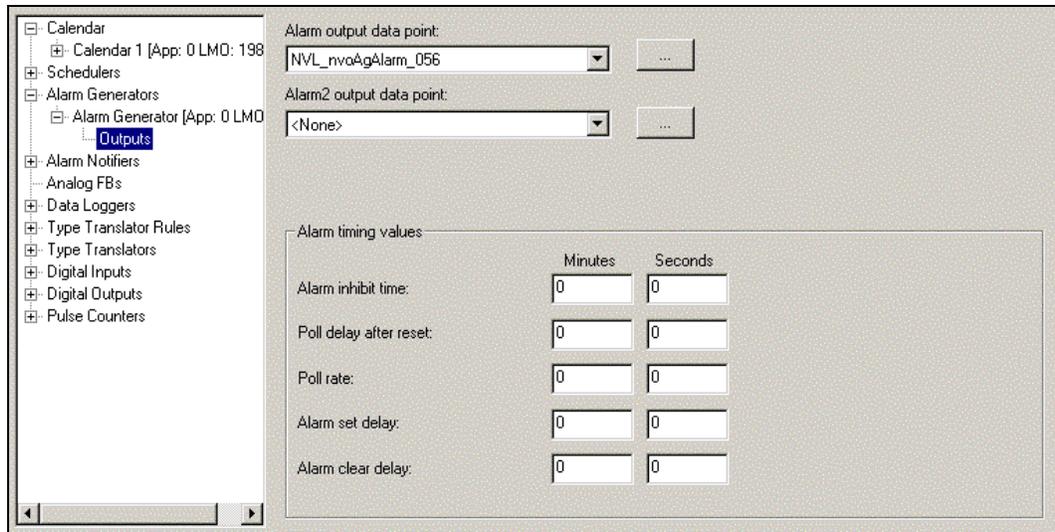


Figure 4.9 Output Tab

Use this tab to set the output data points for the Alarm Generator and to set timing properties. Set **Poll Rate** to a positive value. Click **Help** for a description of this tab.

Configuring the Alarm Notifier Functional Blocks

To configure the **Alarm Notifier – Warning** and **Alarm Notifier – Emergency** functional blocks, follow these steps:

1. Expand **Alarm Notifiers** on the Applications tab, and then select the **Alarm Notifier – Warning** functional block. The Alarm Notifier configuration dialog opens.
2. Select the **Global** tab.
3. Enter a **Description** of the **Alarm Notifier – Warning** functional block.
4. Select the **Inputs** tab.
5. Delete all data points other than the data point associated with the **nviAnAlarm** network variable from the **Inputs** list. The remaining data point will have the name **NVL_nviAnAlarm_xxx**, where xxx is the functional block index of the **Alarm Notifier** functional block.
6. Select the **NVL_nviAnAlarm_xxx** data point from the **Inputs** list and clear the **Acknowledgement Required** and **Clear Required** check boxes.
7. Select the **Email** tab.
8. The **Enable Data Points** field displays the **SNVT_switch** data points defined on the selected functional block.
9. Select the **NVL_nviAnEnable** data point and set the **To**, **Subj**, and **Message** fields as shown in Figure 4.10.

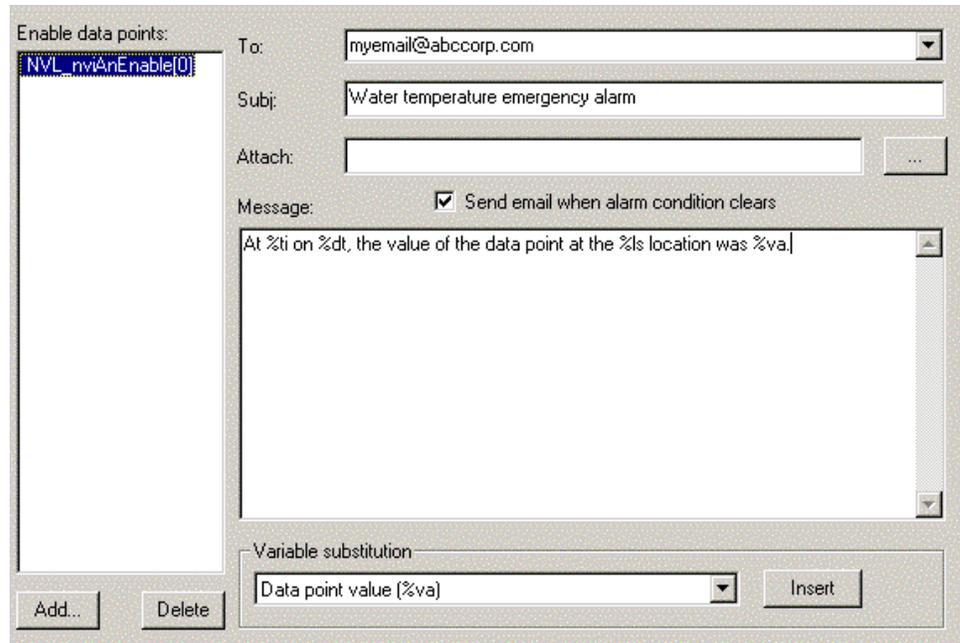


Figure 4.10 Email Tab

10. Enter your own email address in the **To** box. To send this email to multiple recipients, use a semi-colon as an address separator.

This example email message uses variable substitution to send specific information about the time and location of the alarm. You can insert selected substitutions from the **Variable Substitution** list when entering data into the **Message** field. Click **Help** for a complete list of the substitutions you can use.

When an alarm changes state (*i.e.* when it becomes active or when it becomes passive), the Alarm Notifier will send the emails that are associated with all data points in the **Enable Data Points** list that are set to On (**100.0 1**). The default value of the **nviAnEnable** input network variable is **100.0 1**, so if no other input is received on this network variable, the email shown above will be sent when the alarm is activated.

11. Select the **Outputs** tab.
12. The **Enable Data Points** list displays all the **SNVT_switch** data points on the selected functional block.
13. Click the **Add** button below the **Output Data Points** list to add output data points. The Add Data Point dialog opens. Select the data point associated with the **nvoAnAlarmFlag[n]** output network variable on the **Alarm Notifier – Warning** functional block, and then click **OK**.
14. Select the **Data Points** tab and the **Presets** sub-tab.
15. Select the data point associated with the **nvoAnAlarmFlag** network variable and define two presets, **On** and **Off**. Set **On** to a value of **100.0 1** and **Off** to a value of **0.0 0**, as shown in Figure 4.11.

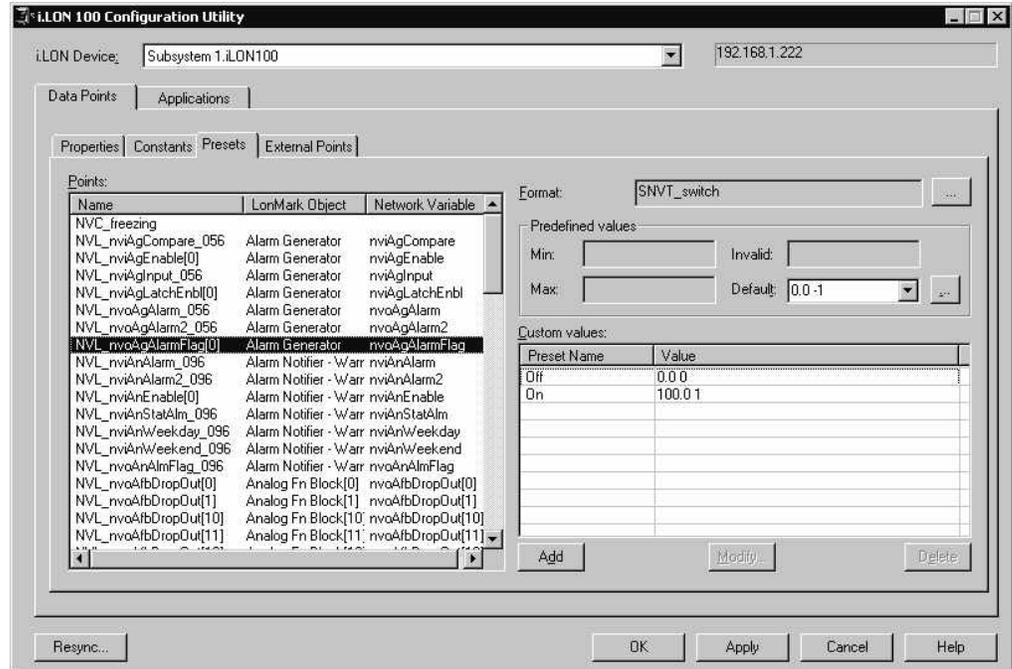


Figure 4.11 Data Point Presets

These presets are used to send output values from the Alarm Generator. See Chapter 4 of the *i.LON 100 e3 User's Guide* for more information on creating presets.

16. Select the **Applications** tab to return to configure the **Alarm Notifier – Warning** functional block.
17. In the **Outputs** tab, double-click the **Active Value** column next to the data point added in step 11 (**NVL_nvoAnAlarmFlag_xxx**), and then select **On** from the shortcut menu. The **Active Value** is the value to which the data point will be set when there is an active alarm. You can also enter the data value in this column directly (*i.e.* **100.0 1**).
18. Double-click the **Passive Value** column and select **Off** from the shortcut menu. The **Passive Value** is the value to which the data point will be set when there is no alarm.

The **Output** tab should now appear as shown in Figure 4.12.

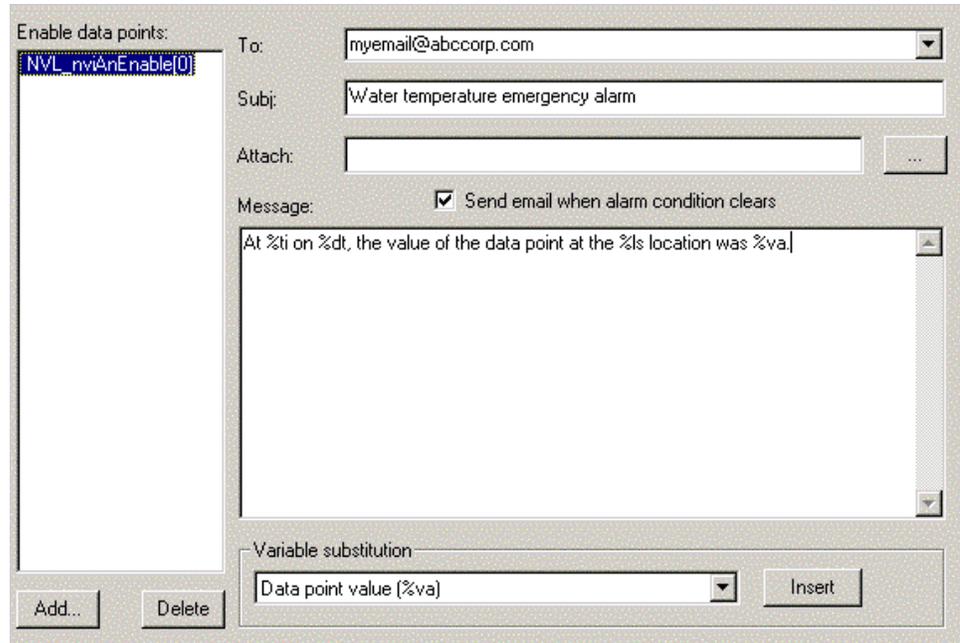


Figure 4.13 Email Tab

27. Enter your own email address in the **To** box.
28. Select the **Alarms** tab.
29. Select the **AL_LOW_LMT_ALM_1** alarm type and clear the **Enable Alarm** check box. This prevents the **Alarm Notifier – Emergency** functional block from registering an alarm when the first low limit is passed.
30. Click **Apply**.

Simulating an Alarm

To simulate an alarm, follow these steps:

1. Right-click the **Alarm Generator** functional block, and then select **Browse** from the shortcut menu. The LonMaker browser opens.
2. Click the **Monitor All** button on the toolbar. The browser starts monitoring all points in the browser window, allowing you to see network variable value updates as they happen.
3. To simulate a warning level alarm, select the **nviAgInput** network variable, enter a value of 3, and then click the Set Value button (). The *i.LON* 100 server updates the **nvoAgAlarm** output network variable to a condition of **AL_LOW_LMT_ALM_1**. Be sure that the browser is displaying the value of **nviAgInput** in SI units (*i.e.* Celsius).
4. Open a Web browser (Internet Explorer 6 or better) and point it to the IP address of your *i.LON* 100 server.
5. Click **View** under the **Alarming** heading.
6. Select **Alarm History** from the **View** list. The following alarm appears at the bottom of the page, as shown in Figure 4.14.

ACK	CLR	Alarm time	Location	Point	Pri	Grp	Src	Value	Unit	Description	Comment
<input checked="" type="checkbox"/>	<input type="checkbox"/>	2002/08/23 07:30:23	iLON100	NVL_nviAnSNVTalarm1	0		1.3	35		Low limit alarm 1	

Figure 4.14 Alarm Entry

7. To simulate an emergency alarm, return to the LonMaker browser and set the **nviAgInput** value to -2. The *i.LON 100* server updates the **nvoAgAlarm** output network variable to a condition of **AL_LOW_LMT_ALM_2**.
8. To simulate no alarm condition, set the **nviAgInput** value to 40.
9. Return to the Web browser and then click **Clear** on the left side of the Web page.
10. Select **Alarm Notifier – Emergency** from the **Alarm Log** list. A new alarm appears in the list at the bottom of the page.
11. Set the CLR check box and click **Update Log** to clear the alarm.

Detecting Heartbeat Failures

You can use the **Alarm Notifier** functional block to detect heartbeat failures. To do this, follow these steps:

1. Drag an Alarm Notifier functional block shape from the *i.LON 100* stencil to the LonMaker drawing.
2. Determine what network variable will be monitored for heartbeat failure. If the selected network variable has an associated external data point on the *i.LON 100* device skip to step 4.
3. Connect the network variable to be monitored to the **nviAnStatAlm** input network variable on the Alarm Notifier functional block. You may need to change the type of the **nviAnStatAlm** network variable to do this.
4. Right-click the Alarm Notifier functional block created in step 1, and then select **Configure** from the shortcut menu. Enter the IP address or fully qualified host name of the *i.LON 100* server when prompted. The *i.LON 100* Configuration Plug-in opens.
5. Select the **Inputs** tab.
6. Remove all input data points from the Inputs tab by selecting them and clicking **Delete**.
7. Click **Add** and add the data point to be monitored for heartbeat failure. This is either the external data point identified in step 2 or the **NVL_nviAnStatAlm_xxx** data point on the Alarm Notifier functional block.
8. When there is a heartbeat failure, the Alarm Notifier will register an **AL_NUL** alarm condition. Configure the **Email** and **Output** tabs to respond as desired.

Latching, Acknowledging, and Clearing Alarms

In some situations, you may want an alarm to stay active even when the condition that caused the alarm has been returned to normal. One example of this is shown above, with the **Alarm Notifier – Emergency** alarm requiring a manual clear via the Web page.

If a piece of hardware fails, causing an alarm that causes a backup piece of hardware to be activated, the alarm should stay active until manually cleared.

The *i.LON 100* server provides three ways of implementing this behavior:

- Select the **Clear Required** check box on the **Inputs** tab of the **Alarm Notifier** configuration dialog tab. This method is demonstrated in the example above. Selecting this check box causes the **Alarm Notifier** to maintain the alarm state until the alarm is cleared using the **Alarm Summary** Web page (see the help for these Web pages for more information). The condition that caused the alarm should be removed or the alarm will just immediately be regenerated. To clear the alarm, a user must have access to the **View - Alarm Summary** Web page. You can grant this access with the **Security** Web page, as described in Chapter 2 of the *i.LON 100 e3 User's Guide*.
- Select the **Acknowledgement Required** check box on the **Inputs** tab of the **Alarm Notifier** configuration dialog tab. Selecting this check box causes the Alarm Notifier to maintain the alarm state until the alarm has been acknowledged with the **Alarm Summary** Web page. An alarm can be acknowledged even while the alarm condition is still true. However, the Alarm Notifier will not terminate the alarm until the condition has been removed.
- Use the **nviAgLatchEnable** network variable on the **Alarm Generator Functional Block**. When this **SNVT_switch** network variable is set to On (100.0 1), the **Alarm Generator's nvoAgAlarmFlag** network variable will remain set to On (100.0 1) even after the alarm condition has ended. Set this network variable to Off (0.0 0) to have the **nvoAgAlarmFlag** network variable reflect the state of the alarm.

4

Scheduling

This chapter describes how to use the *i*.LON 100 scheduling applications to create an Event Scheduler. You can use Event Schedulers to schedule daily, weekly and monthly updates to the data points on your network.

Scheduling Overview

The *i*.LON 100 contains three types of functional blocks that you can use for scheduling applications — the **Scheduler**, **Calendar**, and **Real Time Clock** functional blocks. The **Scheduler** and **Calendar** functional blocks allow you to define *event schedulers* that specify the *schedule events* for each day over a period of time. Each Event Scheduler consists of a *day-based* schedule and a *date-based* schedule.

Day-based schedules occur every week on a specified day. Each Event Scheduler can contain up to seven day-based schedules (*i.e.* one for each day of the week). You can assign a single day-based schedule to multiple days of the week (*i.e.* you could define one day-based schedule for all weekdays and one for weekends). You can create a *weekly schedule* by assigning a day-based schedule to each day of the week.

Date-based schedules occur on a set of dates defined by an *exception*. Exceptions are ranges of dates such as “January 12th to February 2nd” or “Every other Monday” or “the third Monday of every month.” The exceptions defined on the calendar can be used by any number of schedulers.

The **Real Time Clock** functional block maintains the current date and time that is used by the **Scheduler** and **Calendar** functional blocks, and can also provide date and time data to other devices. The *i*.LON 100 server includes one **Real Time Clock**, one **Calendar** functional block, and 40 **Scheduler** functional blocks. In this chapter, these three functional blocks are collectively referred to as the *Event Scheduler* application.

Creating a Schedule

A typical schedule consists of a weekly schedule that defines the daily schedule to be used for each day of the week, and unlimited exception schedules that allow alternate daily schedules to become active on certain user-specified dates. The steps involved in creating a schedule are:

1. Plan your schedule. For example, “On weekday mornings at 6:00 AM the heating system should be started and the thermostat set to 65 degrees. At 8:00 AM, the thermostat should be set to 70 degrees.” etc. See *Planning Out Your Schedule* for more information.
2. Create a new Event Scheduler.
3. Using the schedule written in step one, decide what data points need to be written by the scheduler.
4. Define presets for each of the data points that are to be updated by the Event Scheduler. You can also enter the actual values of the data points, but presets may be easier to use.
5. Define a schedule for each day of the week. Each daily schedule consists of a series of *events*. An *event* consists of a time and a value (*i.e.* 06:00:Warmup). When an event occurs, the specified data point will be updated to the specified value. Each day of the week can have a different schedule.
6. Define exceptions to the daily schedules created in step 5. Exceptions are date ranges during which the weekly schedule will be overridden. These exceptions can be specific ranges of dates (*i.e.* “March 27th-April 6th, 2002”) or recurring dates that occur in a certain pattern (*i.e.* “The second

Tuesday of every month” or “Every other Saturday”). For each of these exception schedules, use the **Scheduler** functional block to define the schedule that will be used on that day.

Planning Out Your Schedule

Before implementing a schedule using the **Scheduler** and **Calendar** functional blocks, you should write out the schedule so you have a full understanding of all the schedules and events you will need to create.

Example of Planning Out a Schedule

The following is an example of a simple schedule that might be created to schedule HVAC and lighting controls in a retail store:

- On weekdays, the heat should be set to 60 degrees at 7:00 AM (Warmup), 70 degrees at 9:00 AM (Open), and turned off at 7:00 PM (Closed).
- On weekdays, the lighting should be turned on at 9:00 AM (Open) and turned off at 7:00 PM (Closed).
- On weekends, the heating and lighting should remain off.
- On the first Sunday of each month, for inventory, the heat should be set to 65 degrees at 9:00 AM (OpenInventory) and turned off at 6:00 PM (Closed).
- On the first Sunday of each month, for inventory, the lighting should be turned on at 9:00 AM (OpenInventory) and turned off at 6:00 PM (Closed).
-
- December 23rd to January 3rd, but you would like to be able to change the dates of the vacation every year.

The Scheduler Functional Block

Before attempting to create an Event Scheduler with the Configuration Plug-In, you should review the background information presented in the next few sections. The *i.LON 100* includes 40 **Scheduler** functional blocks. You can use a scheduler to generate time-based updates to the data points in your network. You will select a data point or group of data points for each scheduler you create.

You will define daily and exception daily schedules on each scheduler. You can create up to seven daily and two exception schedules on a single scheduler. Each schedule is identified by a unique name. For example, you could define two daily schedules, one named “Weekday” (to occur every weekday) and one named “Weekend” (to occur every weekend day) and two exception schedules, “Holiday” (to occur on specified holiday dates) and “Inventory” (to occur on a specified regular interval). The dates for the “Holiday” and “Inventory” schedules are derived from exceptions defined in the **Calendar** functional block.

A schedule event specifies a time and a value. At the specified time, the value is sent to all of the scheduler’s output data points (defined on the **Outputs** tab); if the scheduler drives output data points of different types, you will need to define presets for each of the data points. For example, you can create a scheduler that sets two data points – one SNVT_switch data point and one SNVT_temp_f – to a MorningWarmup preset at 6:00 on weekdays, a DayTime preset at 8:00 on weekdays, and a NightWeekend preset at 18:00 on weekdays; on weekends, the data points are left set to the NightWeekend preset on weekends. Using the **Presets** tab, you would define the presets for these data points as shown in Table 5.1.

Table 5.1 Preset Samples

	MorningWarmup	DayTime	NightWeekend
SNVT_switch	100.0 1	100.0 1	0.0 0
SNVT_temp_f	25	22	18

You can create exception schedules that become active based on *exceptions* that you define using the **Calendar** functional block. You can use exceptions to select exception schedules that become active on specific dates that require a unique schedule, such as holidays or special events, and assign them a unique schedule within your scheduler. You can set up a separate daily schedule for each exception. This allows you to specify what value you want your data points to use on each exception date at any given time.

When using exceptions, multiple schedule events may become active at the same time. You will assign a *schedule priority* to each schedule event. This value can be from 0 (highest priority) to 255 (lowest priority). If two schedules conflict—for example a daily schedule and an exception schedule—the one with the higher priority takes precedence.

You can create one or more outputs from a scheduler by adding dynamic network variable outputs to the Scheduler functional block. You can connect these outputs to any input network variables that require activation on a scheduled interval.

Each **Scheduler** functional block contains a single **nviEsEnable** network variable input as shown in Figure 5.1.

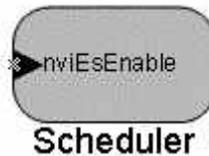


Figure 5.1 nviEsEnable

The **nviEsEnable** input is a **SNVT_switch** network variable that disables the **Scheduler** functional block. To disable a Scheduler functional block (*i.e.* to not update the points as specified in the *i.LON 100 Configuration Plug-in*), set this network variable to Off (**0.0 0**). In the absence of any input, this network variable defaults to On (**100.0 1**), so the **Scheduler** functional block will be enabled unless explicitly disabled.

To generate outputs from a scheduler, use an LNS installation tool such as the LonMaker tool to add one or more output dynamic network variable outputs to it, define preset values for these outputs and then configure the schedules for these outputs, as described later in this chapter.

The Calendar Functional Block

The *i.LON 100* includes 1 **Calendar** functional block. You can use this functional block to create up to 256 exception exceptions for your Scheduler functional blocks. Each exception that you create represents a date, or a group of dates. When you reference an exception in a scheduler, you will be able to assign an exception schedule to each exception. You can use exceptions to assign alternate daily schedules to use on particular dates such as holidays or special events.

The **Calendar** functional block contains a single **nviEcDateResync** network variable input and a single **nvoEcDateEvent** output as shown in Figure 5.2.

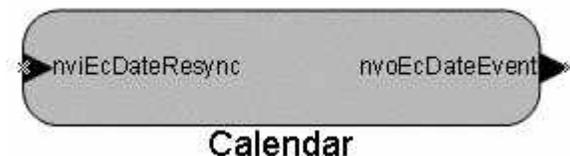


Figure 5.2 Calendar Functional Block Network Variable

The **nvoEcDateEvent** output network variable reports the status of each exception that you define. This output is internally connected to the *i.LON 100* Node Object functional block, and to all the *i.LON 100* **Scheduler** functional blocks. Every day at 12:00pm (noon) it resynchronizes by sending all **Scheduler** functional blocks updates informing them of the exception status and time to the next date event. When the clock is set manually before 12:00pm on a particular day, the **Calendar** will resynchronize immediately. After this irregular resynchronization, the daily resynchronization will be cancelled for that day, and it will resume at noon on the following day.

Use this network variable to synchronize schedules on multiple *i.LON 100* servers, as described in the *Using the Same Exceptions for Multiple i.LON 100* section later in this chapter.

The **nviEcDateResyn** network variable is a **SNVT_switch** input network variable. The network variable requests an update on the status of all defined exceptions. When the state of this input changes from Off to On, the **Calendar** functional block sends a series of updates on the **nvoEcDateEvent** output, one for each defined exception. The input is On when the state value is one (1) and the level value is greater than zero (0). This input is internally connected to the *i.LON 100* Node Object functional block and all the *i.LON 100* **Scheduler** functional blocks.

Use this network variable to synchronize schedules on multiple *i.LON 100* devices, as described in *Using the Same Exceptions for Multiple i.LON 100*, later in this chapter.

If you are using the *e3* software, Echelon recommends that you use the Web pages to configure your Event Calendars, as the Event Calendar application instance you are configuring may not be compatible with the Configuration-Plug-In.

Using the Same Exceptions for Multiple *i.LON 100* Servers

If you have multiple *i.LON 100* devices in the same LONWORKS network that use the same exception schedule (holidays, maintenance days, etc.), you can bind the **Calendar** functional block of the source *i.LON 100* server to the **Node Object** functional block of the destination *i.LON 100*. To do this, follow these steps:

1. Connect the **nvoEcDateEvent** output network variable on the source *i.LON 100* device's **Calendar** functional block to the **nviDateEvent** input network variable on the destination *i.LON 100* device's **Node Object** functional block.
2. Connect the **nvoDateResync** output network variable on the destination *i.LON 100* device's **Node Object** functional block to the **nviEcDateResync** input network variable on the source *i.LON 100*

device's **Calendar** functional block.

In addition, the configuration of the **Calendar** functional block is stored in the **/root/config/software/EventCalendar.xml** file. If you want to copy schedule information to an *i.LON 100* that is not in the same LONWORKS network, you can configure the **Calendar** on one *i.LON 100* server (*i.e.* define all exceptions) and FTP this file to the same location to all other *i.LON 100* server that should use the same schedule.

The Real Time Clock Functional Block Shape

The *i.LON 100* includes 1 **Real Time Clock** functional block. It maintains the current date and time. It includes the input and output network variables shown in Figure 5.3.

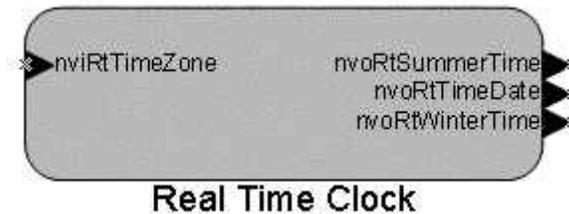


Figure 5.3 Real Time Clock Network Variables

nviRtTimeZone is a **SNVT_time_zone** input network variable that sets the time zone. You can also set the time zone on the **Setup - Time** Web page as described in Chapter 2 of this document.

nvoRtTimeDate is a **SNVT_time_stamp** output network variable that reports the current date and time. You can set the current date and time (configure the *i.LON 100* to obtain this information from an SNTP server) on the **Setup - Time** Web page, as described in Chapter 2 of this document.

nvoRtSummerTime is a **SNVT_time_stamp** output network variable that reports the date of **nvoRtWinterTime** is a **SNVT_time_stamp** output network variable that reports the date of the end of daylight savings time.

Creating a Schedule

A typical schedule consists of a weekly schedule that defines the daily schedule to be used for each day of the week, and up to two exception schedules that allow alternate daily schedules to become active on specified dates. Before creating a schedule, you should already have created a LONWORKS network that controls the functions you wish to schedule. To create a schedule with the Configuration Plug-In, follow these steps:

1. *Plan Out Your Schedule* – Write out your desired schedule. For example, “On weekday mornings at 06:00 the heating system should be started and the thermostat set to 65 degrees F (18 degrees C). At 08:00, the thermostat should be set to 70 degrees F (21 degrees C).” etc. See the *Planning Out Your Schedule* section on page 39 for an example schedule.
2. *Add a Scheduler Functional Block to the LonWorks Network* – Use an LNS installation tool such as the LonMaker tool to add a Scheduler functional block to the network drawing. See the *Adding a Scheduler Functional Block to a Network* section later in this chapter for more information.

3. *Define Outputs* – Using the schedule written in step one, decide what data points in your network need to be written by the scheduler. Optionally add dynamic output network variables to the **Scheduler** functional block and then add the appropriate data points to the **Scheduler** functional block’s **Data Points** window. Connect these output network variables to the corresponding input network variables. See the *Defining Outputs* section later in this chapter for more information.
4. *Define Presets* – Use the **Data Point Presets** window of the **Scheduler** functional block to define presets for each of the data points defined on the **Data Points** window. See the *Defining Presets* section later in this chapter for more information.
5. *Create Daily Schedules* – Use the **Scheduler** functional block to define a schedule for each day of the week. Each daily schedule consists of zero or more *events*. An *event* consists of a time and a preset value (*i.e.* 06:00:Warmup). When an event occurs, the data points specified on the **Data Points** window will be set to the values associated with the preset on the **Data Points Presets** window. Each day of the week can have a different schedule. See the *Creating Daily Schedules* section later in this chapter for more information.
6. *Create Exceptions* – Use the **Calendar** functional block to define exceptions: date ranges during which the weekly schedule will be overridden. These *exceptions* can be specific ranges of dates (*i.e.* “March 27th-April 6th, 2002”) or dates in a certain pattern (*i.e.* “The second Tuesday of every month” or “Every other Saturday”). For each of these exception schedules, use the **Scheduler** functional block to define the schedule that will be used on that day. See the *Creating Exceptions* section later in this chapter for more information.
7. *Optionally Create One-time Overrides* – You can optionally use the **Scheduler** functional block to create a daily schedule that will override the schedule for a specified range of dates. See the *Creating a One-time Override Schedule* section later in this chapter for more information.

Adding a Scheduler Functional Block to a Network

To add a **Scheduler** functional block to a LONWORKS network, follow these steps:

1. Open the network that controls the functions you want to schedule with an LNS installation tool such as the LonMaker tool.
2. Use an existing *i.LON 100* device in the network, or create and commission a new *i.LON 100* device.
3. Add a **Scheduler** functional block. For example, using the LonMaker tool, drag a Scheduler functional block shape from the *i.LON 100* Shapes stencil to the LonMaker drawing. The LonMaker New Functional Block wizard appears.
4. Associate the **Scheduler** functional block shape with one of the **Scheduler** functional blocks on the *i.LON 100* device. See the installation tool documentation for more information.

Defining Outputs

To define outputs, follow these steps:

1. Determine which data points should be modified by the scheduler.
2. If these data point represent input network variables on a LONWORKS network, you can optionally create complimentary dynamic output network variables on the **Scheduler** functional block. You may want to create a single output network variable to connect to multiple input network variables (for example, if you want to send a temperature setpoint to multiple heating controllers on the network). See the installation tool documentation for more information on creating dynamic network variables.
3. Start the *i.LON 100 Configuration Plug-in* as described in Chapter 3, *Configuring the i.LON 100 Applications*.
4. Open the **Applications** tab, and select **Data Points** beneath the **Scheduler** functional block you created in the previous step. The **Data Points** dialog appears, as shown in Figure 5.4.

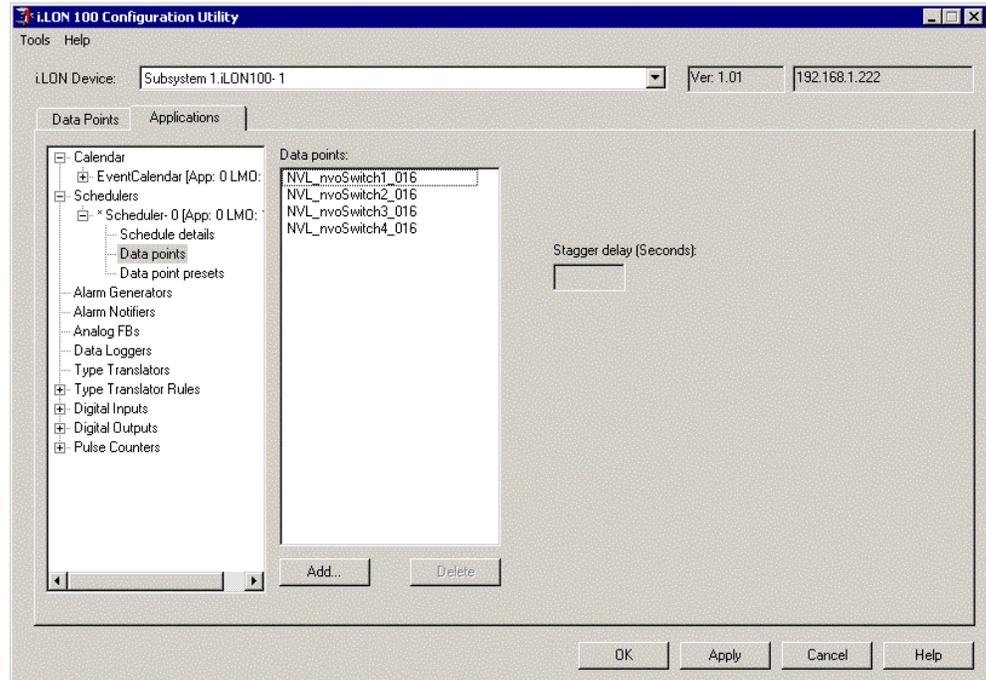


Figure 5.4 Data Point Dialog

5. Click **Add**. The **Add Data Point** dialog opens. This dialog shows all data points on the *i.LON 100* server.
6. Press and hold the <CTRL> key and then select each of the data points associated with the network variables you created in step 2. You can click on the column headings in the **Add Data Point** dialog to sort the data points according to different criteria—this may make it easier to find the correct data points.
7. Optionally set a **Stagger Delay** for each data point. This is the amount of time that the Scheduler will wait before updating the specified data point. This allows you to gradually bring up or bring down a system. For example, imagine a schedule that controls the power for 100 stores in a

mall. The schedule indicates that power should be turned on at 8:00. However, turning on power for 100 stores at once can cause a power surge. To avoid this, use varying **Stagger Delays** for the different points to bring power up 1 or 2 stores at a time.

8. Click **OK**. The data points will be added to the **Data Points** list.

Example of Defining Outputs

In the sample schedule described above, two outputs are required, one to control the setpoint on the HVAC controller, and one to turn the lights on and off. Two output network variables are dropped on the **Scheduler** functional block, one of the network variable type used by the HVAC controller on the network (**nvoTemp**) and one of the network variable type used by the light controller on the network (**nvoLights**). If you are using the LonMaker tool, the **Scheduler** functional block now appears as shown in Figure 5.5.

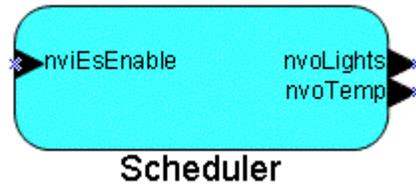


Figure 5.5 Scheduler Functional Block

When you add the two data points associated with the output network variables to the **Data Points** pane, the list appears as shown in Figure 5.6.

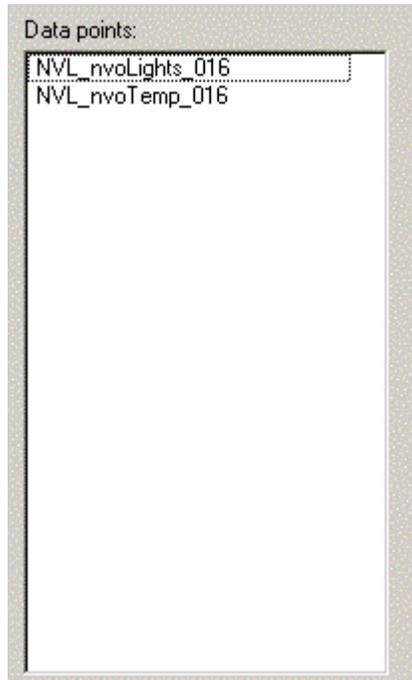


Figure 5.6 Data Point List

Defining Presets

To define presets for the outputs configured on the **Data Points** page, follow these steps:

1. Define each event that will be triggered by any schedules. In the *Example of Planning Out a Schedule* procedure, a name was given to each of the events. Use your written out schedule to determine what events will occur and what state the outputs should be set to for each event.
2. Under the appropriate **Scheduler** in the applications tab, select **Data Point Presets**. The **Data Point Presets** dialog opens, as shown in the Figure 5.7.

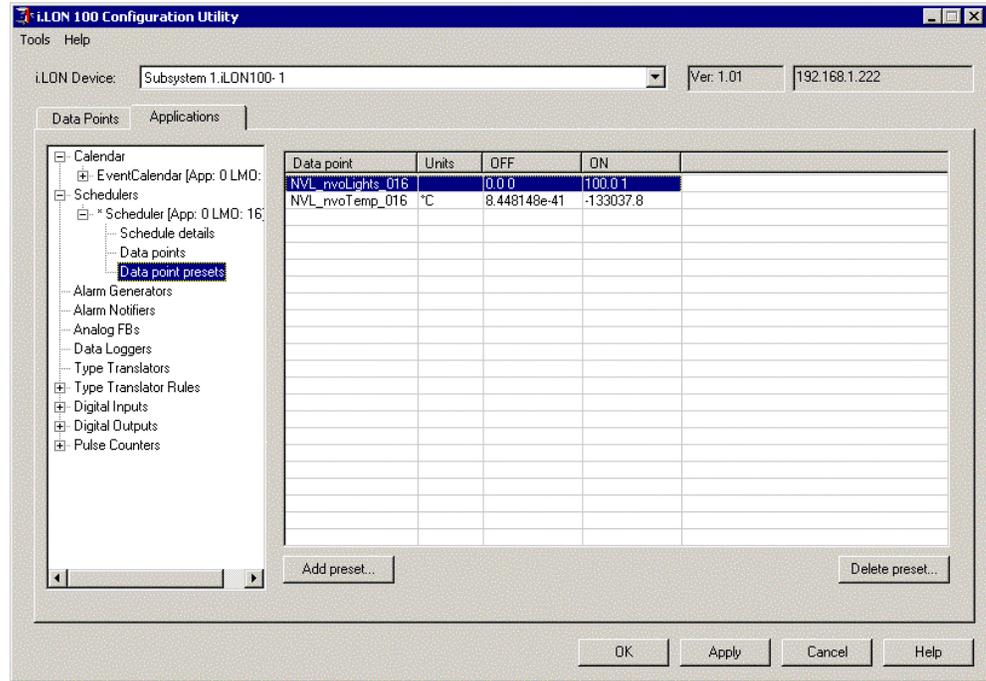


Figure 5.7 Data Point Presets Dialog

All data points added to the **Data Points** list in the previous step will be listed on the dialog. Some presets may already have been defined on the **Data Points** tab of the *i.LON 100 Configuration Plug-in*.

3. Click **Add Preset**. The **Add Preset Definition** dialog appears, as shown in Figure 5.8.



Figure 5.8 Add Preset Definition Dialog

4. Enter the name of the preset in **Preset Name**. Leave **Copy Values from Preset** set to **<None>**.

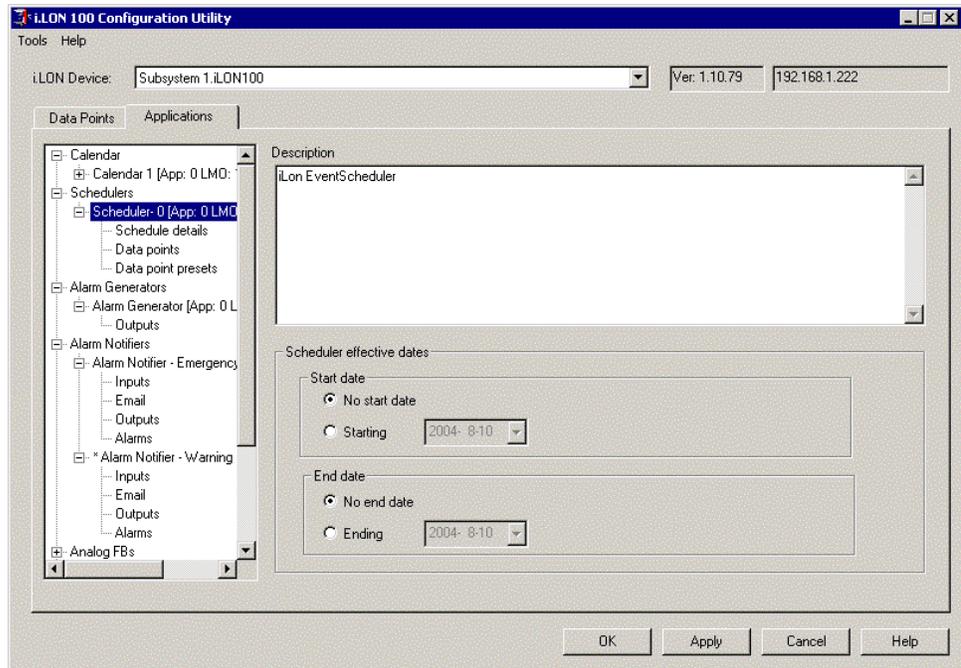


Figure 5.10 i.LON 100 Configuration Utility Scheduler Dialog

2. Select the **Start Date** and **End Date** for this schedule.
3. Click **Schedule Details** under the selected Scheduler in the Applications list. The **Schedule Details** dialog opens, as shown in Figure 5.11.

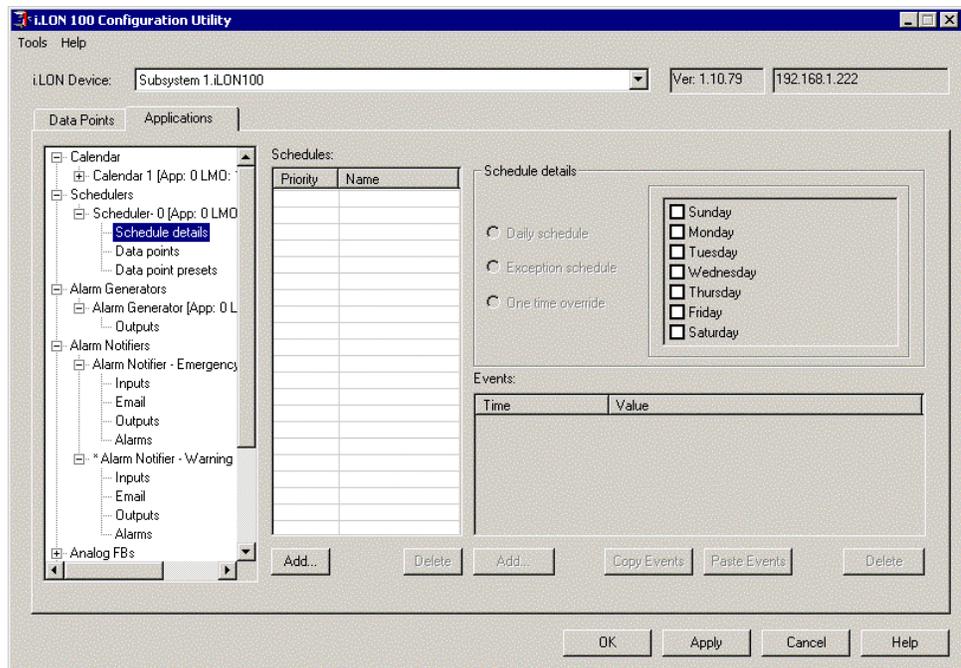


Figure 5.11 Schedule Details Dialog

4. Click **Add** under the **Schedules** list to open the **Add New Schedule** dialog, shown in Figure 5.12.

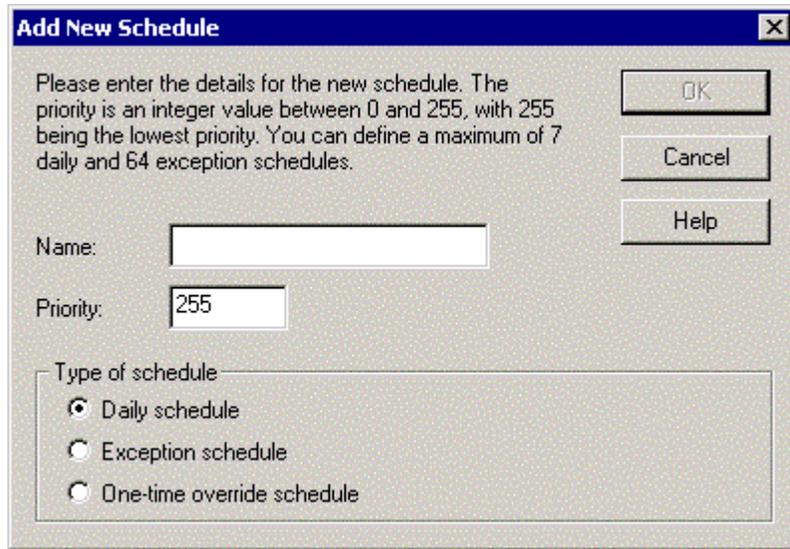


Figure 5.12 Add New Schedule Dialog

5. Configure the properties on the Add New Schedule dialog. Table 5.2 describes these properties.

Table 5.2 Add New Schedule Properties

Property	Description
Name	The name of the new daily schedule. This can be up to 227 characters.
Priority	The priority of the new daily schedule. This value can be from 0 (highest priority) to 255 (lowest priority). If two daily schedules conflict — for example a daily schedule and an exception — the one with the higher priority value takes precedence.
Type of Schedule	When creating a daily schedule, select Daily Schedule .

6. Click **OK** to create the new daily schedule. The new daily schedule appears in the **Schedules** list.
7. Select the days of the week to which this schedule will apply using the check boxes on the right side of the dialog. Select the check box for all days that this daily schedule should be active.
8. Add one or more events to the schedule. To add an event, click **Add** under the **Events** list or double-click the first column of the first empty row in the **Events** list. The **Add New Event** dialog opens, as shown in Figure 5.13.

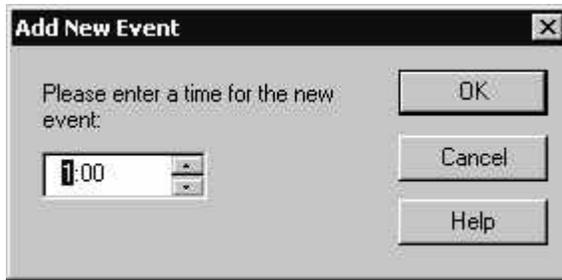


Figure 5.13 Add New Event Dialog

9. Set the time for the new event and then click **OK**. A new event appears in the **Events** list. The **Time** will be the time you selected, and the **Value** will be empty.
10. Select a preset value for the new event. This should be one of the presets created in the previous step.
11. Create additional daily schedules. For example, you can create one schedule for Monday through Thursday, one schedule for Friday, and one schedule for the weekend. You can create up to 7 daily schedules in a single Scheduler functional block.
12. Click **OK** or **Apply** to save the changes to the *i.LON 100* device.

Example of Creating a Daily Schedule

Figure 5.14 shows the weekday schedule created to match the example described previously.

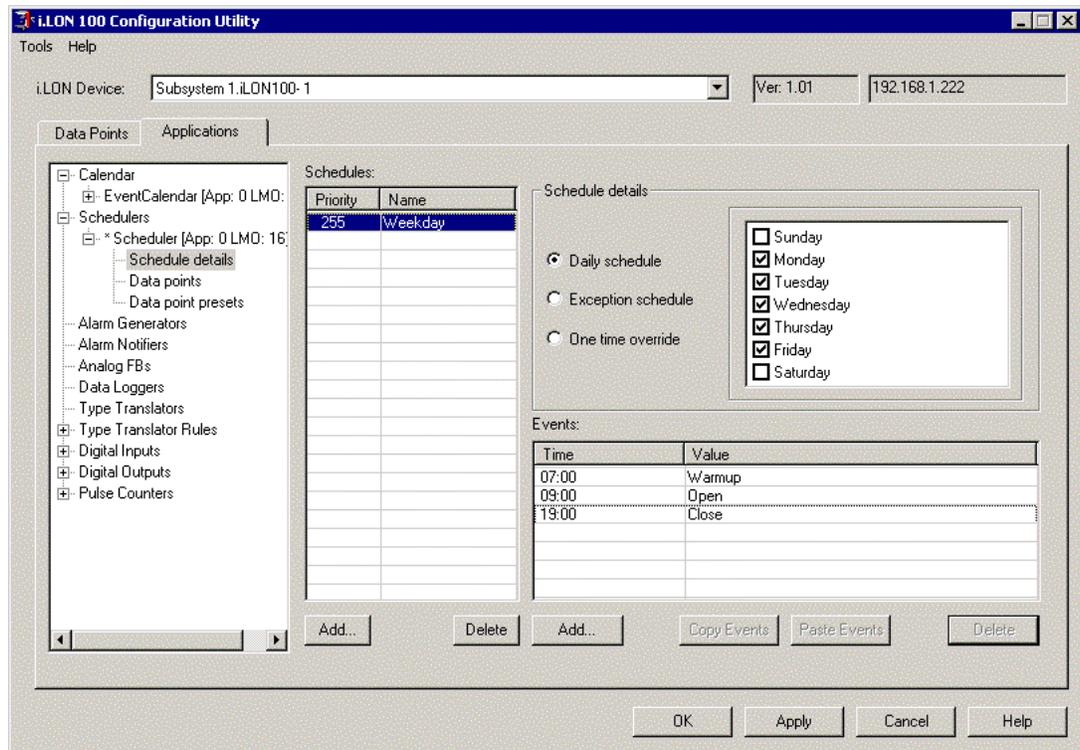


Figure 5.14 Example Daily Schedule

Creating Exceptions

Once you have created daily schedules for each day of the week, you can add exceptions to the schedule. You can create an exception using the **Calendar** functional block, and then add it to one or more **Scheduler** functional blocks. You do not have to add a **Calendar** functional block shape to the LonMaker drawing in order to be access the **Calendar** functional block from the *i.LON 100* Configuration Plug-in.

You can create two kinds of exceptions: *one-time exceptions*, and *recurring exceptions*. One-time exceptions specify a single range of dates. For example, if a building were being shut down for two weeks for maintenance, you could create a one-time exception for those dates. Recurring exceptions occur at regular intervals. For example, you can create a recurring exception for a yearly holiday or for an inventory procedure that takes place on the second Monday of every month.

To create an exception, follow these steps:

1. Open an existing LonMaker network or create a new LonMaker network as described in the *LonMaker User's Guide*.
2. Use an existing *i.LON 100* device in the LonMaker network, or create and commission a new *i.LON 100* server.
3. Right-click the *i.LON 100* server shape and select **Configure** from the shortcut menu. The *i.LON 100* Configuration Plug-in opens.
4. Select the **Applications** tab, and then select **One-time Exceptions** or **Recurring Exceptions** from beneath the **Calendar** application on the left side of the *i.LON 100* Configuration Plug in.

NOTE: If you are using the *e3* software and you attempt to configure an Event Calendar that has previously been configured with the Web pages, a dialog will appear to warn you that the Event Calendar may be incompatible with the Configuration Plug-In. Echelon recommends that you use the Web pages to configure your Event Calendars whenever possible.

5. On either tab, click **Add** to open the **New Exception** dialog, as shown in Figure 5.15.

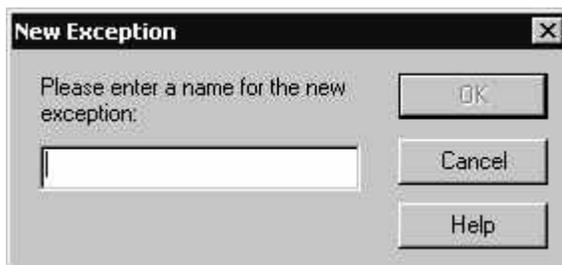


Figure 5.15 New Exception Dialog

6. Enter the name of the new exception and then click **OK**. The new exception will be added to the **Exceptions** list.
7. Select the new exception from the **Exceptions** list. The fields in **Exception Details** will highlight.
8. If you are in the **One-time Exceptions** tab, **Exception Details** dialog appears as shown in Figure 5.16.

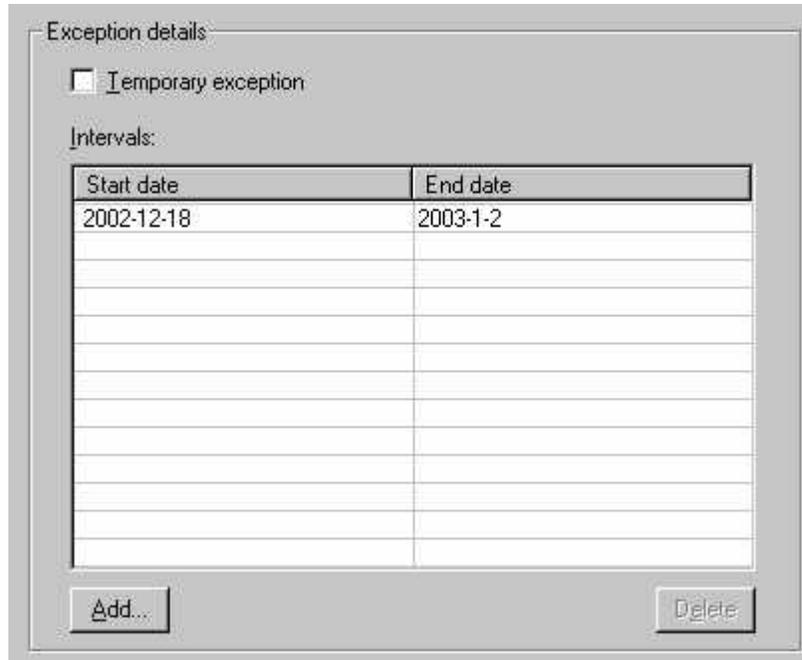


Figure 5.16 Exception Details Dialog

9. Use the **Start Date** and **End Date** columns to select the start and end dates for the exception. Each exception can have up to 16 date ranges. Select the **Temporary Exception** check box to have the exception removed once the last date specified in the **End Date** column has passed. If this check box is cleared, the exception will remain and the dates can be updated (*i.e.* if you have an exception schedule that will be used at irregular intervals).

If you are in the **Recurring Exceptions** window, the **Exception Details** dialog appears as shown in Figure 5.17.

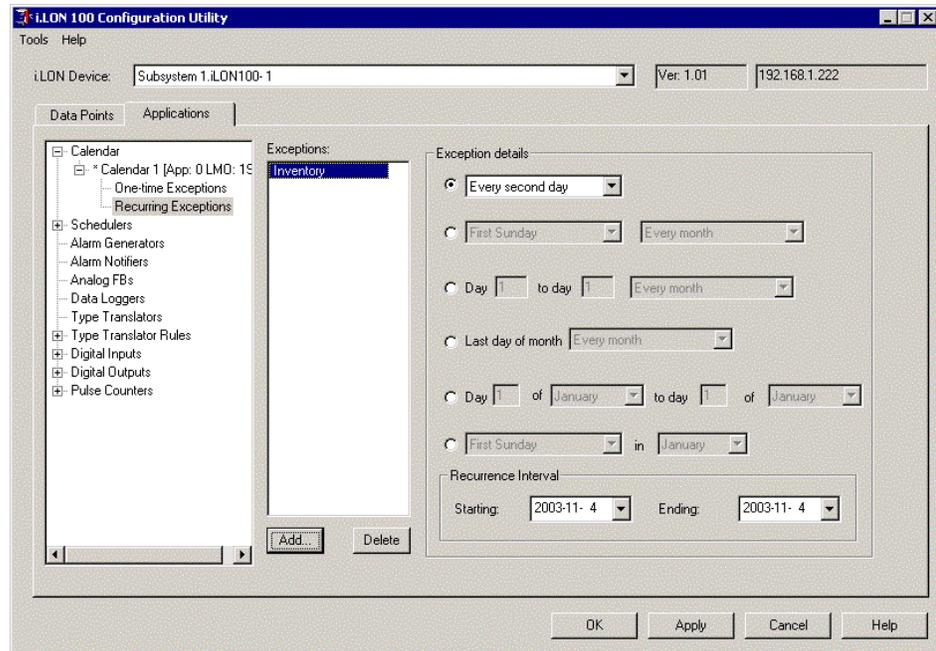


Figure 5.17 Exception Details Dialog

These fields allow you to set how often the exception occurs. Click **Help** for more information.

10. Once you have defined one or more exceptions, click the **Scheduler** tab and the **Schedule Details** tab. The exceptions you created are listed in **Exceptions**.
11. Click **Add** below the **Schedules** list to open the **New Schedule** dialog, as shown in Figure 5.18.

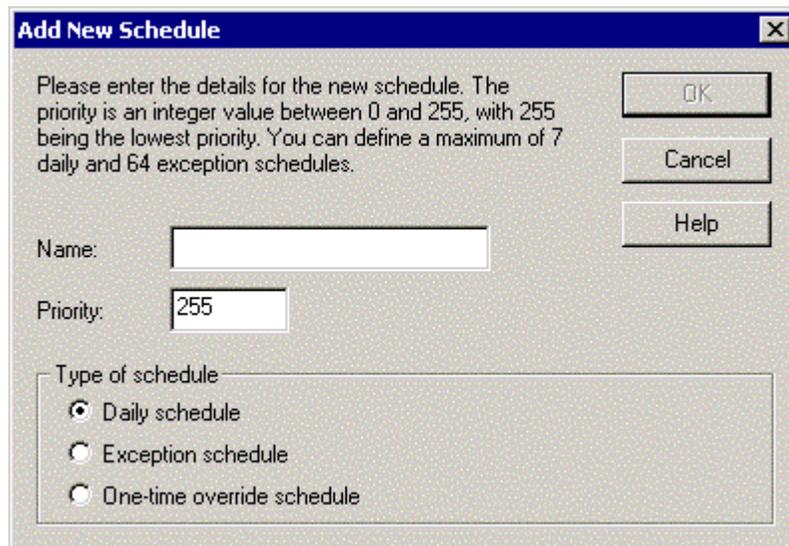


Figure 5.18 Add New Schedule Dialog

- Configure the properties on the Add New Schedule dialog. Table 6.3 describes these properties.

Table 6.3 Add New Schedule Properties

Property	Description
Name	The name of the new schedule. This field can be up to 227 characters.
Priority	The priority of the new schedule. This number can be from 0 (highest priority) to 255 (lowest priority). If two schedules conflict—for example a daily schedule and an exception schedule—the one with the higher priority value takes precedence. When creating an exception schedule, set the Priority higher than the priority used for the daily schedule.
Type of Schedule	When creating an exception schedule, select Exception Schedule .

- Click **OK** to create the new exception schedule. The new schedule will appear in the **Schedules** list.
- Select the new schedule from the **Schedules** list. Since you have created an exception schedule, the **Days** field is grayed out, while the **Exceptions** field is highlighted.
- Select the exceptions to which the new schedule will apply. Each schedule can apply to multiple exceptions. For example, if you created recurring exceptions for Christmas, the Fourth of July, Presidents Day, and New Year’s Day, you can create a holiday schedule that would apply to those dates.
- Optionally add one or more events to the schedule (you can leave a schedule empty to indicate that the scheduler should remain dormant on that day). To add an event, click **Add** under the **Events** list or double-click the first column of the first empty row in the **Events** list. The **Add New Event** dialog opens, as shown in Figure 5.19.

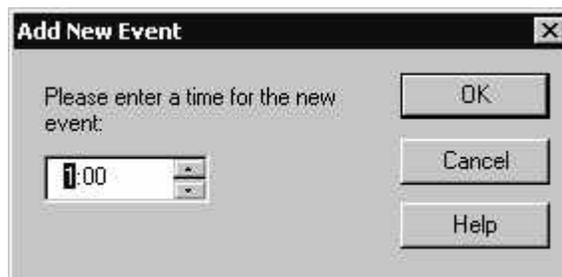


Figure 5.19 Add New Event Dialog

- Use this dialog to set the time for the new event and click **OK**. An event will appear in the **Events** list. The **Time** will be the time you selected, and the **Value** will be **Default**. This indicates that at the specified time, all the data points specified in the **Data Points** tab will be set to their **Default** value, as specified in the **Presets** tab.

18. To change the value to which the data points will be set, click the value to open a shortcut menu that allows you to set any of the preset values available in the **Presets** tab for the collection of data points in the **Inputs** tab. You can set the value for multiple data point of the different types as long as the preset value you specify is defined for each data point. You can also enter a value manually, but you must ensure that it is a valid value for all data points in the **Inputs** tab (the Configuration Plug-in will not validate it).
19. Create additional exception schedules as described above. You can create up to three exception schedules in a single Scheduler functional block.

Example of Creating Exception Schedules

In the example described above, two exception schedules are created, one recurring exception (Inventory, on the first Sunday of each month) and one one-time exception (Vacation, dates to be set manually every year).

The Recurring Exceptions window for the Inventory schedule appears as shown in Figure 5.20.

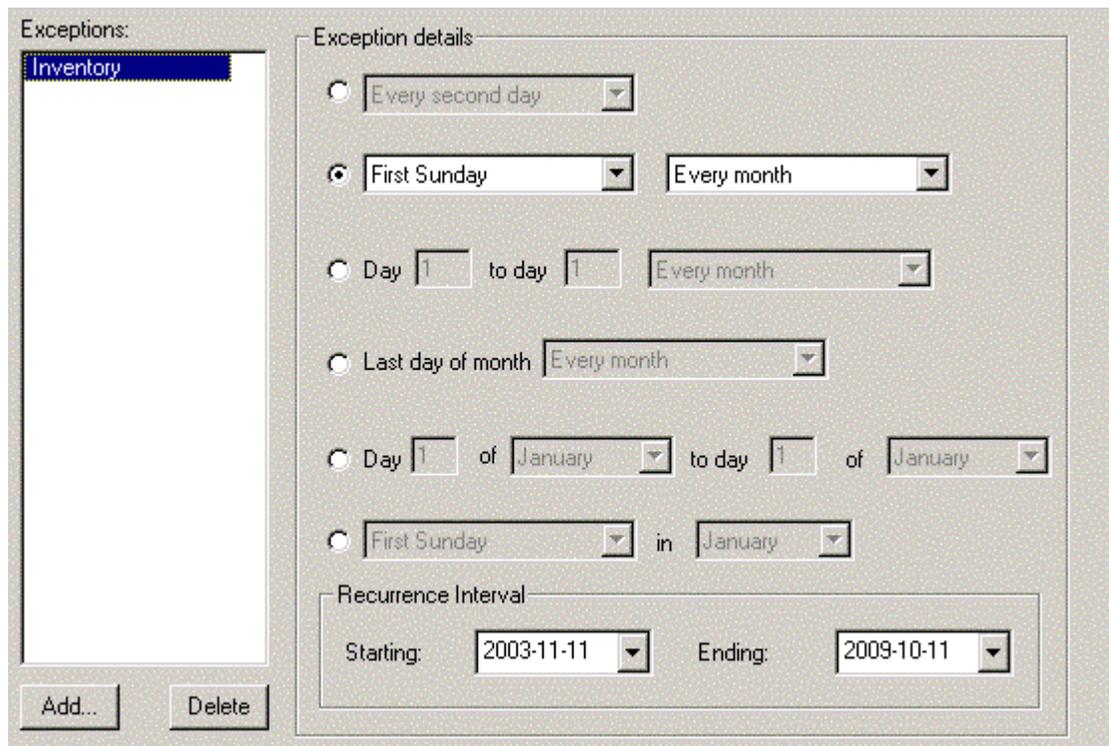


Figure 5.20 Recurring Exceptions Window

The One-time Exceptions window for the Vacation schedule appears as shown in Figure 5.21.

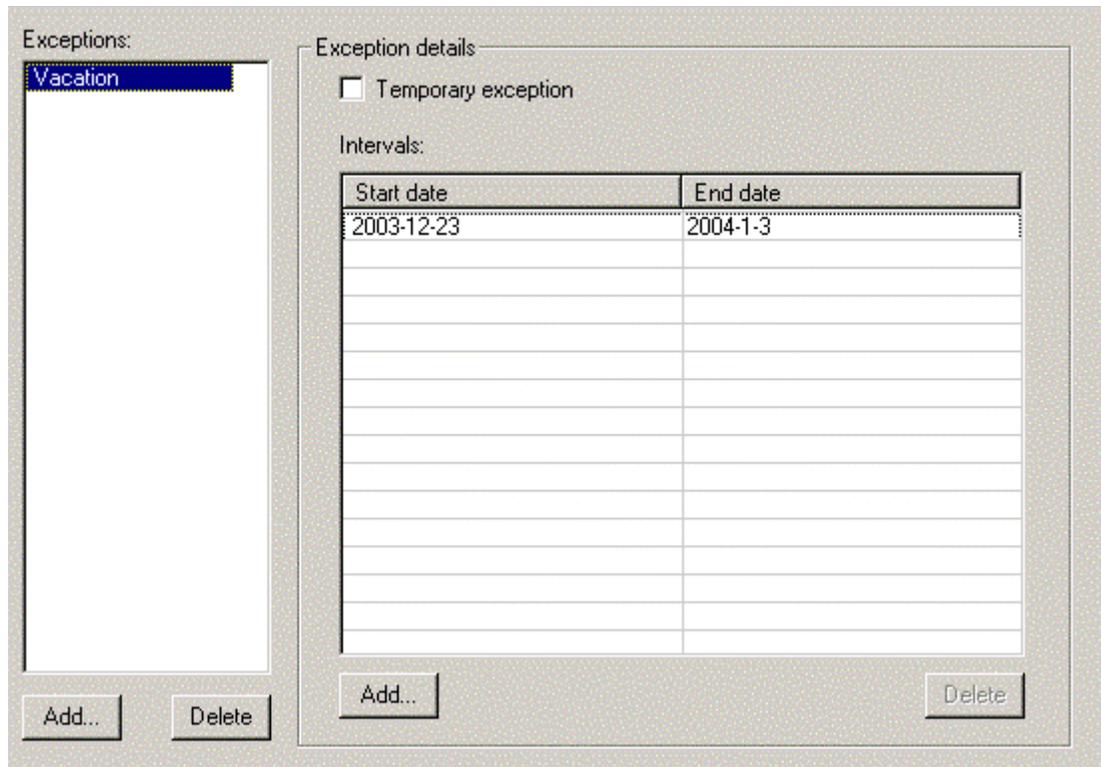


Figure 5.21 One Time Exceptions Window

Deleting Exception Schedules

You can delete one-time or recurring exceptions in the **Calendar** functional block by selecting the exception from the **Exceptions** list, and then clicking **Delete**. If you delete an exception, you should assure that you remove the reference to the exception from all **Scheduler** functional blocks on the *i.LON 100*. If a **Scheduler** references a non-existent exception, the **Scheduler** may be rendered inoperational.

Creating a One-time Override Schedule

You can create a one-time override schedule to implement a temporary exception schedule for a single range of dates. A one-time override schedule is never recurring. One-time override schedules count against the total of two exception schedules (*i.e.* exception schedules) that can exist on a scheduler, so you cannot create one if the scheduler already has two exception schedules defined. To create a one-time override schedule, follow these steps:

1. Right-click the **Scheduler** functional block that you want to override, and then select **Configure** from the shortcut menu. The *i.LON Configuration Utility* opens with the **Scheduler** tab selected.
2. Select the **Schedule Details** tab.
3. Verify that the correct Scheduler functional block is selected in the **Schedulers** folder in the **Functional Blocks** list, and then click **Add**. The **New Schedule** dialog opens as shown in the Figure 5.22.

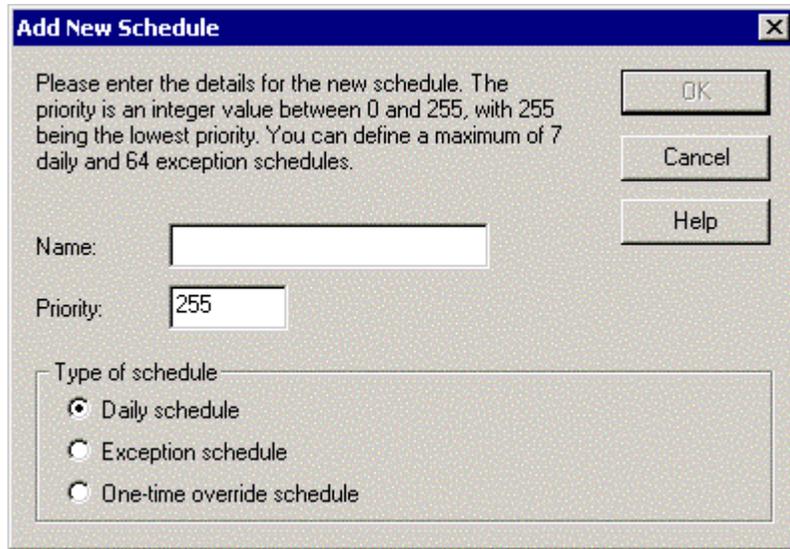


Figure 5.22 Add New Schedule Dialog

4. Configure the properties on the Add New Schedule dialog. Table 6.4 describes these properties.

Table 6.4 Add New Schedule Dialog Properties

Property	Description
Name	The name of the new schedule. Creating a one-time override schedule creates a temporary exception of the same name in the Calendar functional block, so the override schedule name must not match an existing exception name.
Priority	The priority of the new schedule. This number can be from 0 (highest priority) to 255 (lowest priority). If two schedules conflict—for example a daily schedule and a one-time override schedule—the one with the higher priority takes precedence. When creating a one-time override schedule, set the Priority higher than the priority used for the current day's daily schedule.
Type of Schedule	When creating a one-time override schedule, select One-time Override Schedule .

5. Click **OK** to create the one-time override schedule. The new schedule will appear in the **Schedules** list with an asterisk next to it to indicate it is a one-time override schedule.
6. Select the new schedule from the **Schedules** list. Enter the start and end dates for the override schedule under **One-time Exception Details**.
7. Optionally add one or more events to the schedule (you can leave a one-time override schedule empty to indicate that the Scheduler should remain dormant on that day). To add an event, click **Add** under the **Events** list or double-click on the first column of the first empty row in

the **Events** list. The **Add New Event** dialog opens, as shown in Figure 5.23.



Figure 5.23 Add New Event Dialog

8. Set the time for the new event and click **OK**. An event will appear in the **Events** list. The **Time** will be set to match the time you selected, and the **Value** will be **Default**. This indicates that at the specified time, all the data points specified in the **Data Points** tab will be set to their **Default** value, as specified in the **Presets** tab.
9. To change the value to which the data points will be set, click the value to open a shortcut menu that allows you to set any of the preset values available in the **Presets** tab for the collection of data points in the **Inputs** tab. You can set the value for multiple data point of the different types as long as the preset value you specify is defined for each data point. You can also enter a value manually, but you must ensure that it is a valid value for all data points in the **Inputs** tab (the Configuration Plug-in will not validate it).
10. Once you have finished entering events for the one-time override schedule, click **OK**. Once the dates of the one-time override schedule have passed, it will be removed automatically.

Recovering from a Power Outage

If the *i.LON* 100 server loses power and power is later restored, it will attempt to set the appropriate outputs given the current date and time. To do this, it will look at the schedule in use for that day and the most recent event for that schedule.

For example, consider a case where an *i.LON* 100 server is executing a weekday schedule with the following events:

06:00	Warmup
08:00	Daytime
18:00	Cooldown
19:00	Nighttime

At 04:30 there was a power outage, and power was restored at 09:00. Since the most recent event was the 08:00 Daytime event, the *i.LON* 100 server would set its output data points to the values specified by the Daytime preset.

Any events that should have happened in the ensuing time (in this case, the Warmup event) will not be processed.

5

Logging Data

This chapter describes how to use the *i.LON 100* Internet Server to save network data into data logs, and how to view the data logs.

Data Logger Overview

The *i*.LON 100 Internet Server can contain up to 10 **Data Loggers**. You can use a Data Logger to record updates to a group of user-specified data points into a *log file*. The log file for each Data Logger is stored in the `/root/data` directory of the *i*.LON 100 server with the file name `logX`, where X represents the index number assigned to the Data Logger.

You can create two kinds of Data Loggers: *historical Data Loggers*, and *circular Data Loggers*. A historical Data Logger stops recording data point updates when its log file becomes full. A circular Data Logger removes the records for older updates when its log file is full, and new updates occur. A Data Logger can save either type of log file in an ASCII-text (.csv file extension) or binary (.dat file extension) format.

The *e3* version of the *i*.LON 100 also features the ability to save the log files as text files in compressed format (.gz file extension) when using the *i*.LON 100 Web pages. A log file that uses the new compressed format can store many more log entries than a log file using the other formats, while conserving the flash memory on the *i*.LON 100. All you need to do is extract the .csv file from the .gz file to view the log.

The Data Logger can record data in two ways: event driven updates and polling. Each data point can have its own set of parameters that determine if the data will be recorded at a specified poll rate (a *poll driven update*) and/or when an update of the data point occurs (an *event driven update*). For event driven updates, you can set the minimum amount of time between updates and the amount of change in the value with the **Min Delta Time** and **Min Delta Change** properties. These filters do not affect data that is recorded at a specified poll rate. When a poll rate is set for a data point in a Data Logger, a value will always be written to the log at that rate.

You can also define a threshold level for each Data Logger. The threshold level represents a percentage. When the Data Logger's log file consumes this percentage of the memory space allocated to it, the Data Logger provides notification that it is time to upload the log and clear out some of the data. The Data Logger makes this notification by updating the Data Logger's alarm data point (called `NVL_nvoDIlevAlarm[X]`, where X represents the index number assigned to the Data Logger) to the status `AL_ALM_CONDITION`. This feature is useful when working with historical Data Loggers, which are disabled when they become full.

You can access the data in a log file by viewing it in the data logging Web page, by manually opening the log file, or by using a SOAP function. You can clear data from a log by sending an update to the data point `NVL_nviDIClear[X]`, where X represents the index number of the Data Logger to be affected, or by using a SOAP function.

Each log can be up to 1024 KB, and the *i*.LON 100 server can log up to 10 MB of data. You can view the amount of free disk space with the **Setup - System Info** Web page.

To create a Data Logger with the Configuration Plug-in, follow these steps:

1. Open an existing LNS network or create a new network. See the *LonMaker User's Guide* for information on how to do this using the LonMaker tool.
2. Use an existing *i*.LON 100 device in the network, or create and commission a new *i*.LON 100 device.

3. Create a **Data Logger** functional block. You can do this with LonMaker by dragging a **Data Logger** shape from the *i.LON 100* Shapes stencil to the LonMaker drawing.
4. To log data using local data points, create one or more dynamic input network variables on the **Data Logger** functional block as described in Chapter 4 of the *i.LON 100 e3 User's Guide*.
5. Start the *i.LON 100* Configuration Plug-In. To do this with the LonMaker tool, right-click the Data Logger functional block and select **Configure** from the shortcut menu. The *i.LON 100* Configuration Plug-in opens with the **Data Logger** tab selected, as shown in Figure 6.1.

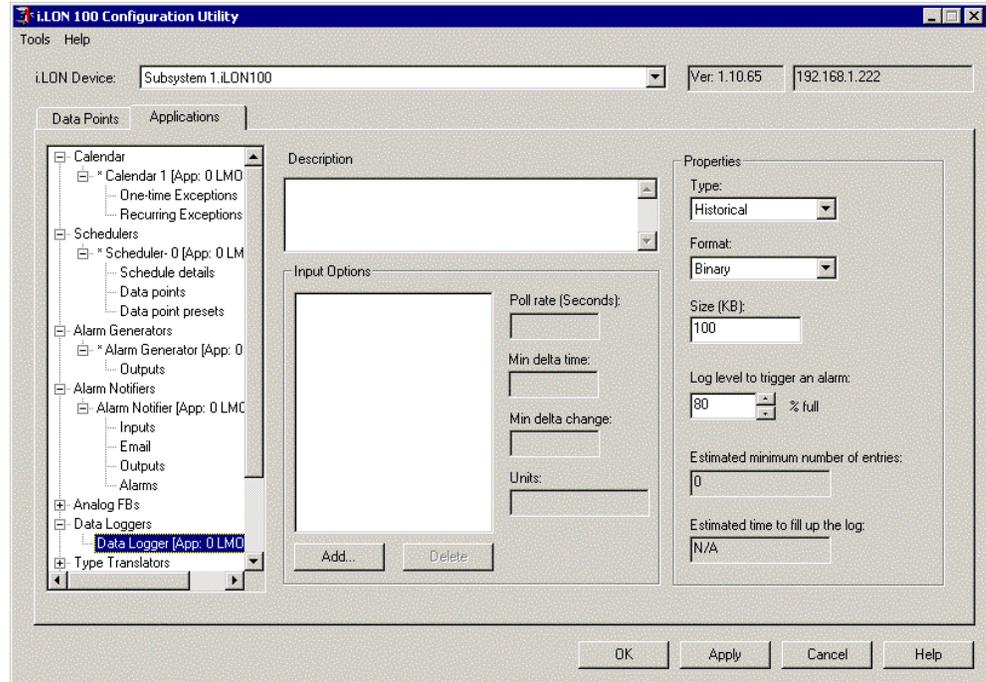


Figure 6.1 *i.LON 100* Configuration Utility Data Logger Dialog

6. Select the data points you want the Data Logger to log. The Input Options list shows all the data points that will be logged by this Data Logger. By default, this list includes all input network variables on the Data Logger functional block. To add additional data points, click **Add** and then select points from the **Add Data Point** dialog. To remove data points, select the point or points to be removed, and then click **Delete** (this will not delete the associated network variable).
7. Select a data point in the list and then configure the properties in the Input Options area to determine how data will be logged for the selected data point.

Table 6.1 Data Point Properties

Property	Description
Poll Rate	How often the value of the data will be logged, in seconds. If this value is set to 0, the value will only be logged when it changes in value.

Property	Description
Min Delta Change	Determines how much the data point's value needs to change between log entries. Set Min Delta Change to 0 to disable logging on value changes. For structure type data points, set the Min Delta Change to any non-zero value to cause any change in value to be logged.
Min Delta Time	Determines the minimum amount of time that must pass between log entries for the data point. The data point will not be logged again within this time, even if the Min Delta Change threshold is passed.
Units	Read-only. The units associated with the data point to be logged. This information is derived from the data point's type.

8. Configure the rest of the properties on the dialog shown in Figure 6.1. Table 6.2 describes these properties.

Table 6.2 Data Log Properties

Property	Description
Type	The type of the data log maintained by this Data Logger functional block. You can select from one of two types: <i>Historical</i> – When the data log fills up, data ceases to be logged. <i>Circular</i> – When the data log fills up, new data overwrites the oldest existing data.
Format	The format of the data logs. Available formats are Binary , Comma Separated Value and Compressed . Data logs can be found on the <i>i.LON 100</i> server in the root\data directory. The name of the log file is log<n>.dat (for binary log files) or log<n>.csv (for comma separated value log files), where <n> is the index number of the Data Logger functional block, or log<n>.csv.gz (for compressed comma separated value log files). Changing the log file format deletes any existing log file.
Size	The size of the data log. Each log can be up to 1024 KB, and the <i>i.LON 100</i> server can log up to 10 MB of data. You can view the amount of free disk space with the Setup - System Info Web page.
Log Level to Trigger an Alarm	When the data log reaches the specified percent capacity, the nvoDILevAlarm output network variable will be set to the AL_ALM_CONDITION status.
Estimated Minimum Number of Entries	The minimum number of entries it will take to fill the data log based on the log size and the size of the data points to be logged.
Estimated Time to Fill Up the Log	The estimated amount of time it will take to fill the log based on the log size, the size of the data points to be logged, and the poll rate.

9. Click **OK** to save your changes. The Data Logger will begin collecting data. You can extract the log file from the *i.LON 100* server and view it at any time, as described in Chapter 6 of the *i.LON 100 e3 User's Guide*.

Using a Data Logger with an Alarm Notifier

You can use an Alarm Notifier to automatically email a data log when it reaches a specified capacity and clear the data log. Typically, you will send the data log as a comma separated value (.csv file extension), which can be read by spreadsheet and database applications. When using a Data Logger with an Alarm Notifier, you should be aware that the **Data Logger** functional block includes the following input and output network variables:

nviDIClear	This SNVT_switch input network variable clears the data log when set to On (100.0 1). Once the data log is clear, reset this network variable to Off (0.0 0) to begin logging again. You can also clear a percentage of the total log by setting the Value field to a non-zero value (e.g. setting this network variable to 60.0 1 deletes all entries up to 60% of the total log — oldest entries first; if the log is less than 60% full, setting this value to 60.0 1 will delete the entire log).
nviDIEnable	This SNVT_switch input network variable enables and disables the Data Logger functional block. Set this network variable to Off (0.0 0) to disable data logging. You cannot clear a data log while logging is disabled.
nvoDILevAlarm	This SNVT_alarm output network variable sends an alarm when the data log reaches a configurable percentage of its capacity. You can configure the percentage as described later in this section.
nvoDIStatus	This SNVT_switch output network variable indicates the current status of the data log. The State field is set to 1 if the data log is active. The Value field indicates the level (how full) of the data log, from 0 to 100%

When using the Configuration Plug-In to create an Alarm Notifier to use with a Data Logger, follow these steps:

1. Create a **Data Logger** functional block and an **Alarm Notifier** functional block.
2. Connect the **nvoDILevAlarm** output network variable of the **Data Logger** functional block to the **nviAnAlarm** input network variable of the **Alarm Notifier** functional block.
3. Connect the **nvoAnAlarmFlag** output network variable of the **Alarm Notifier** functional block to the **nviDIClear** network variable of the **Data Logger** functional block. If you are using the LonMaker tool, your LonMaker drawing should now appear as shown in Figure 6.3

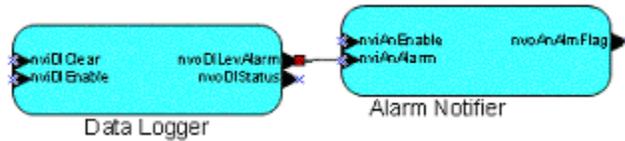


Figure 6.3 Data Logger Drawing

4. Right-click the **Data Logger** functional block and select **Configure** from the shortcut menu. The Data Logger configuration dialog opens.
5. Set **Log Level to Trigger an Alarm** to the percentage at which you want the data log to be emailed and cleared.
6. Set **Format** to **CSV Text**.
7. Expand **Alarm Notifiers** on the **Applications** tab, and then select the **Alarm Notifier** functional block. The **Alarm Notifier** configuration dialog opens.
8. Select the **Inputs** tab and verify that the **NVL_nviAnAlarm[n]** data point is listed as an input.
9. Select the **Email** tab. In the **To** box enter the email address to whom the log should be sent. In the **Attach** box, enter **/data/log<n>.csv**, where **<n>** is the index of the **Data Logger** functional block. Clear the **Send Email When Alarm Condition Clears** check box so that the **Alarm Notifier** does not send two emails with the attached data log.
10. Select the **Outputs** tab. Add the **NVL_nvoAnAlarmFlag[n]** data point to the **Output Data Points** list. Make sure that this is the only data point in the list. Set the **Active Value** to the same percentage as was set for the **Data Logger** in step 5, in the proper format for SNVT_switch. For example if you set the percentage to 75.0%, in the **Data Logger**, you should format this value in the **Alarm Notifier** as "75.0 1". Set the **Passive Value** to **Off (0.0 0)**. You can set these values directly or define them as presets, as described in Chapter 4 of the *i.LON 100 e3 User's Guide*.

When the **Data Logger** functional block reaches the specified percentage of its capacity, the **nvoDI LevAlarm** network variable will send an alarm condition. The **Alarm Notifier** functional block will attempt to email the data log to the specified address. Once the email has been sent successfully, the **Alarm Notifier** will set the **nviDIClear** network variable on the **Data Logger** functional block to the specified percentage, clearing the data log by the amount that was sent. If new data is received between the time that the **Data Logger** set its **nvoDI LevAlarm** network variable, the next attachment will include these duplicate entries.

6

Using Pulse Counter Inputs

This chapter describes how to use the *i*.LON 100 server's Pulse Counter to measure electrical energy, volume, rate or flow, or power.

Pulse Counter Overview

The *i.LON 100* server includes two pulse counter inputs. These can be used to connect the *i.LON 100* server to a any device with a pulse output that does not include a LONWORKS interface. Generally, the hardware input is from some meter that measures energy or the flow of a liquid or gas, and generates a pulse for some predefined unit like kilowatt-hours, liters, or gallons. These pulses are counted and the differential is calculated by the *i.LON 100* server and stored in output data points. You could link these data points to an Alarm Notifier or a Data Logger to monitor these hardware inputs with the *i.LON 100* server.

The *i.LON 100* server contains 2 Pulse Counter application instances – one for each pulse output device. If you want to change the type of data that is sent by the Pulse Counter to match your application, you must use the *i.LON 100* Configuration Plug-in to change the output data point types. These types cannot be changed using the Pulse Counter Web pages.

The Pulse Counter functional blocks include the input and output network variables shown in Figure 7.1.

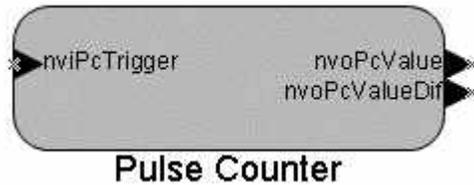


Figure 7.1 Pulse Counter Inputs and Outputs

Table 7.1 describes these network variables.

Table 7.1 Pulse Counter Inputs and Outputs

Name	Description
nvoPcValue	This changeable type output network variable sends the cumulative pulse count data (<i>i.e.</i> the total number of pulse counts received). This value will be calculated every time a pulse is received, but the value will not be sent to the network until the Time Interval (see below) expires or the state field of the nviPcTrigger network variable changes from Off (0) to On (1); polling this network variable at any time will return the most current value. The type of this network variable is set by the <i>i.LON</i> Configuration Software as described below.
nvoPcValueDif	This changeable type output network variable sends the pulse rate. The type of this network variable is set by the <i>i.LON</i> Configuration Software as described below.
nviPcTrigger	This SNVT_switch input network variable is used to synchronize the nvoPcValueDif network variable. When the state field of this network variable changes from Off (0) to On (1), the nvoPcValue and nvoPcValueDif values will be calculated and sent to the network and a new time interval is started.

Using the Pulse Counter

To use the pulse counter functional blocks with the Configuration Plug-In, follow these steps:

1. Open an existing LonMaker network or create a new LonMaker network as described in the *LonMaker User's Guide*.
2. Use an existing *i.LON 100* device in the LonMaker network, or create and commission a new *i.LON 100* device.
3. Drag a Pulse Counter functional block shape from the *i.LON 100* stencil to the LonMaker drawing. The LonMaker New Functional Block wizard opens.
4. Select **Pulse Counter 1** or **Pulse Counter 2** in **Functional Block – Name**. These selections correspond to hardware Meter Inputs 1 and 2 on the *i.LON 100* server. Select the appropriate functional block.

Click **OK**. The Pulse Counter functional block shape appears.

5. Right-click the new Pulse Counter functional block, and then select **Configure** from the shortcut menu. The *i.LON 100* Configuration Plug-in opens and prompts you for the IP address of the *i.LON 100* server.
6. Enter the IP address of the *i.LON 100* server and then click **OK**. The Pulse Counter tab opens, as shown in Figure 7.2.

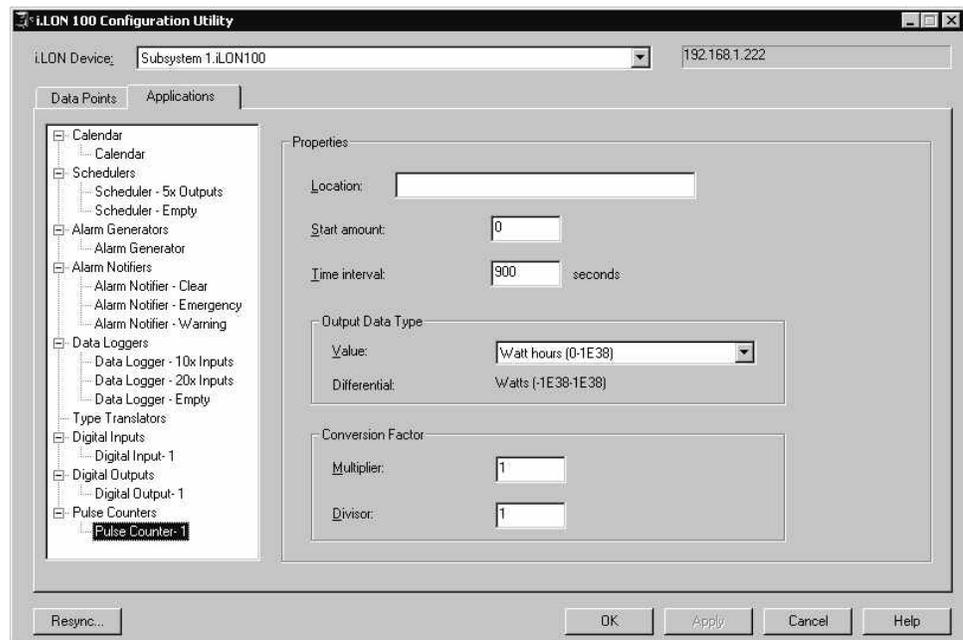


Figure 7.2 Pulse Counter Tab

7. Configure the properties on the Pulse Counter tab. Table 7.2 describes these properties.

Table 7.2 Pulse Counter Tab Properties

Name	Description
Location	A description of the location of the physical input associated with this Pulse Counter functional block. The location can be up to 30 characters. This information will not affect the operation of the functional block.
Start Amount	Specifies the start amount of the Value network variable. This value will be set as the actual value.
Time Interval	Specifies the time interval, in seconds, that will be used to calculate the pulse rate sent on the Differential network variable.
Output Data Type	<p>Select the type of the output data point from the drop-down list. This selection updates the types of the nvoPcValue and nvoPcValueDif network variables. Select one of the following options:</p> <ul style="list-style-type: none"> • <i>Watt hours (0-6553.5)</i> – The Value network variable will be set to a type of SNVT_elec_whr and send total watt-hours as an integer. The Differential network variable will be set to a type of SNVT_power and send power information as an integer. • <i>Watt hours (0-1E38)</i> – The Value network variable will be set to a type of SNVT_elec_whr_f and send total watt-hours as a floating point number. The Differential network variable will be set to a type of SNVT_power_f and send wattage information as a floating point number. • <i>Kilowatt hours (0-65535)</i> – The Value network variable will be set to a type of SNVT_elec_kwhr and send total kilowatt-hours as an integer. The Differential network variable will be set to a type of SNVT_power_kilo and send power information as an integer. • <i>Liters (0-6553.5)</i> – The Value network variable will be set to a type of SNVT_vol and send total volume as an integer. The Differential network variable will be set to a type of SNVT_flow and send flow rate information as an integer. • <i>Liters (0-1E38)</i> – The Value network variable will be set to a type of SNVT_vol_f and send total volume as a floating point number. The Differential network variable will be set to a type of SNVT_flow_f and send flow rate information as a floating point number. <p>When the output network variable reaches the limit defined in parentheses, it will wrap back to zero.</p>
Multiplier Divisor	Defines an input multiplier and divisor to convert the units per pulse of the attached meter to correspond to the output data point type. For example, if the output data type is <i>Watt hours (0-6553.5)</i> and the attached meter sends 1 pulse per 10 Watt Hours, you would set the Multiplier to 10 and the Divisor to 1. This setting would produce a 10 Watt-hour increase in the value on the nvoPcValue network value for every pulse of the electric meter.

8. Click **OK** or **Apply** to save the settings. The *i.LON 100* server will retain the values of the pulse counters when reset, but it will not count any pulses during a reset (a period of up to 2 minutes).

7

Using Digital Inputs/Digital Outputs

This chapter describes how to use the *i*.LON 100 Internet Server's two digital inputs and two digital outputs.

Overview

The *i.LON 100* device includes two digital inputs. These can be used to connect the *i.LON 100* device to a digital device that does not include a LONWORKS interface, such as a switch or pushbutton. The *i.LON 100 Internet Server Hardware Guide* describes how to connect the digital inputs to digital devices.

The *i.LON 100* device includes two dry-contact relay outputs. These can be used to connect the *i.LON 100* device to a digital device that does not include a LONWORKS interface such as a drive contactor or alarm bell. The *i.LON 100* device contains two Digital Output application instances.

This chapter describes how to configure the Digital Input and Digital Output functional blocks with the Configuration Plug-In. Before using the functional blocks, you should understand the network variables each one contains.

The Digital Input functional blocks include the input and output network variables shown in Figure 8.1.

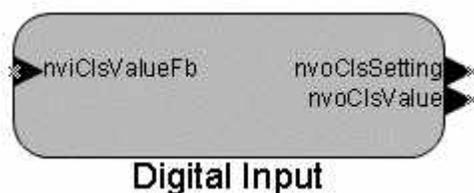


Figure 8.1 Digital Input Network Variables

The network variables shown in Figure 8.1 are described below:

- **nvoClsValue** is a **SNVT_switch** output network variable that sends the state of the digital input (*i.e.* **100.0 1** for On, **0.0 0** for Off). This value is derived from the raw digital input value according to the settings of the Configuration Plug-In, as described later in this section.
- **nvoClsSetting** is a **SNVT_setting** output network variable that sends the state of the digital input (*i.e.* **ST_ON 0 0.0** for On and **ST_OFF 0 0.0** for Off; only the **function** field is used). It is provided for compatibility with systems that use **SNVT_setting**, such as occupancy controllers and constant light controllers.
- **nviClsValueFb** is a **SNVT_switch** input network variable that is used to synchronize a group of switches. See the *LONMARK Closed-Loop Sensor Functional Profile* for more information. This network variable only works when bound.

The Digital Output functional blocks include the input and output network variables shown in Figure 8.2.

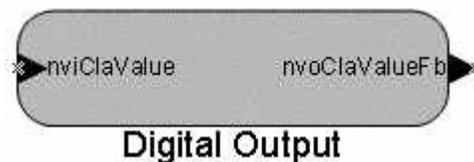


Figure 8.2 Digital Output Network Variables

The network variables shown in Figure 8.2 are described below:

- **nviClaValue** is a **SNVT_switch** input network variable that accepts an On (**100.0 1**) or Off (**0.0 0**) value. This value drives the digital output's hardware relay according to the settings of the Configuration Plug-in, as described later in this section.
- **nviClaValueFb** is a SNVT_switch output network variable that reflects the last value sent to the **nviClsValue** network variable. This is used to synchronize a group of outputs. See the *LONMARK Closed-Loop Sensor Functional Profile* for more information.

Using a Digital Input

To use a digital input with the Configuration Plug-In, follow these steps:

1. Choose one of the two digital inputs. Connect one side of the input to the 12V output and the other side to ground that runs through a switch or dry-contact relay. See the *i.LON 100 Internet Server Hardware Guide* for more information on the digital input. If the contact is open, the digital input will register OFF and if the contact is closed, the digital input will register ON.
2. Open an existing LonMaker network or create a new LonMaker network as described in the *LonMaker User's Guide*.
3. Use an existing *i.LON 100* device in the LonMaker network, or create and commission a new *i.LON 100* device as.
4. Drag a Digital Input functional block shape from the *i.LON 100* stencil to the LonMaker drawing. The LonMaker New Functional Block wizard appears.
5. Select **Digital Input 1** or **Digital Input 2** in **Functional Block – Name**. These correspond to hardware digital inputs 1 and 2. Select the appropriate functional block. Click **OK**. The Digital Input functional block shape appears.
6. Right-click the new Digital Input functional block and then select **Configure** from the shortcut menu. The Configuration Plug-in opens and prompts you for the IP address of the *i.LON 100* server.
7. Enter the IP address of the *i.LON 100* server and then click **OK**. The Digital Input tab opens, as shown in Figure 8.3.

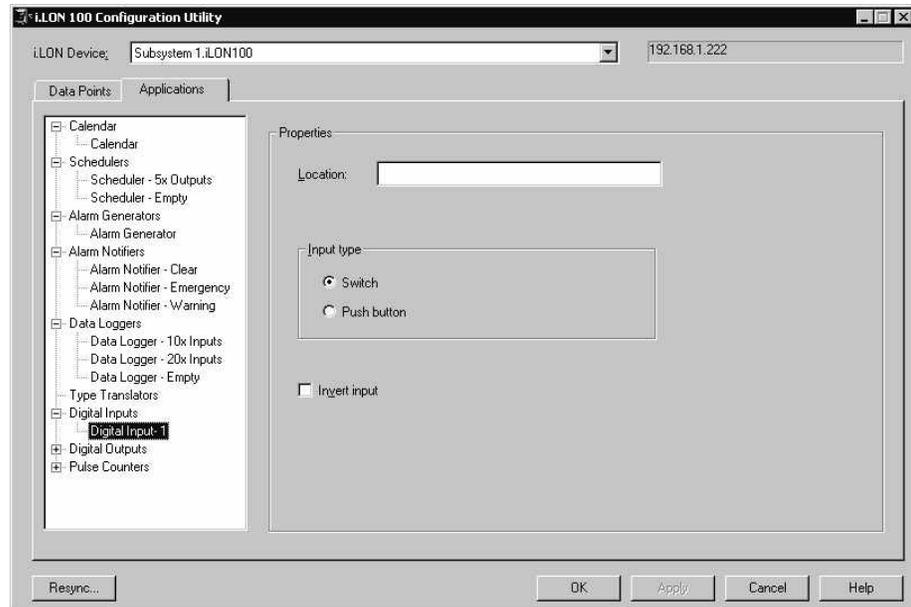


Figure 8.3 Digital Input Tab

8. Configure the properties on the Digital Input tab and then click **OK** to save your changes. Table 8.1 describes these properties.

Table 8.1 Digital Input Properties

Property	Description
Location	A description of the location of the physical input associated with this Digital Input functional block. The location can be up to 30 characters. This information will not affect the operation of the functional block.
Input Type	Select Switch or Push Button . If you select Switch , the nvoCIsValue network variable reflects the state of the hardware input (<i>i.e.</i> when the digital input is On , the nvoCIsValue network variable sends 100.0 1). If you select Push Button , the nvoCIsValue network variable will change state every time the hardware input transitions from a low to high value.
Invert Input	Set this check box to invert the input from the Value network variable. This is typically only used in conjunction with the Switch Input Type .

Using a Digital Output

To use a digital output, follow these steps:

1. Connect the one of the two relay outputs to a digital input.
2. Open an existing LonMaker network or create a new LonMaker network as described in the *LonMaker User's Guide*.
3. Use an existing *i.LON 100* device in the LonMaker network, or create and commission a new *i.LON 100* device.

4. Drag a Digital Output functional block shape from the *i.LON 100* stencil to the LonMaker drawing. The LonMaker New Functional Block wizard appears.
5. Select **Digital Output 1** or **Digital Output 2** in **Functional Block – Name**. These correspond to relay outputs 1 and 2. Select the appropriate functional block.
6. Click **OK**. The Digital Output functional block shape appears.
7. Right-click the new Digital Output functional block and select **Configure** from the shortcut menu. The *i.LON 100* Configuration Plug-in opens and prompts you for the IP address of the *i.LON 100* server.
8. Enter the IP address of the *i.LON 100* server and then click **OK**. The Digital Output tab opens, as shown in Figure 8.4.

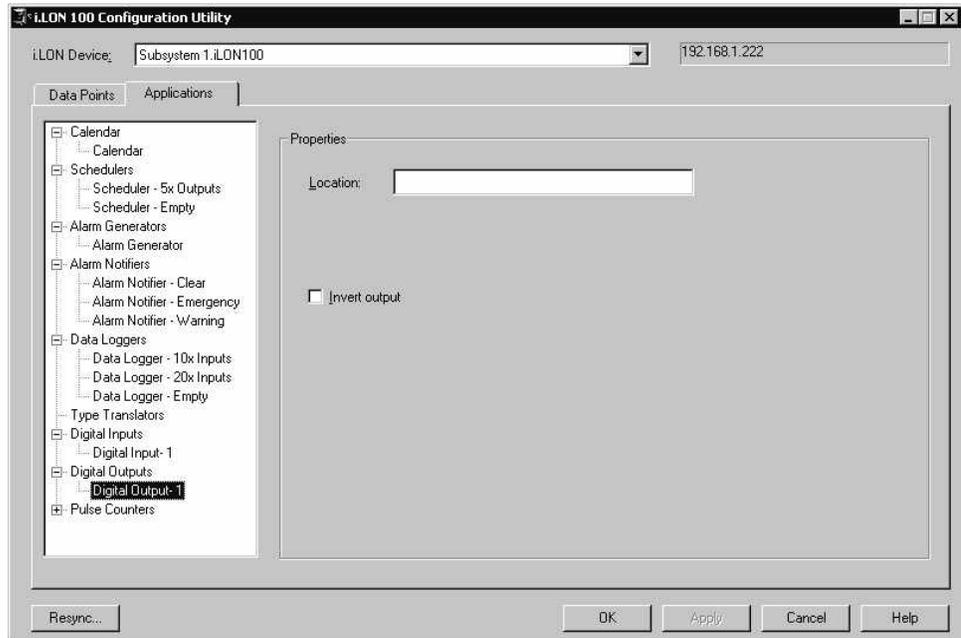


Figure 8.4 Digital Outputs Tab

9. Configure the properties on the Digital Input tab and then click **OK** to save your changes. Table 9.2 describes these properties.

Table 9.2 Digital Input Properties

Property	Description
Location	A description of the location of the physical output associated with this Digital Output functional block. The location can be up to 30 characters. This information will not affect the operation of the functional block.

Property	Description
Invert Output	<p>Select this check box to invert the output sent to the hardware. If this check box is cleared, the Digital Output functional block will close the relay when nviClValue is set to On (100.0 1), and will open the relay when the nviClValue is set to Off (0.0 0). If this check box is set, the Digital Output functional block will open the relay when the nviClValue network variable is set to On, and close the relay when nviClValue is set to Off.</p>

Appendix A

Troubleshooting

This appendix describes how to diagnose and resolve common problems that may occur when using the *i.LON 100* server.

Troubleshooting

I defined a dynamic network variable in the LonMaker tool, but I can't reference it in my Web page. What is wrong?

- The *i.LON 100* will append a number to any network variable you defined in LonMaker. This allows network variable names to remain unique even if two functional blocks have an input named `nviMyInput`. You can see what number is appended by looking at the data server tab on the *i.LON 100* plug-in.

The *i.LON 100* exhibits problems due to a low-memory condition. This could be indicated by one or more of the following: “out-of-memory” messages, slow network access, application performance problems, or even an unexpected reboot.

- Reduce the number of Alarm Notifiers and/or the size limit of the alarm Summary Logs (which are kept in RAM).
- Reduce the number of Web clients that are simultaneously accessing the *i.LON 100*.
- If you are making calls to the SOAP interface on the *i.LON 100*, refer to the *i.LON 100 Internet Server Programmer's Guide* for specific recommendations on limiting SOAP messages.

I have created NVE style points and although I can see them in the plug-in and reference them using the SOAP/XML interface I can not reference them using Web tags.

- *i.LON 100* Web tags do not allow NVE tag names with spaces. Be sure your tag names do not have spaces. Note that the SOAP/XML server and the plug-in do allow spaces under some conditions, but the best practice is never to include spaces in NVE names.

When I run the *i.LON 100* Configuration Plug-in I get an error stating that I could not communicate with the *i.LON 100*

- Verify that the WSDL file is not password protected. Protection is defined in the `webparams.dat` file located at the root of the *i.LON 100*'s flash disk.
- Verify that the IP communication path to the *i.LON 100* is intact. (Try pinging the *i.LON 100*).

I can't make my *i.LON 100* send an email message. What am I doing wrong?

- Emails are sent as a result of alarm conditions. Verify that the alarm condition actually triggers and that the email **should** be sent.

After changing the type of a data point using Configuration Plug-in or the LonMaker browser, the data point starts showing incorrect data.

- The LNS Data Server may be using old format data. Close and restart all LNS applications.

Anytime I reboot or power down and up when using the RNI I lose connection in LonMaker and have to un-attach and reattach.

- This is a known bug that will be addressed in a future release. The current work around is to close and then re-open the system.

When updating the *i.LON 100*, I get an error that the update completed successfully, but the device has not been updated successfully yet. What is wrong?

- This error occurs when you attempt to update the *i.LON 100* device while monitoring many of its network variables with a very short poll time. Try turning off monitoring or increasing the poll time.

I can't open the network using a non-*i.LON 100* network interface. What is wrong?

- If you were using the *i.LON 100* as an RNI, then you switched to another network interface, you need to disable RNI on the *i.LON 100* in order to open the network. This can be done using the console application or the Security Web page.

The *i.LON 100* gets a duplicate IP address assigned from a DHCP server, and either boots using its default IP address (*i.e.* 192.168.1.222), or boots using the duplicate IP address.

- If the *i.LON 100* device boots using its default IP address when it has obtained an address via DHCP, it is probably because the DHCP server assigned an address already in use by another machine on the network. This may happen if the other machine is disconnected from the network long enough for its DHCP lease to expire, yet continues using that address after reconnecting to the network. Make sure all other machines on the network using DHCP have valid leases. If the *i.LON 100* detects this condition, it will boot using its default IP address (typically 192.168.1.222), and will continue attempting to contact a DHCP server. Once it can contact a DHCP server, the *i.LON 100* will reboot to obtain a new IP address.
- The *i.LON 100* may not initially detect the above condition, in which case it will boot using the duplicate IP address. At some point the *i.LON 100* may discover that the address is a duplicate, but will continue using it. Make sure all machines on the network using DHCP have valid leases. Reboot the *i.LON 100* to obtain another address, if necessary, or use a statically configured address.



www.echelon.com