Lookout™

Lookout Developer's Manual

Worldwide Technical Support and Product Information

`ni.com`

National Instruments Corporate Headquarters

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

Worldwide Offices

Australia 02 612 9672 8846, Austria 43 0 662 45 79 90 0, Belgium 32 0 2 757 00 20, Brazil 55 11 3262 3599,
Canada (Calgary) 403 274 9391, Canada (Montreal) 514 288 5722, Canada (Ottawa) 613 233 5949,
Canada (Québec) 514 694 8521, Canada (Toronto) 905 785 0085, Canada (Vancouver) 514 685 7530,
China 86 21 6555 7838, Czech Republic 420 2 2423 5774, Denmark 45 45 76 26 00,
Finland 385 0 9 725 725 11, France 33 0 1 48 14 24 24, Germany 49 0 89 741 31 30, Greece 30 2 10 42 96 427,
India 91 80 51190000, Israel 972 0 3 6393737, Italy 39 02 413091, Japan 81 3 5472 2970,
Korea 82 02 3451 3400, Malaysia 603 9131 0918, Mexico 001 800 010 0793, Netherlands 31 0 348 433 466,
New Zealand 64 09 914 0488, Norway 47 0 32 27 73 00, Poland 48 0 22 3390 150, Portugal 351 210 311 210,
Russia 7 095 238 7139, Singapore 65 6226 5886, Slovenia 386 3 425 4200, South Africa 27 0 11 805 8197,
Spain 34 91 640 0085, Sweden 46 0 8 587 895 00, Switzerland 41 56 200 51 51, Taiwan 886 2 2528 7227,
Thailand 662 992 7519, United Kingdom 44 0 1635 523545

For further support information, refer to the *Technical Support and Professional Services* appendix. To comment
on the documentation, send email to `techpubs@ni.com`.

Important Information

Warranty

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

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

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

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

Copyright

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

Trademarks

Citadel™, ComponentWorks™, FieldPoint™, Lookout™, National Instruments™, NI™, ni.com™, and NI Developer Zone™ are trademarks of National Instruments Corporation.

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

Patents

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

WARNING REGARDING USE OF NATIONAL INSTRUMENTS PRODUCTS

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

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

Conventions

The following conventions appear in this manual:

»	The » symbol leads you through nested menu items and dialog box options to a final action. The sequence **File»Page Setup»Options** directs you to pull down the **File** menu, select the **Page Setup** item, and select **Options** from the last dialog box.
	This icon denotes a tip, which alerts you to advisory information.
	This icon denotes a note, which alerts you to important information.
	This icon denotes a caution, which advises you of precautions to take to avoid injury, data loss, or a system crash.
bold	Bold text denotes items that you must select or click in the software, such as menu items and dialog box options. Bold text also denotes parameter names.
italic	Italic text denotes variables, emphasis, a cross reference, or an introduction to a key concept. This font also denotes text that is a placeholder for a word or value that you must supply.
`monospace`	Text in this font denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames and extensions, and code excerpts.
`monospace bold`	Bold text in this font denotes the messages and responses that the computer automatically prints to the screen. This font also emphasizes lines of code that are different from the other examples.
`monospace italic`	Italic text in this font denotes text that is a placeholder for a word or value that you must supply.

Contents

Chapter 1
Introduction to Lookout

Chapter 2
Developing an Application

Chapter 3
Tutorial

Chapter 4
Serial COM Port Communication

Chapter 5
Creating Lookout Processes

Chapter 6
Graphics

Chapter 7
Expressions

Chapter 8
Logging Data and Events

Chapter 9
Alarms and Events

Chapter 10
Security

Chapter 11
DDE

Chapter 12
Reports and Web Clients

Chapter 13
Networking and Deploying an Application

Chapter 14
Setting Up Redundancy

Appendix A
Numeric, Time, and Date Formats

Appendix B
Structured Query Language

Appendix C
CBL Compiler

Appendix D
Lookout INI File Settings

Appendix E
Networking with DDE

Appendix F
Technical Support and Professional Services

Glossary

Index

1

Introduction to Lookout

This chapter explains how to install and run Lookout, the basics of the Lookout environment, and the architecture of Lookout applications.

Lookout is a Human-Machine Interface (HMI) and Supervisory Control and Data Acquisition (SCADA) software package for industrial automation. Lookout runs under Windows and communicates with field I/O through control hardware. Typical Lookout applications include continuous process monitoring and supervisory control, discrete manufacturing, batch applications, and remote telemetry systems.

Hardware and Software Requirements

Lookout requires the following:

- Pentium class or equivalent PC running at 90 MHz or faster

- At least 32 MB RAM

- 65 MB free disk space, plus possibly 100 MB or more, depending on how much historical data you intend to log to the Citadel database; you should also have about 50 MB of disk space for file swapping on Windows 2000/NT/XP computers

- Windows 2000, Windows NT version 4 or later, Windows XP, or Windows Me/98; National Instruments strongly recommends using Windows 2000/NT/XP for optimum stability

- If you plan to install the Historical Data Viewer along with Lookout, and you are running Windows NT, you must have service pack 6 or later installed

- Monitor display setting of at least 800×600 pixels

- Network card and TCP/IP networking installed on the computers you want to connect, if you intend to take advantage of Lookout networking

Note Performance of Lookout depends on the number, size, and complexity of processes you are running. While the minimum requirements are fine for some processes, you should plan on using more powerful computer configurations for complex and data intensive applications.

- You must be properly set up on a network with TCP/IP protocols installed if you want to use the networking capabilities of Lookout. You must be able to ping any computer on your network that you intend to have as a part of your Lookout network. To test this, access a command prompt and enter the following command:

  ```
  Ping compname
  ```

  where *compname* is the name of the computer you want to ping.

  If TCP/IP is functioning properly on your computer, you should receive a response that looks something like the following example:

  Pinging compname [123.45.67.89] with 32 bytes of data:

  Reply from 123.45.67.89: bytes=32 time<10ms TTL=128

  Reply from 123.45.67.89: bytes=32 time<10ms TTL=128

  Reply from 123.45.67.89: bytes=32 time<10ms TTL=128

  Reply from 123.45.67.89: bytes=32 time<10ms TTL=128

  If TCP/IP is not working properly on your computer, consult your system administrator. Refer to Chapter 13, *Networking and Deploying an Application*, for more information on networking and network configuration with Lookout.

Tip If your computer does *not* have a command prompt available in the Windows **Start** menu or in **Start»Programs**, select **Start»Run** and enter command.com in the **Run** dialog box. This opens a command prompt you can use for the ping command.

Installing Lookout

This section describes how to install Lookout. For more information about upgrading from earlier versions of Lookout, refer to the release notes for this version of Lookout.

1. Shut down all applications that may currently be using ODBC such as spreadsheets, word processors, database programs, MS Query, and similar applications.

Note If you are installing over an existing copy of Lookout, you may want to save any .ini files in your Lookout folder to another location. Restoring these files after installation will preserve your configuration settings.

2. Insert the Lookout CD into your CD drive.

3. The Lookout CD-ROM has autorun capability. If for some reason the autorun fails to start the CD installation routine, select **Start»Run** and enter *N*:\Lookout\setup.exe where *N* represents your CD-ROM drive. (To install the free client version, enter *N*:\Free_Client\setup.exe. To install the Lookout Player, enter *N*:\LookoutPlayer\lk51_web.exe.) Then click **OK**.

   Choosing a custom installation provides you with certain installation options, including the following:

   • Lookout allows you to choose whether to install its help. Use **Browse** to enter the name of an alternate directory or click **Next** to accept the recommended directory name.

   • During installation, Lookout can install its ODBC driver. Make sure you select the **Citadel ODBC** option if you want to be able to access Lookout's Citadel database.

   • When installing Lookout on a Windows 2000/NT/XP system, you are presented with a list of options on what you may install. Make sure you install the optional **NT keyboard driver** under Windows NT if you want to be able to block unauthorized users from using certain keyboard commands such as <Alt-Tab> to switch out of Lookout.

4. Follow the remaining instructions to complete the Lookout installation.

Registering Lookout

If this is your first installation of Lookout on the computer you are using, or if you have any lost or corrupted your registration information, then the first time you launch Lookout, it prompts you for registration information. Be sure to register your Lookout Package to receive your permanent unlock code. As an unregistered package, Lookout is limited to 50 I/O points and one client connection, and runs for only 30 days.

When you register Lookout, you unlock it for permanent use at your appropriate I/O count. If you do not register Lookout by the end of the 30 day period, it lapses to a demo system. You must complete the registration form and send it to National Instruments in order to register Lookout. When we receive your registration form, National Instruments generates a keycode to unlock Lookout and faxes or mails it to you. You also can get your keycode through the Internet at ni.com/keycode. If you have not registered your copy of Lookout, you are guided through the

registration. If you have not yet created your personal profile, you must do so before you can register Lookout.

Lookout requires a hardware key in some countries. Contact National Instruments if you are not sure whether your system requires a hardware key. If you were supplied a key with Lookout, be sure to plug it into the parallel port on your computer before activating Lookout.

Starting Lookout for the First Time

1. Launch Lookout by selecting **Start»Programs»National Instruments Lookout**.

   A dialog box appears asking you to register Lookout. If you are ready to register Lookout, click **Enter new keycode**. The registration dialog box appears.

2. Enter your name in the **Name** field.

3. Enter the **Organization** name exactly as it appears with the keycode information sent in response to your registration, including punctuation marks. This text is used in combination with the keycode, so it must be exact.

   (If you have not yet received your keycode from National Instruments, you can enter your registration data later. Select **OK** and Lookout will inform you that you have not registered your package yet. Select **OK** again until Lookout launches.)

4. Enter the **Serial Number** of your package. (This can be found on your registration form.)

5. Enter your 12-character **Keycode**. The keycode is not case sensitive and you can leave the hyphens out if desired. Notice that there are no spaces near the hyphens.

6. Click **OK**.

    If you enter the proper information correctly, Lookout appears on your screen with no process running. If you are certain that you typed the information correctly and Lookout does not accept it, contact National Instruments for assistance.

    The first time you start a development version of Lookout, it opens with the Administrator user account logged in, without requiring any login. You should assign a password to the Administrator account before allowing others who should not have Administrator privileges to use your copy of Lookout. Refer to Chapter 10, *Security*, for more information about setting up accounts and passwords.

Changing I/O Count Registration

If you want to change the number of Lookout I/O points you are using, or make other changes in system capabilities or client connections, first contact National Instruments to obtain a new Lookout keycode. Then select **Options»System»Change Registration Info** and enter the new keycode.

Adding Client Connections

You are limited in the number of client connections you are allowed to maintain in Lookout. You can add or remove client licenses from any copy of Lookout. For example, if you have a copy of Lookout running a server process with two clients connected and need to increase the number of client connections to three, you need only get a license for one more client connection, and enter that new license code.

Note   If you are logged on to a Windows 2000/NT/XP computer as Guest or Restricted User, certain Lookout tasks will not work correctly, because they require writing to the registry or WINNT folder. These tasks include creating a user account, adding a client license, and registering a Logos computer.

1. To add or change client connection information in Lookout, select
    Options»System from the menu, and click the **Change Client
    License** button. Click the **Add** button. The following dialog box
    appears.
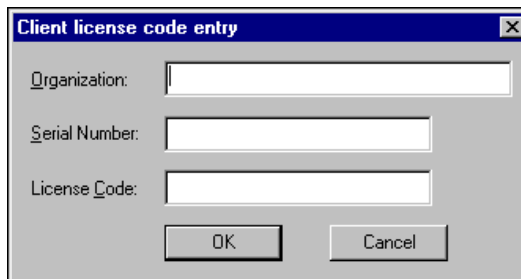
2. Enter your **Organization** name as you reported it in your registration
    request. Enter the **Serial Number** and the **License Code** you received.
    Click **OK**.

If your registration attempt fails, check to make sure you entered all of the
information correctly. Contact National Instruments if correcting these
entries does not fix the problem.

A newly installed version of Lookout will run with one client connection
for 30 days before reverting to demo program mode.

Lookout Features and Capabilities

Lookout offers the following features and capabilities.

*   Serial port communication—You can communicate with certain
    control hardware through the serial ports of your computer. Lookout
    arbitrates serial port usage. Refer to Chapter 4, *Serial COM Port
    Communication*, for more information on configuring
    communications.

*   Graphics—Lookout has an extensive library of standard graphics, as
    well as the Image Navigator and its library of graphics, that you can
    use as you build your user interface. You can also create your own
    custom graphics for use in Lookout. Refer to Chapter 6, *Graphics*,
    for more information on graphics.

*   Alarms—In Lookout you can generate, display, log, filter, group,
    and print alarms. Refer to Chapter 9, *Alarms and Events*, for more
    information.

- Multimedia—Lookout allows you to play sound wave files. Also, you can create color animations on your Lookout panels using the Animator, Multistate, and Pipe objects. Refer to these object topics in *Lookout Help* for more information about implementing these features.

- Security—Lookout provides a security system for both local and network security. You can configure network security, control security, viewing security, and action verification. You can specify the level of access and control for each operator or networked machine. Refer to Chapter 10, *Security*, for more information about security.

- Data and event logging—You can store real-time system information to disk in comma-delimited ASCII files, or you can log historical data to the Citadel database. Lookout logs operator commands, such as closing a process file or flipping a switch. Along with each event, Lookout logs the account name (operator), date and time of the event, name of the object adjusted, and the before and after settings of the object. Refer to Chapter 8, *Logging Data and Events*, for more information about logging.

- ODBC support—Because of the Lookout Open Database Connectivity (ODBC) support, you can use other applications, such as Microsoft Access, to query the Citadel historical data. Refer to Appendix B, *Structured Query Language*, for more information about ODBC.

- DDE support—Lookout can send its live process values to other applications, and it can receive real-time values from other applications. Lookout acts as both a DDE client and a DDE server. Refer to Chapter 11, *DDE*, for more information about DDE.

- Networking—Lookout provides high-speed client-server networking through the use of TCP/IP. You can monitor and control your processes from multiple workstations on a network. Refer to Chapter 13, *Networking and Deploying an Application*, for more information about networking Lookout computers.

- OPC—Lookout is a fully-compliant OPC client and server.

- Redundancy—You can configure two computers for redundancy, providing automatic transfer of monitoring and control if one of the computers fails. Refer to Chapter 14, *Setting Up Redundancy*, for more information about setting up redundancy.

- Web capabilities—With the Report object and Web Client, you can monitor and even control processes through Microsoft's Internet Explorer. Using the Lookout WebBrowser ActiveX control, you can access the Web from inside a Lookout process.

Lookout Environment

Figure 1-1 shows a Lookout screen in edit mode.

1	Menu Bar	5	Status Bar	9	Active Alarms
2	Root Selection Box	6	Object Location	10	Lookout Workspace
3	Lookout Object Explorer	7	Object Size	11	Selected Display Element
4	Alarms Window	8	Object Name	12	Process Control Panel
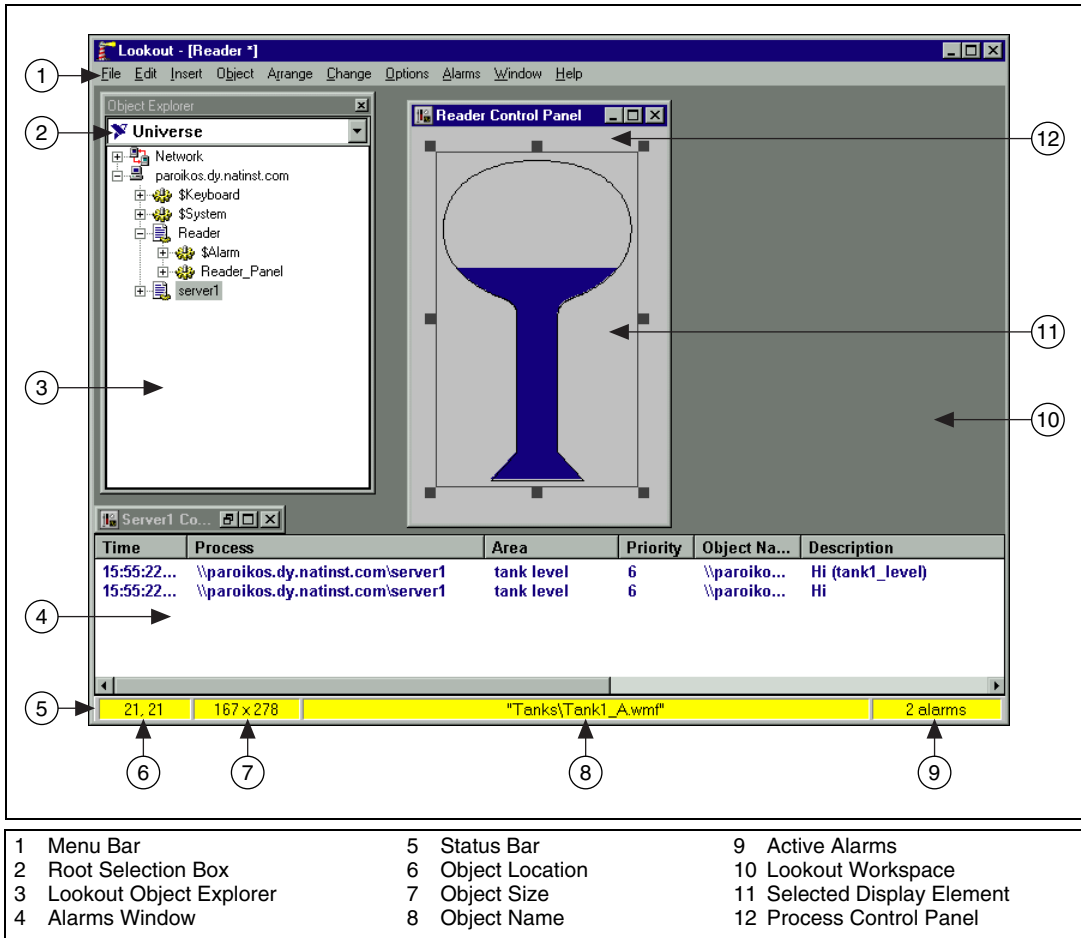
Figure 1-1. Lookout Environment

Lookout has both edit and run modes. You can toggle between them by pressing <Ctrl-Space> on your keyboard, or by selecting or deselecting **Edit Mode** in the **Edit** menu. Some menus and utilities are available in only one mode or the other. The status bar is yellow when Lookout is in edit mode.

Depending on the security access level of the user logged in to Lookout, certain menus and tools may be unavailable. Refer to the *Lookout Help* for more information about *Lookout Menu Commands*.

Menu Bar

The menu bar displays the currently available menu commands, which change depending on whether Lookout is in run mode, edit mode, or is open without any processes running. These commands may or may not be available to the operator, depending on his or her security level.

Status Bar

When Lookout is in run mode, the status bar is gray. The time and date are displayed on the left end of the bar. The account name of the currently logged on operator comes next. The company name as entered during registration appears in the middle, and the alarm status is on the right end of the status bar. The control panel navigation arrows, which scroll through various control panels, may or may not be visible, depending on your system options settings.

| 07/26 14:58 | Administrator | National Instruments Internal Use Only | ← | → | 0 alarms |

In edit mode the status bar is yellow. The X and Y coordinates of the currently selected item (if any) appear on the left end of the bar. The width and height of the currently selected item come next. The name, file name, or expression of the selected item appears in the middle, and the alarm status appears on the right end of the status bar.

| 192, 220 | 111 x 20 | \\TRIPPER\Watcher2\Waveform1.triangle_180 | ← | → | 0 alarms |
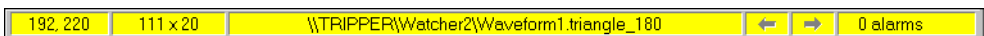
To toggle in and out of edit mode, select **Edit»Edit Mode**, or press <Ctrl-Space>.

Lookout Workspace

The workspace is the area in which you view and operate control panels. The visible workspace on your screen is only a window into the Lookout virtual workspace. If control panels or their associated icons are partially or completely outside the visible workspace, Lookout automatically displays horizontal and vertical scroll bars.

Object Explorer

The Lookout Object Explorer is a useful tool for exploring and editing your Lookout processes. You can access the Lookout editing functions from this window by right-clicking the objects displayed there. You can also create processes, objects, folders within processes, and then organize objects by moving them from one location to another within a process, create objects, drag displayable objects and display expressions to different panels, and more.

To use the Object Explorer, make sure Lookout has a process running and is in edit mode, then select **Object»Object Explorer**.

Right-clicking an object or process inside the Object Explorer displays some variation of the following menu.

```
New Object...
New Folder
New Symbolic Link...
Make Root
Refresh              Ctrl+R

Cut                  Ctrl+X
Copy                 Ctrl+C
Paste                Ctrl+V

Rename
Delete               Del

Edit Connections...
Edit Database...

Configure Security...
Properties...
```
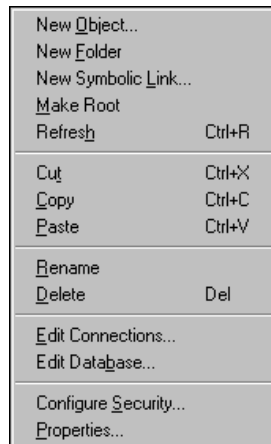
You can rename folders and relocate objects with the Lookout Object Explorer as well.

When you leave edit mode, the Lookout Object Explorer window closes.

Control Panels

Control panels provide the display area for any switches, knobs, bar graphs, digital displays, trend graphs, and other components that you want to use to monitor and control your operations.

You can move the panels around the screen by grabbing the title bar of a panel with the mouse cursor and dragging it to a new location. There is

no limit on the number of control panels you can create or the number of objects displayed on any one panel.

When in edit mode you can access many of the Lookout editing functions by right-clicking an object or expression in the panel, or on the panel itself.

There are three types of control panels: normal, pop-up, and pop-up with no icon. A normal control panel can be maximized, normal size, or minimized within the Lookout workspace. Control panels can also "pop up" when an event occurs, such as when a push button is pressed or when an alarm is activated. When a pop-up control panel is displayed, it remains on top of all other panels until you minimize it.

Alarms Window

Below the Lookout Object Explorer and the Reader Control Panel in Figure 1-1 is the Alarms Window. You can choose how the Alarms Window is displayed by selecting **Alarms»Display Options**. In Figure 1-1, the Alarms Window is set to display at the bottom of the workspace.

Connection Browser

Use the Connection Browser to register network computers, navigate quickly between objects, follow connections, monitor values, identify unused objects, and organize the objects in a process.

To access the Connection Browser, select **Object»Connection Browser**. To browse the connections in your Lookout processes, Lookout must be in edit mode. After opening the browser, double-click items in the Connection Browser to expand the view and see the existing connections.

When you select an individual object to show connections, your browser screen looks something like the following illustration.
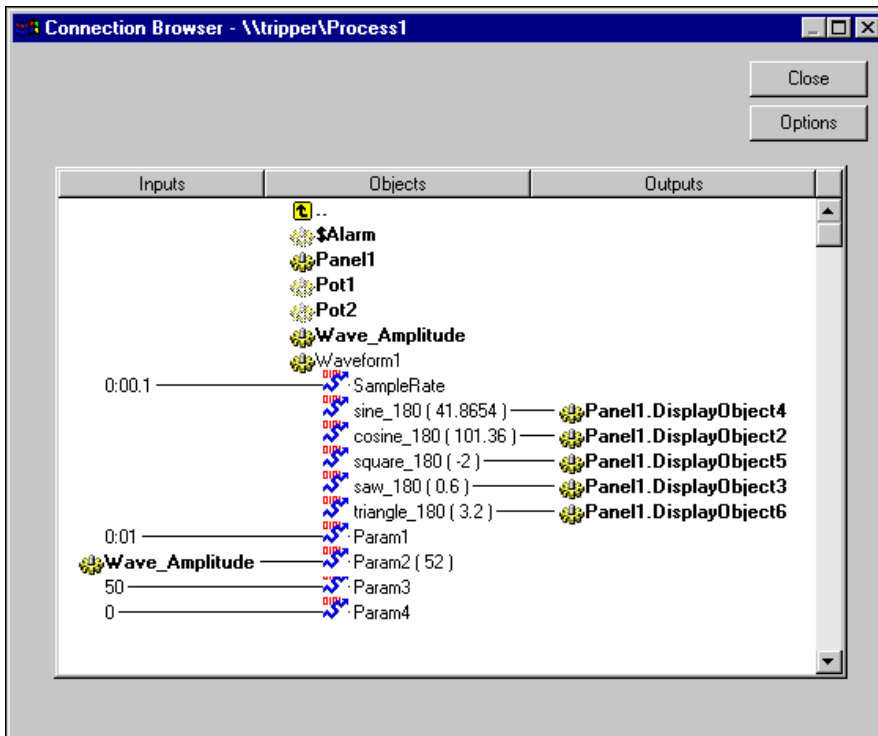
In this example, the Waveform object named `Waveform1` has been selected. All the various display sites and connections are shown, including the way this Waveform is connected to a Pot (Wave_Amplitude) controlling the amplitude. The values of the different waveforms are shown in parenthesis to the right of each output.

A value must be connected to or displayed to appear as live data in the Connection Browser. Using data as an item in a HyperTrend, however, is not a Lookout connection, so data displayed in this manner will not appear live on the Connection Browser.

The `Pot1` and `Pot2` object icons are faded in appearance, meaning that their outputs are not being used by any other object.

While you can browse from one process to another when both processes are open on one computer, you cannot browse across the network to a process running on another computer. The Connection Browser will show an object from another computer on the network on the screen, but cannot browse that object.

You can use the Connection Browser to move objects from one location into another inside a given process, using the typical Windows drag-and-drop technique. You can also create new objects or folders, and edit existing objects or connections. Right-click in the Connection Browser to get the menu to choose one of these operations. But carefully plan how you want to arrange the objects in each process, and be aware of how path relativity works in Lookout, to avoid connection problems. For more information about how a change in location can affect object connections, refer to the *Path Modes in Lookout* section of Chapter 5, *Creating Lookout Processes*.

Refer to the *Registering Network Computers* section of Chapter 13, *Networking and Deploying an Application*, for information on using the Connection Browser to register networked computers in Lookout.

Connection Browser Options

Click the **Options** button to activate the Connection Browser Options dialog box.

If you want to eliminate object parameters (in order to focus on data member connections) disable the **Show parameters** option.

Disabling the **List Panel displays as connections** removes all display components from the Connection Browser view of a control panel.

Disabling the **Show constant connections** removes all static values connected to an object parameter or data member. Disabling this option can help you focus on variables and connections to changing values originating in other objects.

Select **Save settings on exit** to keep your configured options for the Connection Browser the next time you open the browser in a Lookout session.

Selecting **Play sound effects** enables your Windows system alarm sound when you attempt an illegal operation in the Connection Browser.

Operator Input and Navigation

Operators may navigate using a mouse, trackball, touchscreen, or keyboard. When the cursor moves over a controllable object, the cursor turns into a hand, indicating you now have control of the object. Controllable objects include such things as Switches, Pots, and Pushbuttons.

When using a keyboard, the arrow keys move the cursor around the screen. The <Tab> key jumps the cursor from one controllable object to another, and the <Space> bar acts as the left mouse button, so you can click a controllable object without actually using a mouse.

You can also tie keyboard function keys to control panels and other objects, so you can switch between control panels or issue control commands just by pressing a function key. Refer to the *Lookout Help* for more information about *$Keyboard*.

Virtual Keypad

When you are in run mode, you can click a digital Pot control and bring up a virtual keypad to enter numeric values, either with a mouse or a touchscreen.

Virtual Keyboard

Lookout also has a virtual keyboard you can use with a touchscreen or a mouse.

To enable the virtual keyboard, select **Options»System** and then check **Left Mouse Click** or **Right Mouse Click** in the **Virtual Keyboard Pops Up** section of the dialog box.

When this feature is enabled, clicking in a data entry field or touching the field on a touchscreen displays the virtual keyboard, as shown in the following illustration.

Lookout Architecture

Lookout consists mostly of objects, connections, and services. Developing a Lookout application is a matter of creating, configuring, and connecting objects. Lookout and the Lookout services handle the connections between your computer and various PLCs or other controllers; between your computer and various sensors; between your computer and other computers; and between your computer and the Lookout database, Citadel.

Client and Server Processes

A Lookout process is a program that you create to perform some specific job. Lookout can run any number of processes at one time. You can open and close a process without disturbing other processes that are running at the time. Your processes can report and analyze data and control machinery, all while interacting with other Lookout processes running on your computer and others scattered across the network.

It is important to make a distinction between client and server processes as you design and develop your Lookout HMI. In general, server processes communicate with your control hardware; client processes connect to server processes and provide the user interface for operation of your system. Refer to the *Implementing a Server/Client Design* section of Chapter 2, *Developing an Application*, for more information about server and client processes.

Client and Server Versions of Lookout and Driver Objects

Lookout is available in two basic types of packages: server versions and client versions.

Use server versions of Lookout to develop Lookout processes and run those processes on computers with physical connections to field hardware. Server versions of Lookout come with various I/O and client connection licenses, either with development capabilities or as run-time only. You must use a server version of Lookout to create or edit processes that contain driver objects (used to communicate with control hardware).

The client development version of Lookout can be used to create and edit client processes. Client versions of Lookout cannot run processes that contain a Lookout driver object. The client run-time version of Lookout is distributed at no charge.

Event-Driven Processing

Lookout is entirely *event-driven*, not *loop-driven*. Loop-driven applications execute code sequentially from top to bottom, and loop back to the top to execute the same code over and over. Commands may execute millions of times before the result changes, which wastes computer processor time and slows down responses to frequent events. In contrast to this programming model, Lookout is entirely *event-driven*. Each object remains inactive until an event occurs.

When an incoming signal changes in Lookout, the object processes the value according to its functionality. Objects send out signals only when the result of their processing changes. This event-driven paradigm demands much less processor time than a loop-driven design. It closely simulates the responsiveness of physical switches, push buttons, and relays, and is the reason Lookout is so fast.

Lookout Objects

Lookout architecture is based on objects and connections between objects. Objects are software representations of such things as switches, timers, or PLCs. Each object encapsulates a specific functionality. Each object has a set of parameters, which basically define and configure the object, and a set of data members, which act as input and output points. Figure 1-2 shows the functionality, data members, and parameters of an object.
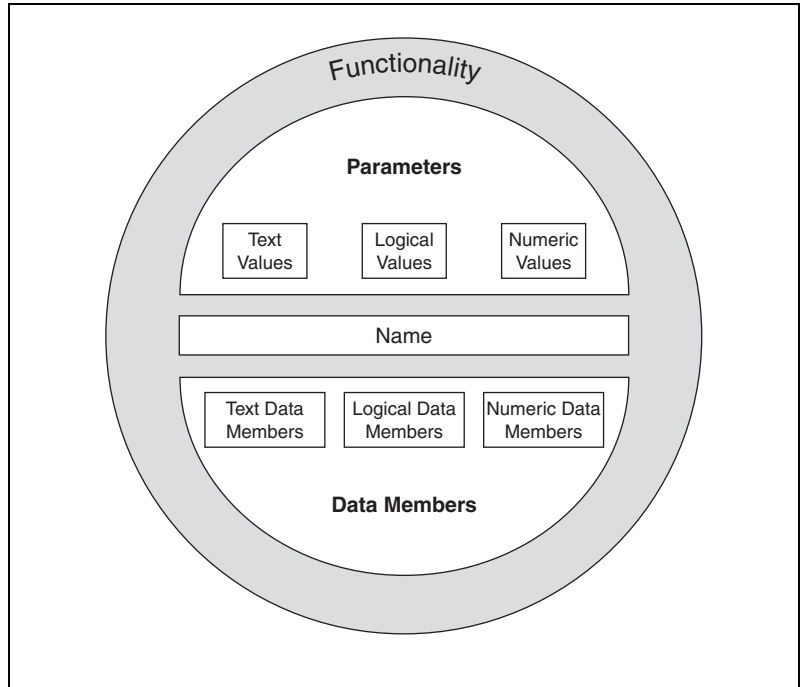
Figure 1-2. Objects Encapsulate Data Members, Parameters, and Functionality

Lookout object categories include calculation, control, display, drivers, logging, monitoring, multimedia, networking, statistical process control, and timers. Driver objects allow communication with specific control hardware models. Also, Lookout automatically creates three global objects: $Alarm, $Keyboard, and $System. Refer to the *Lookout Help* for more information about these objects.

ActiveX Objects

Another special type of object is the Lookout ActiveX object. You create an ActiveX object to use an ActiveX control in a Lookout process. Refer to the *ActiveX Objects* section of Chapter 5, *Creating Lookout Processes*, for more information about using ActiveX objects in Lookout.

Parameters

Parameters define a Lookout object's characteristics and functionality limits. Parameters appear in the dialog box when you create an object or modify its properties. For example, **Baud rate**, **Parity**, and **Stop bits** are a few of the parameters that might define how a driver object works.

Another example would be the **Control security level** of a Switch object. The *Lookout Help* topics for each object contain a list of object parameters.

Some parameters require constant or specific values. Many parameters can contain *expressions*, as indicated by a yellow field, which means that the parameters can change programmatically. Refer to Chapter 7, *Expressions*, for more detailed information on expressions. For added flexibility, some parameters double as data members.

Data Members

Object *data members* contain information about the current state of the object, such as the value, whether the control is visible or hidden, and so on. Driver object data members represent the I/O values of your control hardware. The Lookout online help for each object contains a list of that object's data members.

A driver object representing a PLC might have hundreds of data members, but a Switch object has less than 10 data members. Every data member is a single value that can be one of three types: numeric, logical, or text.

You can configure the data member to suit your specific needs, including properties such as alarms, logging, or scaling. You can even create one or multiple aliases for a data member, each with a different configuration.

Logical Data Members

Logical data members represent a binary or on/off state. Lookout recognizes the following logical constants:

- An on state: yes, true, on
- An off state: no, false, off

Lookout is not case sensitive in this instance.

Refer to the *Data Type Matching and Conversion* section of Chapter 5, *Creating Lookout Processes*, for information on how Lookout interprets numeric or text data when it is passed to a logical data member.

Numeric Data Members

A *numeric* data member is a floating point number representing analog values such as a tank level, voltage, or temperature. Numeric data members also represent time, either as a span of time or as a particular time of day, week, month, or year. Numeric signals range from -1.7×10^{275} to 1.7×10^{275}.

Refer to the *Data Type Matching and Conversion* section of Chapter 5, *Creating Lookout Processes*, for information on how Lookout interprets logical or text data when it is passed to a numeric data member.

Numeric constants are entered using decimal digits (0–9), the minus sign (–), the exponent symbol (E or e), and the time format separator (:). Some examples of numeric constants would be 0, –123.779999, 1.5E7, –3.7E-3, –.0036, or 23356636.234579.

For more information about numeric display formats, including time and date values, refer to Appendix A, *Numeric, Time, and Date Formats*.

Text Data Members

Text data members contain text character strings consisting of any displayable characters A–Z, a–z, or 0–9. You can use text signals to display alarm descriptions on the alarm panel, to display labels on a control panel, and in expressions. You can enter text as constants, or you can construct them with the many text functions that are available for use in expressions. Be sure to enclose text constants in double quotes (" ") when using them within expressions.

Some examples of text constants would be the following:

*   "Water Temperature:"
*   " "   (empty text string)
*   "Low level in 'Polymer 2' tank"
*   "gpm"
*   "Fire in the: ""Pink room""" (an expression containing a quote)

Refer to the *Data Type Matching and Conversion* section of Chapter 5, *Creating Lookout Processes*, for information on how Lookout interprets logical or numeric data that is passed to a text data member.

(implicit) Data Members

Many objects have an (implicit) data member. An (implicit) data member is read-only. It is either logical, numeric, or textual, depending on the type of object, and it follows the same rules that apply to all other data members. The (implicit) member represents what Lookout considers to be the most commonly used data member of that object type. In some cases, it is the only data member for an object. Existing (implicit) data members are defined in the object data member topics in the *Lookout Help*.

If you specify an object without specifying a data member, you are implying the (implicit) data member. For example, instead of typing `Pot1.numeric` where `Pot1` is the name and `numeric` is the current read-only value of the Pot, you could enter `Pot1` and Lookout treats that as the implicit read-only value of the Pot.

Because (implicit) data members are read-only, you cannot write to them or make a remote connection to them. For example, to specify the position of a Switch as a read/write value, you would use the Switch's `value` data member.

Data Quality

Lookout uses data quality attributes to keep track of any problems with your data. If there is a communications failure with a device, if the network connection to the source of the data is bad, if the data source is undefined, if the value is stale, or if some other problem arises, Lookout creates an alarm to report the condition. Lookout places a red X over the affected control or expression when there is a data quality problem. Lookout also provides functions to check for quality and report about quality problems.

Connections

You can connect objects in Lookout, allowing signals to pass between them—much the same way you would connect a light switch to a light bulb. You can do this by connecting data members to each other, or by connecting data members to parameters.

Figure 1-3 shows a data member to parameter connection, and a data member to data member connection.
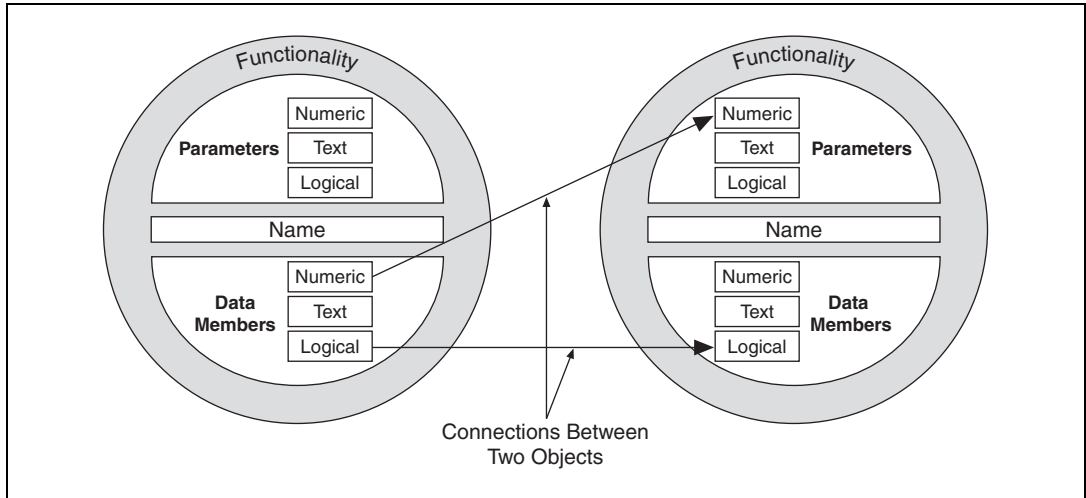
Figure 1-3. Example Connections between Two Objects

For example, you might make the numeric data member of a Pot object the source for the **Hi Level** parameter of an Alarm object. When you adjust the Pot, the Alarm **Hi Level** would change.

Your connections report on and react to events as they happen, not in some arbitrary order as in a loop-driven paradigm. Refer to the *Connecting Objects* section of Chapter 5, *Creating Lookout Processes*, for more information about making connections in Lookout.

Setting System Options

You can configure certain system options in Lookout that determine the way your version of Lookout operates. Select **Options»System** from the menu bar. The System Options dialog box appears as shown in the following figure. Some default settings may be different for your computer.

Note Only users logged on with security levels of 9 or greater can access the System Options dialog box.
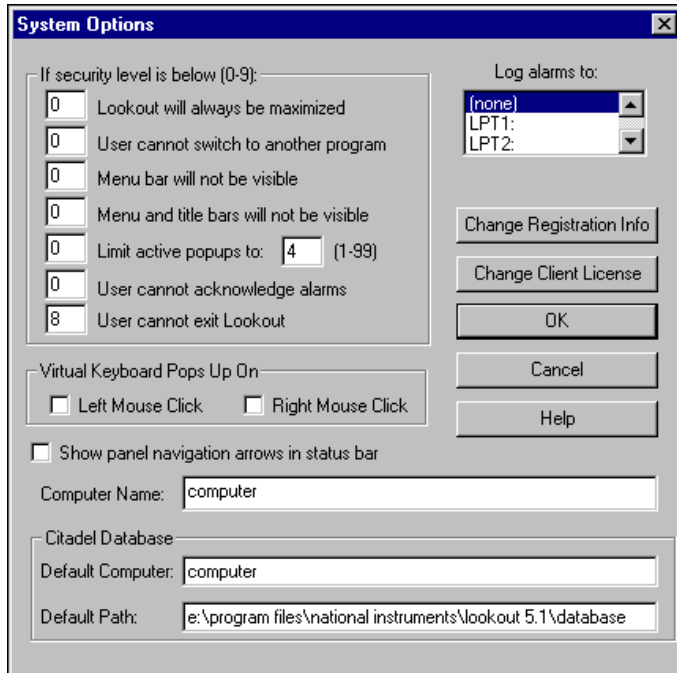
- **If security level is below (0-9)**—You can set local limits on how Lookout runs when users with different security levels are logged on. Use these selection boxes to set these limits. Refer to Chapter 10, *Security*, for full details on the security features of Lookout.

- **Log alarms to**—If you have a printer directly connected to your computer, you can direct that all alarms be printed when they occur. Set this field to the communications port to which your printer is connected. Refer to Chapter 9, *Alarms and Events*, for information on the alarm and event logging features of Lookout.

- **Virtual Keyboard Pops Up On**—Lookout features a virtual keyboard that you can access by clicking a control that accepts a text or numeric input. You can set whether a right-click or a left-click pops up this keyboard.

- **Show panel navigation arrows in status bar**—Select this checkbox if you want to use panel navigation arrows. These arrows, which appear in the right side of the status bar, activate control panels in the order in which you last accessed them. This feature is most convenient when you have a large number of control panels in a process and need to cycle through a subset of them several times in a short period of time.

- **Computer Name**—This field shows the network name of the computer you are working on. If this field is blank, and you intend to use the networking capability of Lookout, you need to check your network settings to make sure your computer is properly named for network operations.

- **Citadel Database**—These fields set the default destination for historical data logged by Lookout. All processes running under a single instance of Lookout will use this data path to log data to Citadel, unless you override this setting when you create the process.

  You set the **Default Path** and the **Default Computer** separately. Use the complete path for the **Default Path** setting, and the fully qualified computer name for the **Default Computer** setting.

- **Change Registration Info** and **Change Client License**—Click these buttons to enter new registration or license information.

2

Developing an Application

This chapter gives an overview of the steps and recommendations for developing an application in Lookout.

Example Applications

Before you develop a Lookout application, you might want to view the example processes that ship with Lookout. In Lookout, select **File»Open** and open any of the processes in the `Examples` folder. The *Lookout Examples* topic in the *Lookout Help* describe the examples and provide links to them.

Lookout Client/Server Application Design

A Lookout application usually consists of one or more server processes connected to one or more client processes. You can develop both on the same computer, but server and client processes have distinct roles.

Server processes implement the following:

- Connect directly to control hardware and devices.
- Log data, execute control logic, define alarms, implement redundancy, and implement network security.
- Not intended to be mobile.
- Can do anything a client process can do. Can include an operator interface with control security, usually for troubleshooting purposes.

Client processes implement the following:

- Connect to server processes, not directly to hardware.
- Provide an operator interface to display and control the condition of your system.
- Show server process alarms, log events, and implement control or network security.

- Should connect to server processes using a Symbolic Link so that they are mobile and work on any computer. Should synchronize system clocks to server machines.

- Can export a Web Client or report.

- Should not perform any of the tasks the server process has been assigned. Lookout client software does not support certain server features such as logging to Citadel or communicating with hardware.

Figure 2-1 shows an example of a Lookout application setup.
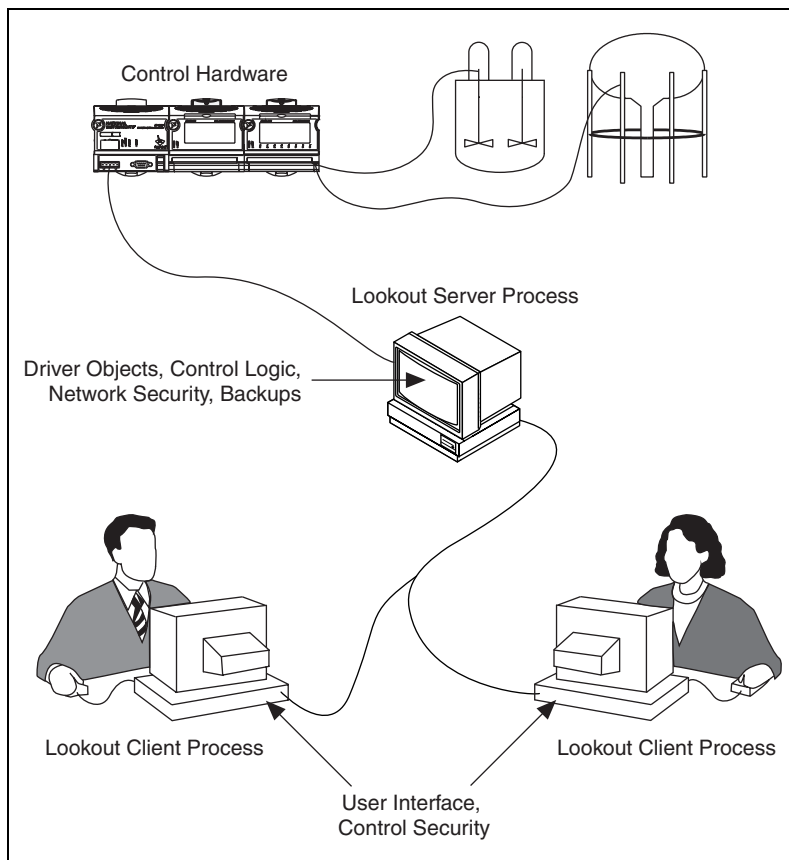
Figure 2-1. Example of a Lookout Application Setup

Development Process

The following steps are a general outline that you can use in developing your application with Lookout.

1. Plan your application requirements such as control hardware, computer operating system, networking, topology, code reuse, security, logging, redundancy, and performance. Do not skip this step, because it impacts your decisions at every stage of development. Design decisions that you make during development may be difficult to change later.

2. Connect your computer(s) to the control hardware. Configure, test, and troubleshoot the connection. Refer to the control hardware's documentation for information about connecting to it.

3. Create your Lookout server process(es). In a server process you implement direct communication with your control hardware (using driver objects with scaling, logging, alarms, and aliases configured), control logic, network security, and backups. Refer to relevant chapters later in this manual for more information.

4. Create your Lookout client process(es). A client process is the interface intended for operation of your application. When creating a client process, you connect to server process (using a Symbolic Link), set up alarms, build a graphical user interface, and implement control security. Refer to relevant chapters later in this manual for more information.

5. Set up reports. You have several options for generating printed or online reports from your Lookout application. Refer to Chapter 12, *Reports and Web Clients*, for more information about setting up reports.

6. Deploy, network, document, and troubleshoot your application. Refer to Chapter 13, *Networking and Deploying an Application*, for more information.

Design Recommendations

The following sections highlight some important considerations that you might find helpful as you begin to develop your application.

Choosing an Operating System

Lookout is supported on Microsoft Windows operating systems only, including Windows Me/98 and Windows 2000/NT/XP. You may prefer Windows Me/98, but National Instruments strongly recommends using Windows 2000 SP1, Windows NT with Service Pack 6a, or Windows XP. These systems are designed for business use and deliver reliable solutions. You can read more about Windows operating systems and choosing an operating system on Microsoft's Web site.

Protecting Against Power Failure

Power failure is a potential threat to system stability. If a computer is turned off suddenly, data kept in memory is lost and your files may be corrupted. The only effective protection against power failure is an Uninterruptible Power Supply (UPS). A UPS is a device that allows your computer to keep running for at least a short time when the primary power source is lost. It provides complete isolation between the AC power source and the computer and has backup battery power if there are blackouts and brownouts. A quality surge protector will protect your computer from most electrical surges and spikes if you do not need battery backup.

Designing a Redundant System

To maximize system robustness, design your system to be redundant. With Lookout, you can configure two process control computers for redundancy, providing automatic transfer of control if the primary computer fails. Refer to Chapter 14, *Setting Up Redundancy*, for more information about setting up redundancy. If you plan to implement redundancy for your system, refer to that chapter for important information about how you must set up your processes, so that you can avoid rework later.

Backing Up Data

Make sure you periodically back up your historical data. Your Citadel historical data cannot be backed up by third-party tools while it is in use. You must stop the Citadel service before making a copy. However, you can use the CitadelControl object in Lookout or the Historical Data Viewer in

MAX to back up data without stopping the Citadel service. Refer to the *CitadelControl* topic in the *Lookout Help*, or to Chapter 8, *Logging Data and Events*, for more information about logging and backing up data.

Implementing a Server/Client Design

It is best to use a server/client design as explained in the *Lookout Client/Server Application Design* section of this chapter. You can develop both on the same computer, though you may want to develop them in separate instances of Lookout to better simulate the connections.

Using the Lookout Help

The *Lookout Help* contains detailed information about each object and instructions for completing common tasks in Lookout. Refer to the *Lookout Help* by selecting **Help»Contents** in Lookout.

Using Powerful Objects

You should not overlook or miss these objects, which are likely to be critical to the success of your process development:

- DataTable—Multiplex various data sources onto a single display template, import or export large quantities of data using DDE, import or export data in Excel spreadsheet format, connect to an ODBC source, and execute an SQL statement.

- Sequencer—Cycle through a collection of up to 100 differently configured ON/OFF states for your process, and manage flow control.

- Symbolic Link—Make certain kinds of remote connections easier and more efficient, and manage failover redundancy in Lookout.

- Aggregate—Create a collection of Lookout objects, connected and configured, and use that collection multiple times as an element in your processes.

- Report—Create an HTML page that you can put up on a Web server to allow people to access the reported information.

Refer to the *Lookout Help* for more information on these and other Lookout objects.

Simplifying Your Folder Hierarchy

As you create objects in Lookout, consider organizing them in a meaningful hierarchy of descriptive folders, as shown in the following illustration.
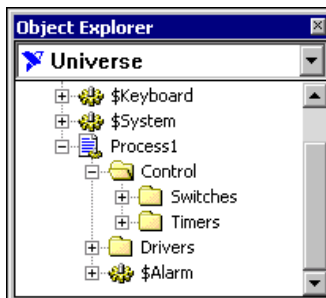
In some circumstances, however, you might need to maintain a flat hierarchy of folders and short paths to data members. Character limitations may apply when performing SQL database queries or selecting start-up processes. You might find it is easy to inadvertently create long names, because paths to Lookout data members are in the form of `ProcessName/Folder/Folder/Object.datamember`.

Lookout warns you when you try to log a data member that is longer than 62 characters, because some applications such as Microsoft Query, Microsoft Access, or Visual Basic, cannot handle names longer than 62 characters if you use them to perform SQL queries on your Citadel database. Others can handle names up to 126 characters long, but any names longer than the maximum acceptable name length are excluded from queries. Refer to Appendix B, *Structured Query Language*, for more information about SQL queries.

You may also encounter character limitations if you need to specify multiple start-up processes. Refer to Chapter 13, *Networking and Deploying an Application*, for more information about selecting start-up processes.

Ensuring Adequate Client Licenses

Make sure that you have the correct client licenses in place for your application. A client license is required for each concurrent Lookout client that needs to connect to a server process running Lookout. Lookout client software and Web Clients require a client license. Contact National Instruments for assistance in choosing the licensing option that best suits your needs.

Making Connections Correctly

Making the correct kind of connections as you develop your server and client processes can save you troubleshooting and rework time later. Refer to the *Connecting Objects* section of Chapter 5, *Creating Lookout Processes*, for more information about connection types and the appropriate path modes to use in your connections.

Implementing a Control Strategy

You should carefully plan your control strategy. For example, you might implement control logic such as monitoring a hardware output and comparing that value to unacceptable high and low limits. You could then have your process respond to the reported value in a number of different ways, depending on the result. You should decide how often to poll certain values and what events should take place once a value leaves its acceptable range. Refer to the *Control Logic* section of Chapter 5, *Creating Lookout Processes*, for more information about setting up control logic.

Taking Advantage of Object Explorer and Connection Browser

The Object Explorer and Connection Browser utilities in Lookout are efficient tools for viewing and editing the objects and connections in your process. Both of these utilities are accessible from the **Objects** menu while in edit mode. Refer to Chapter 1, *Introduction to Lookout*, for more information about these utilities.

Ensuring that Logging is Enabled

By default, data members are not configured for logging. You must modify the configuration of each data member whose value you want to store in Citadel. For best performance, enable logging for only those values that need to be logged, and configure appropriate deviation filters and alarm deadbands for your data members. Refer to the *Editing Data Member Configuration* section of Chapter 5, *Creating Lookout Processes*, for more information about modifying data member configuration.

Multiplexing Panels in Complex Processes

If your application requires multiple control panels that are similar or exactly the same, consider multiplexing your data onto one single panel for a simpler, more efficient design. Refer to the *Creating Panels and Navigation* section of Chapter 5, *Creating Lookout Processes*, for more information about multiplexing panels and designing navigation schemes for your operators.

3

Tutorial

This chapter gives a brief tutorial to help you become familiar with the basics of Lookout development.

Create a Server Process

In this exercise, you simulate connections to hardware. You will use this process as the server process in later exercises.

Note While *server* is a networking term, a Lookout server process does not necessarily need to be connected to a network.

This exercise shows how easy it is to create and configure Lookout objects, make and edit connections between those objects, and configure data for logging.

1. Start Lookout by selecting **Start»Programs»National Instruments Lookout**.

2. Browse the information offered in the default process by clicking the topic links.

3. Close the default process by selecting **File»Close All**. Click **Yes** to stop the process.

4. Select **File»New** to create a new Lookout process.

   Notice that the status bar at the bottom of the Lookout workspace is now yellow. This signals that Lookout is in edit mode. If you have trouble completing any of the steps in this tutorial, check to see whether Lookout is in edit or run mode. You can change modes by pressing <Ctrl-Space> or selecting **Edit»Edit Mode**.

5. In the **Create Process** dialog box, name your process Server, as shown in the following illustration. Leave the rest of the fields with their default settings.

Create Process ⊠

Process **N**ame:

| Server |

── State Information: ─────────────────────────────

  ⦿ Save State File with **P**rocess File

  ○ Save State File in **L**ookout Folder (Based on Process Name)

  ☐ Save **S**tandby State File:

    | | **B**rowse...

  ☑ Sa**v**e state file(s) every | 60 | (1-1440) minutes.

── Citadel Database: ──────────────────────────────

  ☑ **U**se Default Values

  Computer Name:

  | paroikos.dy.natinst.com |

  Citadel Database **F**older:

  | f:\national instruments\lookout\database |

      [OK]   [Cancel]   [**H**elp]
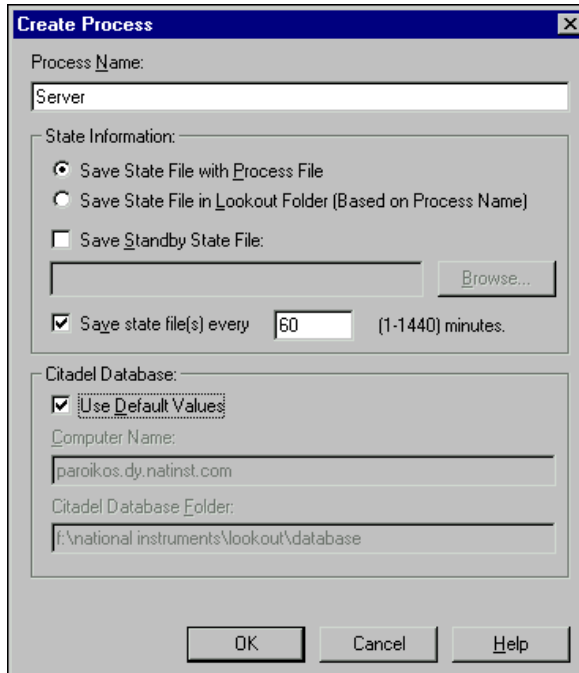
6. Click **OK**.

7. In the **New Control Panel** dialog box that appears, change the **Title** (not the **Name**) to Lookout Server Control Panel.

8. Click **OK**.

9. Select **Object»Object Explorer**.

   Use the Lookout Object Explorer to examine Lookout processes running on any Lookout computer on your network, create new objects, arrange objects in hierarchical folders, make connections, edit data members, and register computers running Lookout on your network.

10. Arrange your Lookout workspace so that you can see both your control panel and the Object Explorer.

Create a Server ModbusSlave Object

✎ **Note** These exercises use the ModbusSlave driver object because it does not return errors if it cannot locate hardware connected to your serial port. You can use the ModbusSlave driver object as you would any PLC or RTU driver object, if your control hardware is not available.

1. Right-click the `Server` process in the Object Explorer, and select **New Object** from the menu. The Select object dialog box appears.

2. Select the **ModbusSlave** object, which is in the **Drivers** folder if the objects are categorized. Notice that a description of this object appears in the field to the right of the list of objects.

3. Click **OK**. The New Modbus Slave dialog box appears, as shown in the following illustration. For this example, accept the default values for the object `ModbusSlave1` by clicking **OK**.
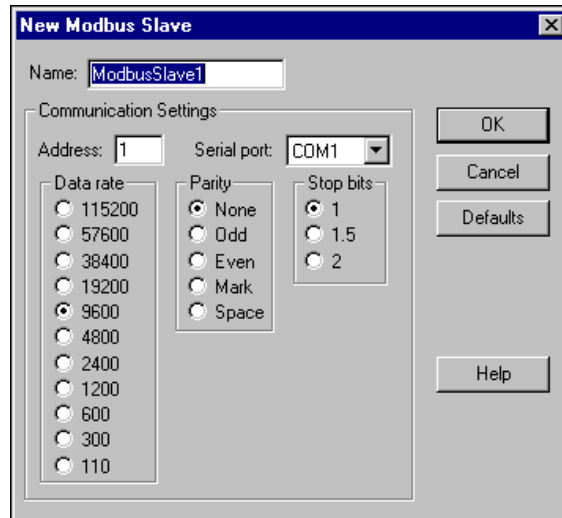
The Modbus Slave object, `ModbusSlave1`, appears under your `Server` process in the Object Explorer.

Create and Connect a Server Pot Object to Your Modbus Slave

Typically, you do not need to develop the user interface for a server process, except for perhaps a few displays to help you verify and troubleshoot connections to hardware. The following steps show how to add some simple displays. In the *Create a Client Process* section, you will construct a more detailed user interface.

1. Expand the tree view of your newly created `ModbusSlave1` object by clicking the + sign to the left of it in the Object Explorer, as shown in the following illustration. This lets you view the data members available for you to use.

A data member is a data source or sink associated with a Lookout object. In the following steps you will display the ModbusSlave's `40001` data member on your control panel. This data member might represent a particular hardware register.

2. Right-click the `40001-49999` data member range in Object Explorer and drag it to your control panel. Release the mouse button and select **Create Control»Pot**.
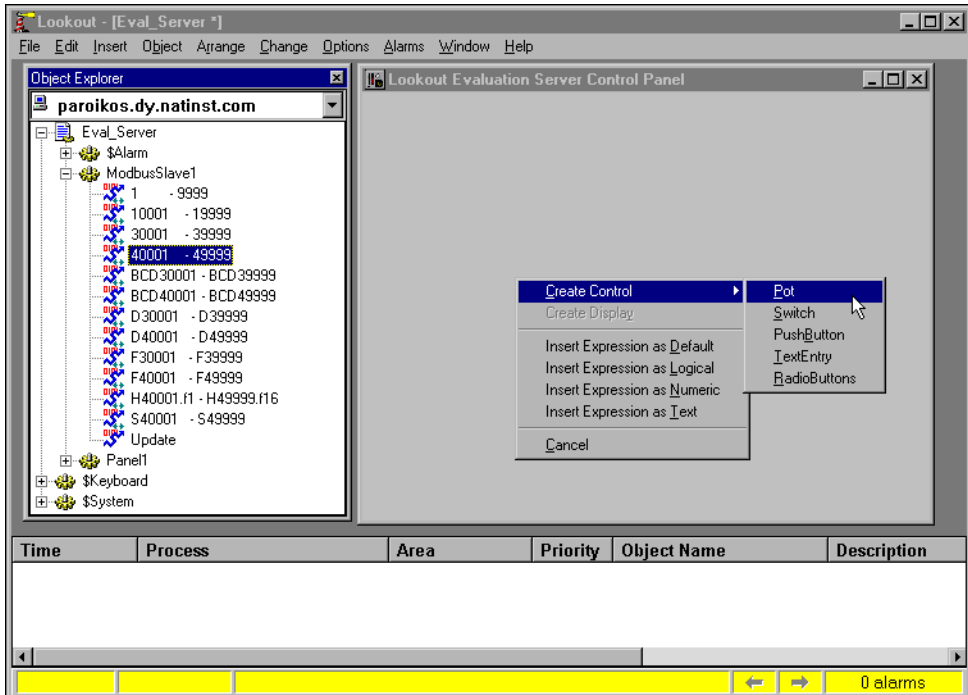
💡 **Tip**   Pot is an abbreviation for potentiometer. In Lookout, you use Pot objects to set or control analog outputs.

3. The New Potentiometer dialog box appears, as shown in the following illustration. It is connected to `40001` because the first available data member is used by default when you select a range.

New potentiometer:

Name:	Pot1
Minimum:	0
Maximum:	100
Resolution:	1

Position source

○ Local
⊙ Remote
URL := ModbusSlave1.40001

○ DDE
Service:
Topic:
Item:

Control security level: 0   ☑ Log events

[OK]   [Cancel]   [Help]
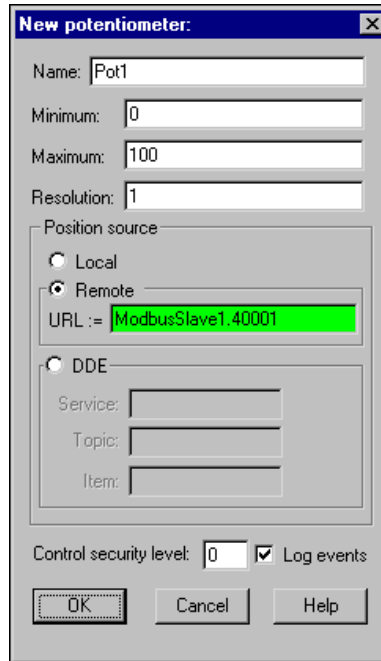
4. Accept the default values for Pot1 by clicking **OK**. The Insert Pot dialog box appears.

5. Select the vertical slider (top row, second graphic). Select the **Use as Default** checkbox, then click **OK**. You can position your object anywhere on the control panel.

6. To check that your connection works, place a digital display of the ModbusSlave1.40001 data member on your control panel by left-clicking 40001–49999 in the Object Explorer and dragging it to your control panel.

7. Use the sizing handles around the number to make the display large enough to show three digits.

8. Right-click the control panel and select **Insert Text/Plate/Inset** and type a percent full label for your digital display with a transparent background, as shown in the following illustration. Click **OK** and position the label next to the digital display.
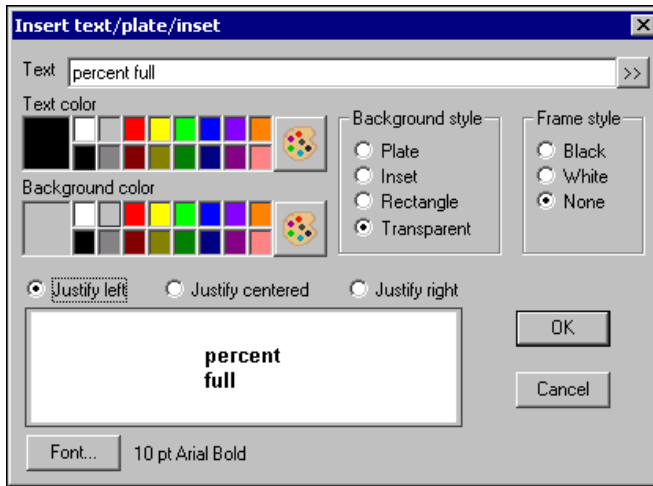
9. View your process in action. Press <Ctrl-Space> to leave edit mode, then use your pointer to move the Pot slider up and down. Notice the number in the digital display changes as you move your Pot control, as shown in the following illustration. The Pot slider object output is connected to the ModbusSlave1 data member 40001, and the value the Pot sets for that data member is shown in the digital display.
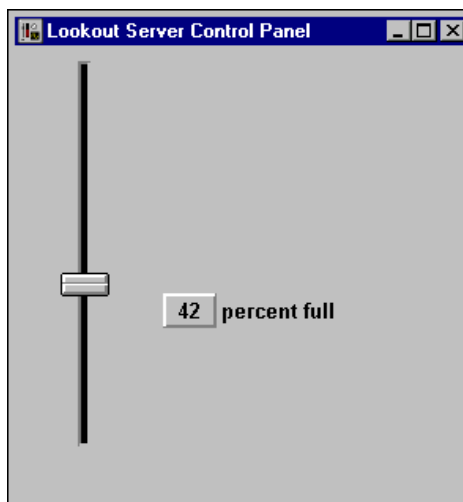
10. Toggle back to edit mode by pressing <Ctrl-Space>.

11. Select **Object»Connection Browser**. In the **Objects** column, double-click your computer name, then Server, then Pot1 to see the Pot's connection to the Modbus Slave 40001 data member in the **Inputs** column. Click **Close**.

The connection between your Pot control and the Modbus Slave driver is an example of a remote source connection. This two-way connection makes the Pot an indicator as well as a control. For more information about the various types of connections, refer to the *Connecting Objects* section of Chapter 5, *Creating Lookout Processes*.

The digital display is an example of an expression. To learn more, right-click the digital display, choose **Object Properties**, then click the **Help** button in the **Revise expression** dialog box that appears.

Configure Data for Logging, Scaling, and Alarms

The following steps show how easy it is to configure logging, scaling, alarms, and more, for your data. You might want to show how many gallons of liquid the tank is holding, display information about the tank and its contents graphically, and trigger alarms when the tank is too full or almost empty. To keep track of this data and to make it easier to handle, Lookout uses the internal Citadel database. Each time you create a Lookout object, every one of its data members is a potential entry in Citadel.

Assume your tank holds 250,000 gallons. You can configure a Citadel entry for your tank to track how full the tank is (in gallons), convert the percentage measurement to a volume measurement, and handle your logging and alarming.

1. Right-click the ModbusSlave1 object in the Object Explorer. Select **Edit Data Member Configuration**.

2. Using the following illustration for help, enter these values in the dialog box.

   a. Select the 40001 data member you connected your Pot object to by clicking 40001-49999 in the **Native members** list. Assign it an **Alias** of TankLevel.

   b. In the **Scaling** section of the dialog box, set the **Raw units** at 0 for the **Minimum** and 100 for the **Maximum**. These are the raw numeric values connected to the 40001 data member, indicating what percentage of the tank is currently full. Set the **Eng. units** to 0 for the **Minimum** and 250000 for the **Maximum**.

Lookout will now automatically scale the 0–100 input to a value between 0–250,000 that you can display using the alias value `TankLevel`.

c. Check **Log to historical database**, and set the **Lifespan** to 5 days.

d. Set the **Deviation** to `1.5`. (This will be used in the next exercise.)

e. Set your **Alarm conditions** as shown. Notice that Lookout lets you divide alarms into groups, called alarm areas. In this case, the alarm **Area** is `Tanks`.

f. Click the **Save** button (or the **Update** button if you saved earlier in the process), then click the **Quit** button to exit the dialog box.
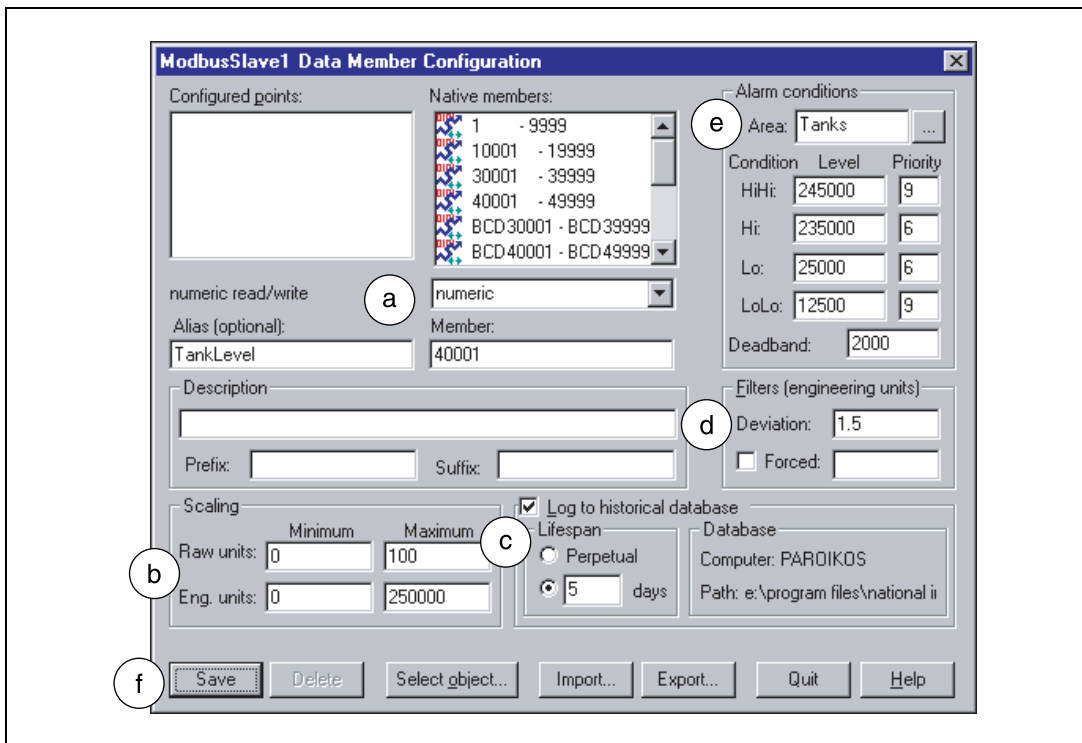
3. Expand the `ModbusSlave1` object in the Object Explorer by clicking the +, and drag the scaled data member `TankLevel` to your control panel. This digital display reports the contents of your tank in gallons.

4. Select **Insert»Text/Plate/Inset**, and type a `gallons` label for this digital display, choose a transparent background, and click **OK**. Position the label next to the scaled tank level display.

Your control panel should now look similar to the following illustration.
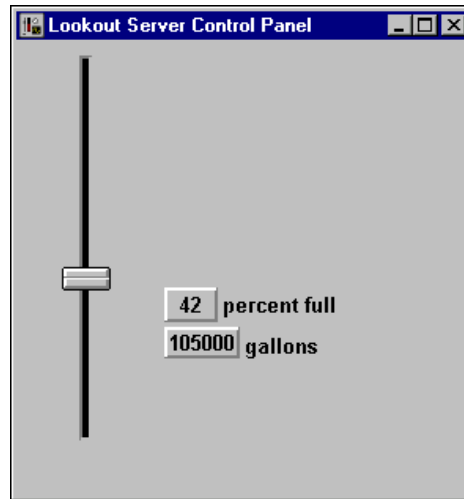
5. Press <Ctrl-Space> to exit edit mode, then move the Pot slider up and down. Moving the Pot to the upper and lower limits of its range triggers the alarms you set for TankLevel in step 2e. Notice that the alarms appear in the Alarms Window. If you cannot see the Alarms Window, select **Alarms»Show**.

6. Right-click an alarm in the Alarms Window and select **Acknowledge All** to acknowledge the alarms. You can click **OK** without entering a comment.
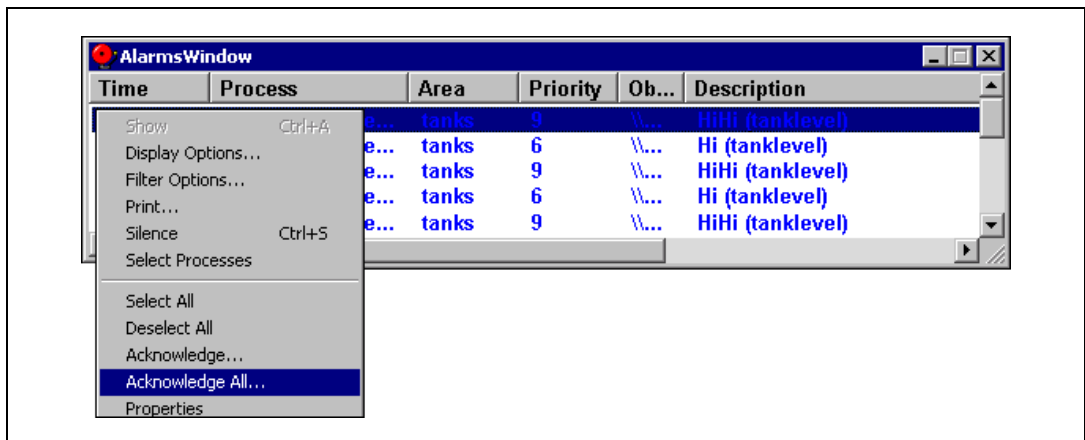
7. Save your server process by selecting **File»Save**. You will be accessing it with the client process you create next.

💡 **Tip**   If you want to change the size and location of the Alarms Window, select **Alarms»Display Options**. To control what is displayed in the Alarms Window, select **Alarms»Filter Options**. For a more detailed description of alarms and alarming, refer to Chapter 9, *Alarms and Events*.

Review—Creating a Server Process

A Lookout server process runs on the computer that has direct or physical connections to your industrial automation hardware. Configure the data for logging, alarms, scaling, and so on in the server process. A server process provides data for clients, and does not need an elaborate user interface.

💡 **Tip**   You can add network and local security to this server process. To learn more about using the Lookout security features, refer to Chapter 10, *Security*.

Create a Client Process

In this exercise, you build a simple Lookout client process to interact with the server you built in the *Create a Server Process* exercise.

The instructions for this exercise assume you are developing your client process on the same computer that is running your server process. In the *Networking the Client Process* section, you will have a chance to move the client to another computer, add a feature, and fix a simulated data quality problem.

1. If the Server process is not already running, launch Lookout and load it using the **File»Open** command.

    If you did not save your server process from the previous example, you can use the Server sample process, found in the Examples folder installed with your copy of Lookout.

2. Minimize Lookout by clicking the minimization box (▬) in the upper-right corner of the Lookout window.

3. Launch a second instance of Lookout, so that it will be easier to construct proper paths for your client process. If your computer performance suffers from running two instances, you can create your client in the same instance of Lookout running your server process, but be careful to form your paths correctly. Refer to the *Path Modes in Lookout* section of Chapter 5, *Creating Lookout Processes*, for more information about Lookout paths.

4. In the second instance of Lookout, close and stop the default process, and select **File»New** to create a Lookout client process.

5. In the **Create Process** dialog box that appears, name your new process `Client`. Click **OK**.

6. In the **New Control Panel** dialog box that appears, change the **Title** to `Client Control Panel`, and click **OK**.

Configure the Panel to Activate Automatically

By default, Lookout control panels are minimized. The following steps show how to make your client control panel active and visible every time you load the process.

1. Select **Object»Edit Connections** and navigate to the Panel object in your client process. Click **OK**. The following dialog box appears.
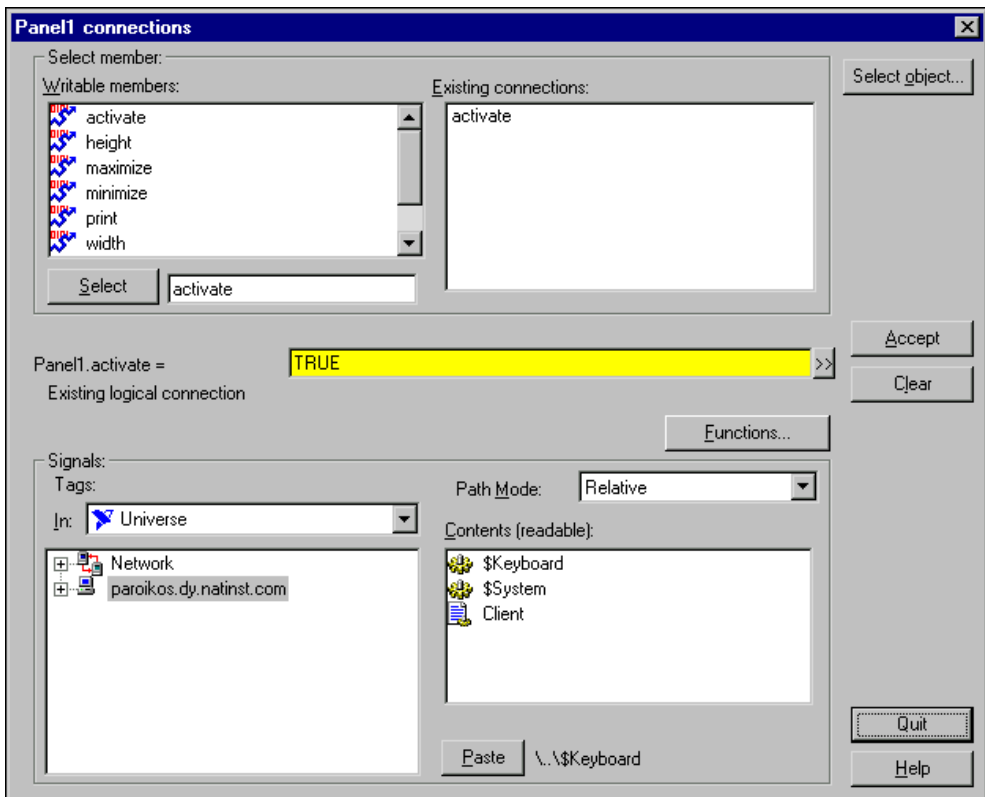
2. Select `activate` from the **Writable members** field and click the
    Select button. Type `TRUE` in the yellow expression field, and click
    Accept. Your client control panel will now activate when you first open
    the process.

3. Click **Quit**.

Set the Root

To work with your processes properly, you need to set the Object Explorer
root.

1. Select **Object»Object Explorer**.

2. Select **Universe** from the combobox at the top of the **Object Explorer**
    window.

3. Expand your view so that you can see both the network presence of your computer and the local presence, as shown in the following illustration.
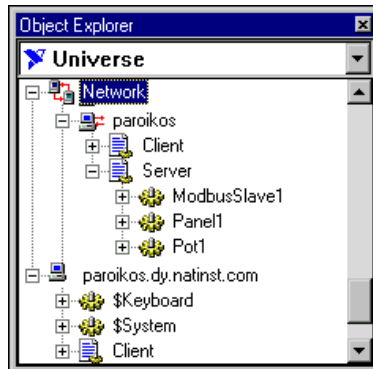
Notice that the Network branch of the tree shows both of your processes, but the local branch shows only your client process. The local branch shows only those processes running in the instance of Lookout you are currently working in. Any process running in another instance of Lookout, whether on another computer or the same computer, appears under the Network branch only.

Create a Client Pot Object

1. In the Network branch of Object Explorer under `Server` and `ModbusSlave1`, right-click and drag the `40001-49999` data member range to your `Client` control panel. Release the mouse button and select **Create Control»Pot**. This Pot will have a remote connection to the ModbusSlave object, defined by a URL.

    The path shown in the green field of this dialog box will work no matter which computer on your network is running this client process.

2. Click **OK** in the **New potentiometer** dialog box, then select the vertical slider and click **OK**.

3. Press <Ctrl-Space> to leave edit mode.

4. Switch to the instance of Lookout running your server process and, if necessary, activate the control panel by selecting **Window»Lookout Server Control Panel**. Press <Ctrl-Space> to leave edit mode in this instance of Lookout if necessary.

5. Arrange your two instances of Lookout so you can see both control panels. Operate the Pot control in the *client* process control panel.

Notice that on the server process control panel, both the Pot control and digital displays respond.

6. Operate the Pot control in your *server* process. Notice that the Pot control in your client process changes to reflect movement of the server values.

Connect a Client Switch Object to Your Modbus Slave

1. In the instance of Lookout you are using to create your client process, press <Ctrl-Space> to return to edit mode (the status bar at the bottom of the Lookout workspace should turn yellow).

2. Right-click your `Client` process in the local branch of the Object Explorer, and select **New Object**.

✎ **Note** Lookout does not allow you to edit a process remotely. If you right-click the `Client` process in the Network branch of Object Explorer, you will not be able to access the **New Object** option.

3. Select **Switch**, which is in the **Control** category if the objects are categorized, and click **OK**. The following dialog box appears.

4. For **Position Source**, select the **Remote** option, and right-click in the green **URL** field. The URL Editor dialog box appears.

When you created your Pot object for your client, Lookout automatically used an absolute path mode to make sure the URL would work from any computer on your network. In this case you will create a computer-relative path that will later fail when you move the client process to another computer. This will demonstrate Lookout's ability to alert you to data quality problems, and give you a chance to learn how to correct such problems.

5. Expand the Network node in the tree that appears, expand your computer under the Network node, then expand the Server process. Select the ModbusSlave1 object as shown in the following illustration.
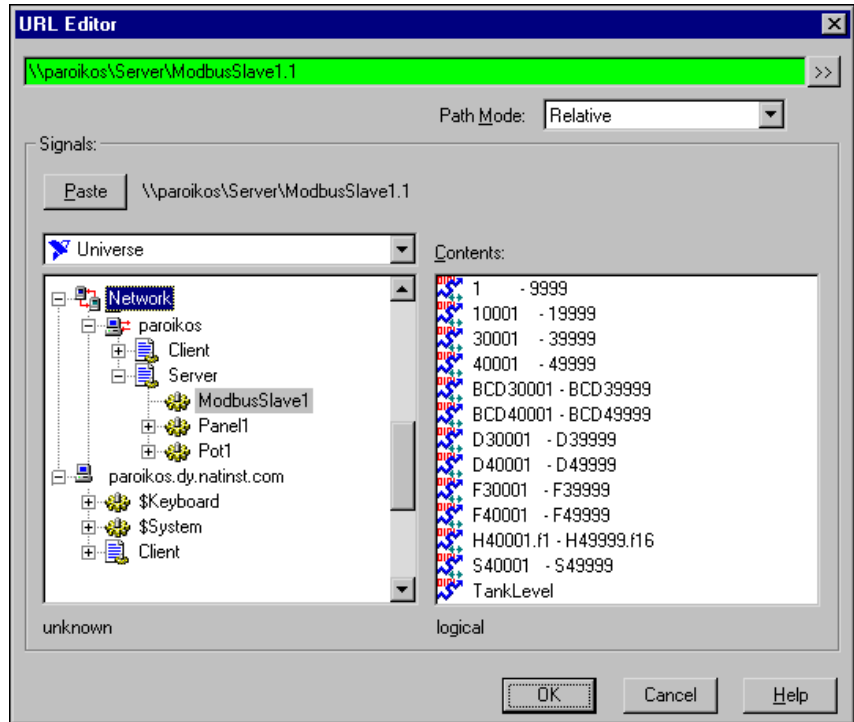
6. Select the 1-9999 data member in the **Contents** list and click **Paste** to insert the URL string into the green URL field. The URL is an absolute path.

7. In the green field, delete your computer name and substitute a period (.) to convert the absolute path to a computer-relative path. The final string should be the following:

    `\\.\Server\ModbusSlave1.1`

    Use computer-relative mode when you want to reference other processes running on the same computer. Once you move the process with this Switch to a client computer, the Switch will attempt to find its remote source on that client computer.

8. Click **OK**.

9. Click **OK** in the **New switch** dialog box to finish creating your Switch object.

10. Drag the Switch object from the Object Explorer to your client control panel. Select the standard switch on the far right as shown in the following illustration, and click **OK**.
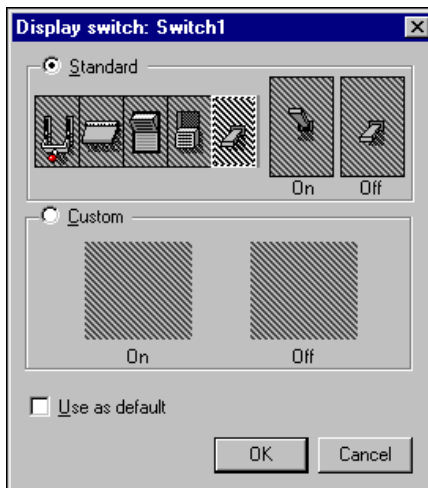
Create a Symbolic Link

So far you have created a Pot in your client process that has a remote source connection to the server, which will work no matter where you move the client process on your network. You also created a Switch that will work when the client process is running on the same computer as the server process. However, if you move your server process to a new computer, or otherwise change the server computer's name, your connections will break. Repairing these broken connections becomes difficult if you have a large number of them. To make maintenance easier, you can use a Symbolic Link connection between a client and a server.

1. In the Object Explorer of the instance of Lookout running the client process, right-click the local branch of your Client process and select **New Symbolic Link**. The Create Symbolic Link dialog box appears.

2. Name the Symbolic Link Link_to_Server and enter the absolute path to your server process in the **URL** field, in the following form:

    your_computername\Server
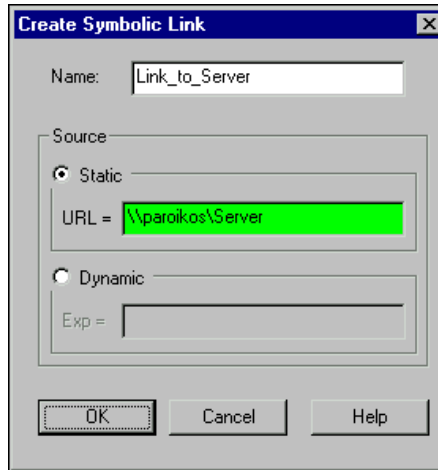
3. Click **OK**.

4. In the local branch of Object Explorer, expand the Symbolic Link you just made. Notice that all the objects in the server process are exposed in the Symbolic Link.
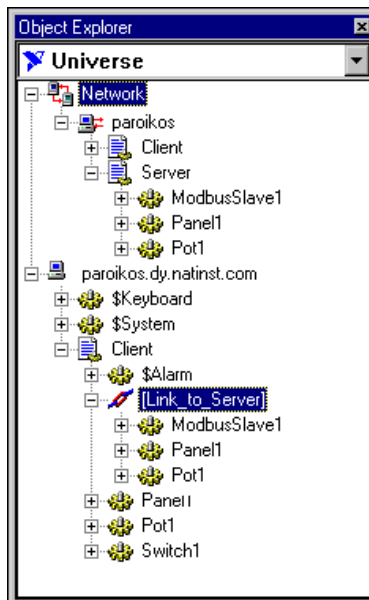
5. Expand the `ModbusSlave1` object under `Link_to_Server`. Drag and drop the `1-9999` data range to your client control panel to insert an LED display that will reflect operation of the Switch you made.

6. Right-click the LED and select **Object Properties** to see that the expression is as follows:

   `[Link_to_Server]\ModbusSlave1.1`

   This indicator is now linked to the server process through the Symbolic Link. You can, in fact, make all your connections through the Symbolic Link, making your client more flexible, portable, and reusable. By editing the path in the Symbolic Link, you can change every connection in your process with one operation instead of changing connections one at a time.

7. Click **OK**. Save your client process.

Create a HyperTrend Object

In the *Configure Data for Logging, Scaling, and Alarms* section, you configured the `TankLevel` data member in your server process for scaling, and set it to be logged to Citadel. As long as your server process is running, Lookout is keeping track of this value in Citadel.

1. Make note of the current time on your computer, then switch to run mode and change the position of the Pot slider on your client or server process a number of times in large increments. Later, you will view the changes you made on the HyperTrend.

2. Press <Ctrl-Space> to return to edit mode.

3. In Object Explorer, right-click your computer in the local branch and select **Make Root**. You no longer need access to the `Server` process in the network view.

4. Right-click your `Client` process in the Object Explorer and choose **New Object**. Select **HyperTrend**, which is in the **Logging** folder if the objects are categorized, and click **OK**.

5. In the **Create HyperTrend** dialog box, click the **Add Group** button. The following dialog box appears.
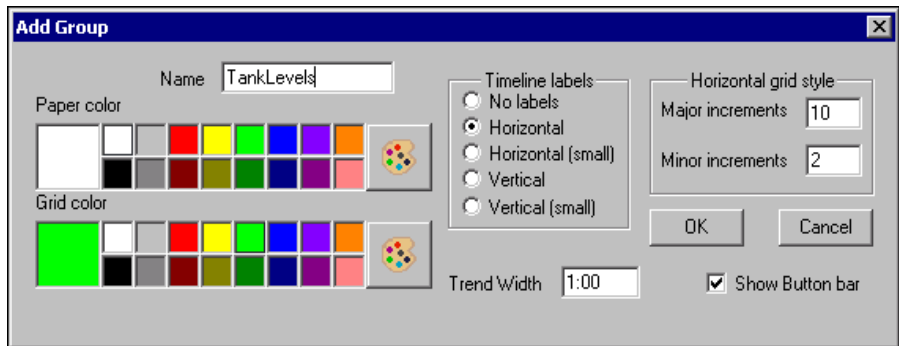
6. Name the group `TankLevels`. The default trend width is one minute; accept this and the other defaults.

7. Click **OK**. Lookout returns you to the Create HyperTrend dialog box. Make sure your new `TankLevels` group is highlighted, and then click the **Add Item** button. The following dialog box appears.
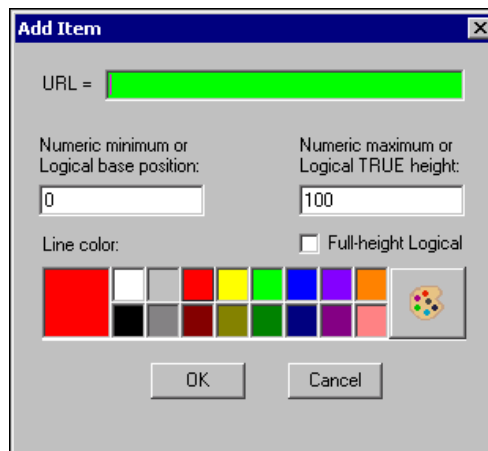
8. Right-click in the green **URL** field to activate the **URL Editor**.

9. Set **Path Mode** to **Relative**, then under your computer name in the left tree view, select the `ModbusSlave1` object under `Link_to_Server`. Select the `TankLevel` data member in the **Contents** list. Click the **Paste** button. The path in the green field should be `[Link_to_Server]\ModbusSlave1.TankLevel`.

10. Click **OK**.

11. Set **Numeric maximum** to 250000 to match the scaled value of the TankLevel data member.

12. Click **OK** in the **Add Item** dialog box and again in the **Create HyperTrend** dialog box.

13. Drag your new HyperTrend object from the Object Explorer to your client process control panel. Use the sizing handles to adjust the size of the chart. Arrange your client process control panel to resemble the following illustration. You can add the scale shown by selecting **Insert»Scale**. You can add the Percentage full label by selecting **Insert»Text/Plate/Inset**.

| 1 | Modbus Slave 1.1 Switch | 2 | Modbus Slave 1.1 LED Indicator | 3 | HyperTrend Cursor |
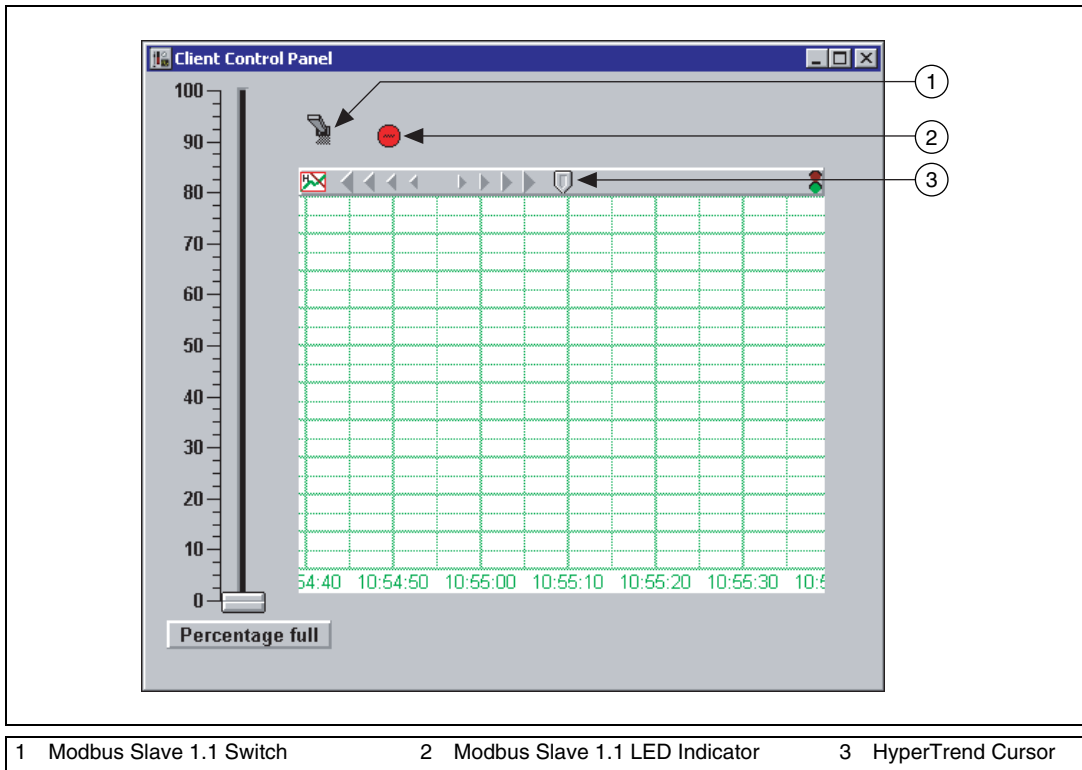
View Your HyperTrend Data

1. Press <Ctrl-Space> to toggle out of edit mode. Move the Pot slider up and down and notice the trace change on the graph.

2. Click the **HyperCursor** icon in the center of the top of the HyperTrend object. The HyperCursor icon is replaced by a solid line down the

center of the chart, and the HyperTrend1 Cursor dialog box appears, as shown in the following illustration.
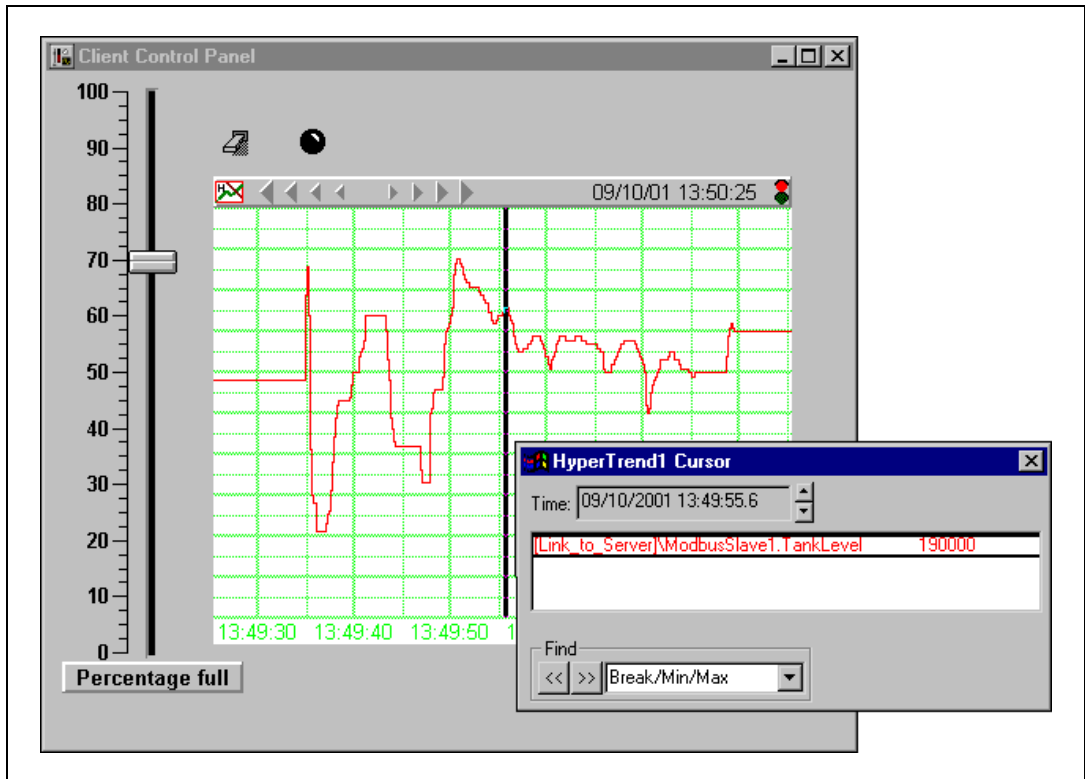
3. Use the **Time** control on the dialog box to scroll back to the time you noted when you changed the TankLevel value.

4. Select **File»Save All** from the Lookout menu to save both your processes. Make sure that your client process is saved in the Lookout folder, so that you can successfully export it as a Web Client.

Review—Creating a Client Process

A Lookout client process interacts with Lookout servers. It should be capable of running on any Lookout computer on your network without modification. To ensure this portability, a Lookout client application should not include driver objects, and it should always use remote connections to a server process.

Networking the Client Process

If you have access to a second computer on your network, you can run your new client on that computer and try out Lookout's easy-to-use networking. This exercise shows how a client process developed in a server computer can be distributed across any machine in a network. The client process works with absolutely no development.

1. Install Lookout on a client computer.

2. Copy your client process files to the Lookout directory on the client computer. If you did not change the name of your process when saving, you should have the following four files that comprise your client process: Client.l4p, Client.l4t, Client.lks, and Client.lka.

✏️ **Note**   For more information about Lookout file types, refer to the *Deploying Lookout Application Files* section of Chapter 13, *Networking and Deploying an Application*.

3. On the client computer, launch Lookout, select **File»Open**, and open Client.

4. Operate the Pot object slider on your client computer. The Pot controls in the server and client processes running on the first computer should respond.

5. Notice the red X over the switch, which indicates a data quality problem. This will be explained later in the *Correct Problems with Data Quality* section.

✏️ **Note**   If you do not see any traces on your HyperTrend, you may need to synchronize your computers. This will be explained later in the *Synchronize Computer Clocks* section.

Register Your Network Computers

While Lookout processes can communicate with each other across the network without any special preparations, you may need to register other computers running Lookout in order to browse across the network for development purposes.

1. On the client computer, press <Ctrl-Space> to put Lookout in edit mode. Select **Object»Object Explorer** if the Object Explorer is not already running.

2. Click the name of your client computer in the window at the top of the Object Explorer, and select **Universe** from the pull down list.

You can now browse every computer registered as running Lookout on your network.

3. Register your server computer (the one running your server process) by right-clicking **Network** and selecting **Register computer**.
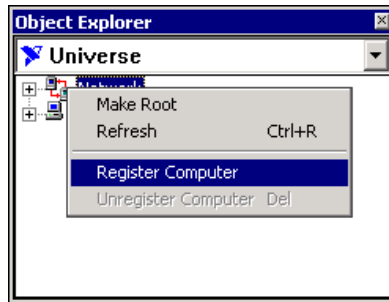
4. In the **Register Network Computer** dialog box that appears, type the name of your server computer into the **Computer name** field, as shown. If you prefer, you can browse for your computer name in the network tree view available in the **Entire Network** field.
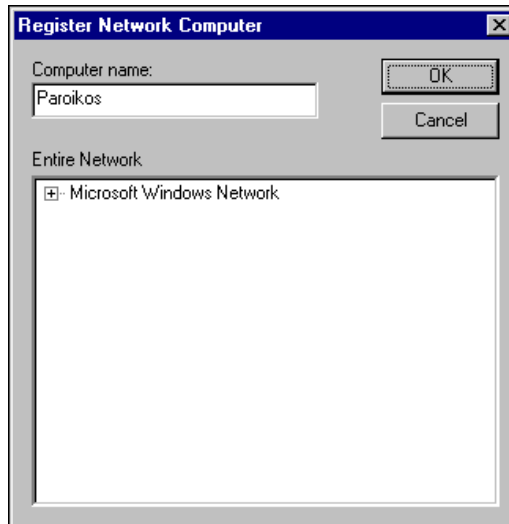
5. Click **OK**.

6. Expand the **Network** node of the tree view in the Object Explorer. You should see both your client computer (the computer you are currently working on) and the computer running your server process.

Correct Problems with Data Quality

The red X superimposed on the client computer's Switch object is an example of the Lookout data quality feature, which alerts you when your process is receiving data that has some problem. Complete these steps to fix the data quality problem.

1. Right-click the switch with the red X over it and select **Object»Properties**.

2. Right-click in the green URL field to change the computer-relative path to an absolute path.

3. In the **URL Editor** dialog box, navigate to your server computer under the `Network` branch. Expand the view to select `ModbusSlave1` under your server process.

4. In the **Contents** field, select `1-9999` and click **Paste**. The new path is an absolute path connection to that data member.

5. Click **OK** twice. The red X should be gone.

Synchronize Computer Clocks

You might notice some time lag in operating your Pot control and seeing the change in your HyperTrend chart on your server computer, which can happen when your networked computers do not have their clocks synchronized. To provide for accurate data logging, follow these steps to have Lookout synchronize your computers.

1. On the client computer, select **Options»Time Synchronization**. The following dialog box appears.
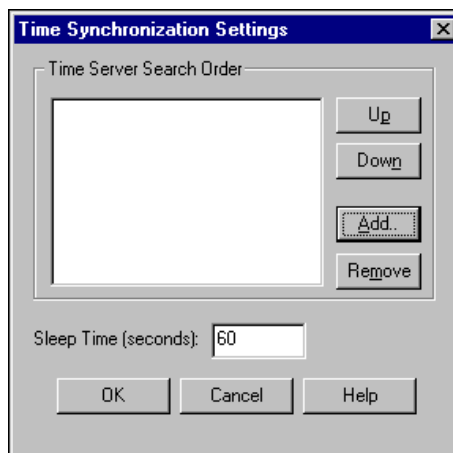
2. Click **Add**. The Add Time Server dialog box appears.

3. Enter the name of your server computer in the **Computer name** field. You can also browse for your computer in the **Entire Network** list.

4. Click **OK**. The Time Synchronization Settings dialog box reappears.

5. Set **Sleep time** to 10. With this setting, your computers resynchronize every ten seconds.

6. Click **OK**. Your computers are now synchronized.

Show Networked Alarms and Events

If you operate your Pot slider on your server computer, you may notice that the alarms you trigger on your server computer do not appear in the **Alarms** window on your client computer. To monitor and acknowledge alarms from a networked computer, you must select the processes you want to monitor as follows.

1. On your client computer, select **Alarms»Select Processes**. The following dialog box appears.
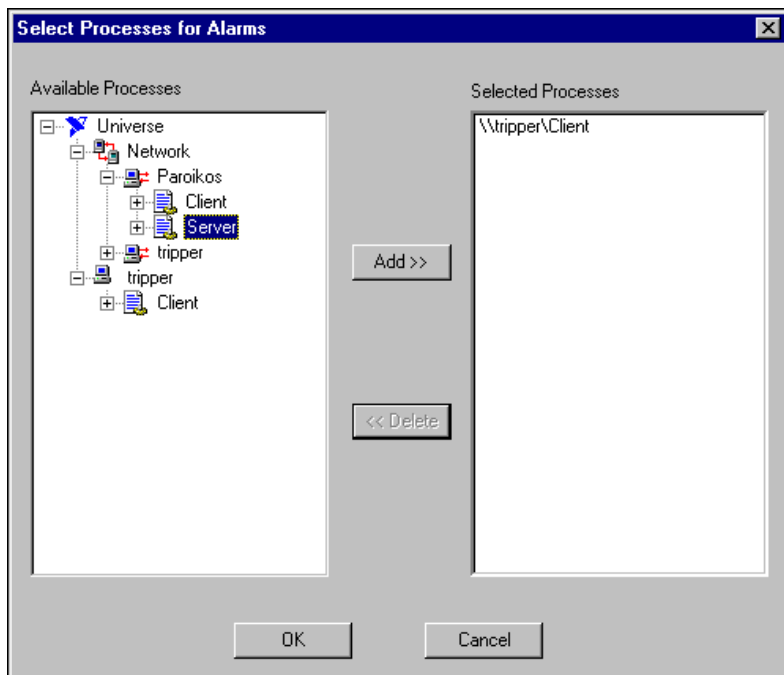
2.   Expand the tree view to show the processes running on your server computer. Select the `Server` process, and click the **Add** button. Click **OK**.

Your client computer can now monitor alarms from your server computer.

Review—Networking

Networking a Lookout application is as easy as copying your client process files to any networked computer. Then you can resolve any data quality problems, synchronize your system clocks, register computers, and access alarm and event information.

4

Serial COM Port Communication

This chapter describes how Lookout regulates serial port usage so that objects gain access to serial ports in an orderly and timely fashion.

Lookout does not associate baud rate, data bits, parity, or stop bits with a particular serial port. Lookout driver objects that implement different protocols and baud rates can use the same port and the same modem or radio frequency. This capability allows you to use various RTUs, PLCs, and other control hardware over a single radio frequency without communication conflicts or special hardware. For example, you can use a single two-way radio connected to a serial port to communicate with several different brands of RTUs out in the field, each one using a different protocol. Or, you can have 75 remote PLCs share a set of five dial-up modems.

Although many Lookout driver objects use serial communication, some Lookout driver objects do not use serial ports but instead rely on their own proprietary network cards to communicate with control hardware. A few examples include Modbus Plus (SA85 card), Data Highway (KT card), and DeltaTau (PMAC card). You do not need to configure serial ports for these objects. Refer to the *Lookout Help* for more information about whether a particular driver object uses serial communication.

How Lookout Allocates Serial COM Port Usage

Lookout allocates serial port usage between driver objects. At the frequency of the object **Poll Rate,** a driver object notifies the communication service that it needs to use a specific serial port to poll a device. If the requested serial port is not in use, Lookout allocates the serial port to the driver object. When the driver object takes control of the serial port, it defines port communication parameters such as baud rate and protocol and polls its device. When polling is complete, the driver object releases the port so that Lookout can allocate it to other driver objects.

You can uniquely configure each serial port for hardwired, radio, or dial-up communications. Refer to the next section, *Defining Serial COM Port Settings*, for more information.

Defining Serial COM Port Settings

Complete the following steps to configure serial port settings for hardwired, radio, and dial-up communications.

✐ **Note**   You must define serial port communication settings on every copy of Lookout. If you have more than one instance of Lookout running on the same computer, each instance must use a different serial port.

Because multiple Lookout instances cannot share the same serial port, if it is necessary for two processes to access the same serial port, they need to be run in the same Lookout instance. If you are designing a system with multiple server process files, and those server files all access the same serial port on a single computer, they need to be run in the same Lookout instance.

1. In Lookout, select **Options»Serial Ports**. The Serial Port Settings dialog box appears.

2. In the **Serial port** field, select the communication port you are defining. Microsoft Windows supports up to 64 serial ports; however, most computers support only two serial ports without additional hardware.

3. Define the serial port configuration for the appropriate communication port. The rest of this chapter contains complete descriptions of the configuration options.

4. Click **Accept** to save the configuration for the serial port.

5. Click **Quit** to exit.

Setting Receive Gap

The **Receive gap** setting is available for all serial connection types. This number specifies the number of empty bytes (or amount of time) a driver receives from a controller before the driver recognizes the end of a message frame and asks for another message. Normally you should leave this at the default setting of 20. However, if you are experiencing garbled communication alarms, you might try increasing this number to as high as 100 to allow more time before Lookout decides it has received a complete message.

Selecting the Serial Connection

Hardwired Serial Connection

Hardwired serial connections require no hardware handshaking for line control. Use this setting for all serial communication types except dial-up telephone and remote radio transceivers. You should also use this setting when directly connecting Lookout to the master repeater on a radio system or through a leased-line modem. Because a master repeater is a full-duplex device that does not require keying and unkeying of the frequency, it acts much like a physically hardwired network. Other hardwired connection types include RS-232, RS-422, RS-485, and leased telephone lines.

Dial-Up Serial Communication

Use the **Dial-up** serial connection when you use a modem in conjunction with a switched telephone line (not leased line). You can customize the dial-up settings for your particular modem and phone line.

The default **Dialing prefix** settings are based on the Hayes Corporation AT command set, which is an industry standard for data modems. Table 4-1 explains the Lookout default settings. For additional commands, refer to your modem operation documentation.

Table 4-1. Dialing Prefix

Prefix	Description
AT	Attention code that must precede all commands
Xn	Result code and dialing options: X4 waits for dial tone before dialing, and recognizes busy signal
Mn	Speaker on or off: M for speaker always off
Vn	Verbal or numeric result codes: V for numeric result codes
En	Local echo mode: E for no echo
D	Dial phone number with modifiers: P for pulse; T for tone

When you use an external dial-up modem with Lookout, the DTR line in your cable between the modem and the computer must be wired straight through. This line is pin 20 on a 25-pin RS-232 connector and pin 4 on a 9-pin connector. Lookout uses the DTR line to command the modem to disconnect (hang up) and return to the command mode.

Some factory modems are not configured to respond to the DTR line. After Lookout first successfully dials out to a remote modem and finishes the polling cycle, it drops the DTR line but the modem remains connected. If the modem does not respond after several seconds of Lookout attempting to raise and drop the DTR line, Lookout generates an alarm stating that the modem is not responding. If you receive this alarm message, your modem is not configured to monitor the DTR line.

The Hayes Corporation standard command for configuring the modem to hang up and enter command mode upon loss of DTR is &D2. You can use a terminal program to make this setting permanent on most modems by entering the modem command AT&D2&W to store the setting permanently in nonvolatile modem memory. Or you can just add &D2 into the **Dialing prefix**. The default **Dialing prefix** is ATX4MVEDT, so you might change it to AT&D2X4MVEDT.

Retries specifies the number of times Lookout dials the specified phone number and attempts to connect to the modem at the other end of the line. If Lookout fails to connect after the specified **Retries**, it generates an alarm and moves on to the next phone number in the polling queue (if a queue has formed).

Wait for connection specifies the length of time Lookout waits to receive a connect signal back from the modem it is calling. The time period begins when Lookout first sends the local modem the dialing prefix command. The time should be long enough for the local modem to receive a dial tone, dial the phone number, allow the remote modem to pick up the line, and send back a connect message. If the specified time is too short, your system could be operating correctly but never make a connection.

Pause between calls is the length of time Lookout waits after hanging up before it sends the local modem the next dialing prefix signal. If the specified time is too brief, your system might not hang up the existing call but still attempt to call the next number.

Note   Your specific modems, radios, and local phone lines might operate faster or slower than the default settings. You might need to use a trial-and-error approach to find the best settings for your system.

For information about troubleshooting dial-up connections, refer to the NI Developer Zone resources at `ni.com/zone`.

Radio (RTS/CTS) Serial Connection

RTS/CTS is a local hardware handshaking mechanism between the local computer and the local communication device. Use the **Radio (RTS/CTS)** serial connection when you connect the serial port to a device that requires RTS/CTS hardware handshaking, such as a radio transceiver that must be keyed up during data transmission and unkeyed during data reception. Other half-duplex communication media, such as RS-485, might require RTS/CTS hardware handshaking. Although the RTS/CTS scheme works identically for other RTS/CTS communication schemes, this example assumes that you are communicating through radio.

When you select RTS/CTS hardware handshaking, Lookout controls the RTS, or request-to-send pin, and monitors the CTS, or clear-to-send pin, during data transmission (pins 4 and 5 on a 25-pin RS-232 connector). Therefore, you must have at least the RTS pin (pin 4) wired *straight through* on your RS-232 cable. The CTS pin (pin 5) is optional.

Lookout initiates a serial transmission on an RTS/CTS port by first asserting RTS to key the radio. Lookout then begins monitoring the state of the CTS pin. When the radio transmitter is fully keyed and ready to transmit, the radio asserts CTS and Lookout immediately begins data

transmission. If the radio does not assert CTS within the **CTS timeout** setting (default is 100 ms), Lookout assumes the radio is ready to transmit and transmits anyway.

The **CTS timeout** setting is the maximum amount of time that Lookout waits after asserting RTS for CTS before transmitting. Most radios typically take between 10 and 80 milliseconds to key up. Consult your radio specifications and DIP switch settings to determine the key-up delay on your radio.

If your radio can assert CTS when it is ready to transmit, add about 50 milliseconds to the radio key-up delay specification and use this total value for the **CTS timeout**. If your radio does not assert CTS, you should begin by adding about 20 milliseconds to your radio key-up time. Then, increase this value in 10 millisecond increments until the remote radio begins to correctly receive the first bytes of the message.

Some radios might assert CTS before they are actually ready to transmit. In this case, disconnect the CTS line (pin 5 on a 25-pin RS-232 connector) and set the **CTS timeout** to a value high enough to let the radio fully key before transmission.

After it transmits the last byte of data, Lookout continues to assert RTS, keeping the radio keyed until the **RTS delay off** time period expires. You should set this value to the default of zero milliseconds so that Lookout unkeys the radio as soon as possible to prepare to receive the response.

When unkeyed, most radios generate an audible squelch tail that the remote device might decode as unexpected garbage bytes. Some remote devices reject the entire message instead of just decoding the valid data and ignoring the extra garbage bytes. In this case, keep the radio keyed for several milliseconds using the **RTS delay off** setting. This time period delays the squelch tail long enough for the remote device to recognize the last data frame as valid before receiving garbage bytes caused by the squelch tail.

If you set the **RTS delay off** setting too high, the remote device begins transmitting its response before the local radio is unkeyed, causing a communication alarm in Lookout.

Serial Port Hangup

You can configure your serial port to use +++ATH hangup as well as DTR hangup. Every serial port you have configured will have a configuration section in the `lookout.ini` file under the port name, such as `[COM1]`. Add the following entry to the file to set your hangup mode:

`DTR_Hangup=`*N*

When *N*=1 (default), that port uses DTR hangup. When *N*=0, the port uses +++ATH hangup.

Diagnosing Serial COM Port Problems

You can create serial port diagnostic files to help solve serial port communication problems.

1. Select **Options»Serial Ports**.
2. Check the **Enable** checkbox.
3. Enter the **File name** to which you want Lookout to log communications, as shown in the following illustration. If you enter a file name only, Lookout creates the file in the `Lookout` directory. You can create a diagnostic file for each serial port you have configured. If you use the same file name for each port, all messages are logged to the same file.
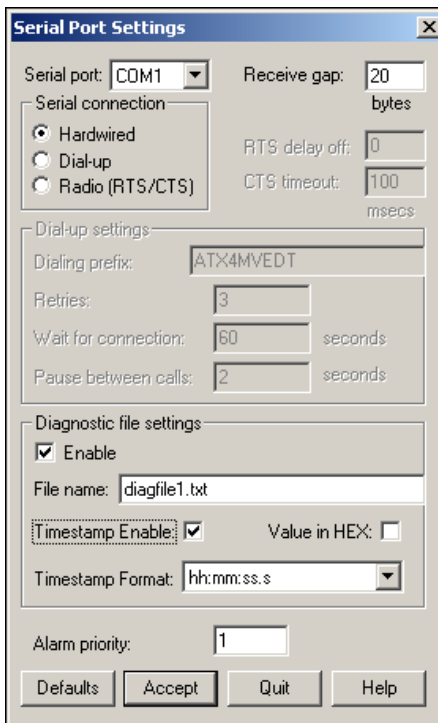
4. If you want timestamps for each communication, check the
    Timestamp Enable checkbox and choose a **Timestamp Format**.

5. To log all parts of a message in hexadecimal, select the **Value in HEX**
    checkbox. Otherwise, printable characters print as their character
    representation instead of in hex format. International characters are
    logged as two bytes (printable or non-printable).

The serial diagnostic file is a text file consisting of requests and replies.
Any byte that can be represented by a printable character is displayed
as the character, unless the **Value in HEX** checkbox was checked. All
non-printable bytes are represented by their hex codes enclosed in square
brackets. The following text shows how a typical request and reply might
appear.

```
Modbus1 ->
[01][03][00][00][00][02][C4][0B]
Modbus1 <-
[01][03][04][00]A[00][00][AA]'
```

In the first line, the Modbus1 object sent out (->) a message, the message shown in line two. The third line shows that Modbus1 received (<-) a message, the message in line four.

If timestamps are enabled for each message, you can match messages in the diagnostic file with alarms in the Alarms Window. A logged communication with timestamps enabled might look like the following:

```
09/27/2001 15:08:12 - Modbus1 ->
[01][03][00][0A][00][01][A4][08]
```

The diagnostic file might expose a missing reply from the hardware, as in the following example:

```
Modbus1 ->
[01][03][00][0A][00][01][A4][08]
Modbus1 ->
[01][03][00][0A][00][01][A4][08]
Modbus1 ->
[01][03][00][0A][00][01][A4][08]
```

5

Creating Lookout Processes

This chapter explains the basics of creating process files, creating objects, making connections, and creating user interfaces.

Creating a Process File

To create a Lookout process, select **File»New**. The following dialog box appears.
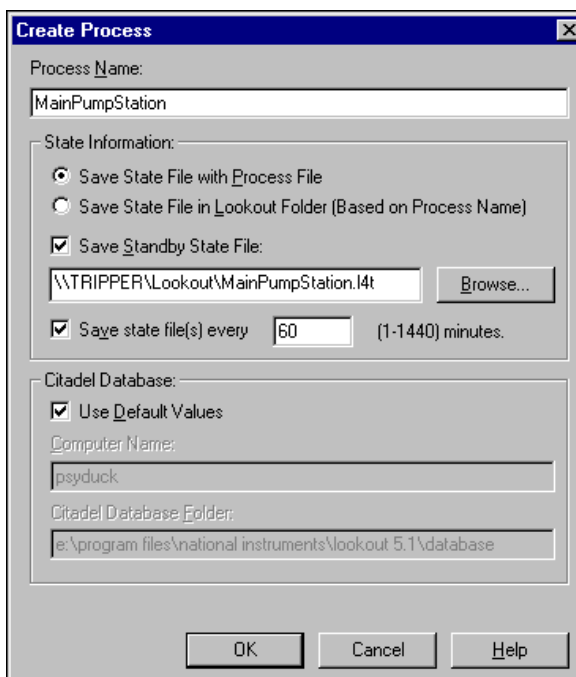
- **Process Name**—This name must follow the same conventions as a Lookout object name. Refer to the *Steps to Create an Object* section for a list of invalid names.

- The **State Information** section is where you specify the locations you want Lookout to save state files for this process.

    – **Save State File with Process File**—Saves the state file in the location where the process file was opened. This is the default behavior.

    – **Save State File in Lookout Folder**—Saves the state file in the Lookout folder of the copy of Lookout your are currently running. The state file name will be the same as the process name. This is useful if multiple computers are sharing a process file. In this case, each computer maintains its own state file locally.

    – **Save Standby State File**—Saves one or more extra copies of the state file in a location of your choosing. Enter a complete path, including state file name, to each location you want to save a state file. If you are saving the state file to more than one backup or alternative location, separate the paths with the vertical bar (|) symbol. For more information about using this setting in a redundant system, refer to Chapter 14, *Setting Up Redundancy*.

    – **Save state file(s) every—**Sets the frequency with which Lookout saves the state file. Remember that saving files uses system resources, so if you have a number of processes, each saving one or more state files fairly often, you may experience some performance problems. Uncheck this checkbox to prevent the state file from being updated. Refer to the *Opening a Process in a Known State* section of Chapter 13, *Networking and Deploying an Application*, for more information.

- The **Citadel Database** section is where you set the location of the Citadel database that Lookout logs data to for this process. Citadel settings you make when creating a process override any settings you may have made in the System Options dialog box.

    – **Use default values**—When enabled, Lookout uses the default location set in the System Options dialog box of any instance of Lookout running the process. If you enter a computer name and a path on that computer to a specific folder, Lookout logs data to that location on that computer, no matter what computer is running the process.

    – **Computer Name**—Designates a specific computer and path for your process to log data to. Enter the fully qualified network name for the target computer, without using international characters.

    – **Citadel Database Folder**—Enter the complete path to the target data directory, without using international characters.

Click **OK** when you are done. Select the options for your control panel in the next dialog box that appears. Refer to the *Creating Panels and Navigation* section for more information about control panels. Click **OK** when you are done.

Note   Consider screen resolution when creating display panels (VGA versus Super VGA). The same panel appears differently on computers using different resolution display drivers. Read about screen resolutions in the *Panel* topic of the *Lookout Help* before you design your panels.

Select **File»Save** to save your process to disk. Every time you save, Lookout updates the .l4p, .lka, .lks, and .l4t files. You can save a process file with any name you choose, including international characters if you want to use them. Your process file name does not have to be identical to your process name. You should save often while working and backing up your work to a separate disk, regularly, to avoid rework.

You can copy Lookout files by selecting **File»Save As** and choosing a different name or location for the process.

You can stay online while creating, updating, or editing your application. There is no need to shut down your process.

Creating Objects

The first objects you should create in your server process are driver objects that represent your control hardware. Driver objects often use your computer communication ports, so you may have to configure your ports for radio, dial-up modem, or hard-wired connections. Refer to Chapter 4, *Serial COM Port Communication*, for more information about configuring communications.

When you create a driver object, its data members represent the most common set of registers, bits, I/O signals, and other values that the physical device can transmit to or receive from Lookout.

Besides driver objects, the objects you choose to create for your process will depend on how you decide to implement your control logic, user interface, logging and reports, and so on.

The following activities may help you choose the objects you need in order to carry out the requirements of your system.

- In edit mode, select **Object»Create**. In the dialog box that appears, you can view the available objects by category such as Calculation, Control, Display, Timers, and so on. To familiarize yourself with the different objects and their functions, expand the view on the left side of this dialog box, click an object name, and read the description that appears to the right.

- Search the *Lookout Help* using keywords, to determine which objects you should use to accomplish a given task.

- Examine the Lookout examples to see how different control strategies can be implemented. Select **File»Open** and navigate to home2.14p. Use the Connection Browser to see how controls and objects are connected.

Steps to Create an Object

Complete the following steps to create a Lookout object.

1. Open or create a new process if one is not already open in Lookout. Make sure you are in edit mode. (Select **Edit»Edit Mode** or press <Ctrl-Shift> if necessary.)

2. There are two ways to create new objects in Lookout:

   - Select **Object»Object Explorer** from the menu to activate the Lookout Object Explorer. Right-click the process you want to create an object in and select **New Object**.

   - Select the **Object»Create** command from the Lookout menu bar.

   The Select object dialog box appears, as shown in the following illustration.
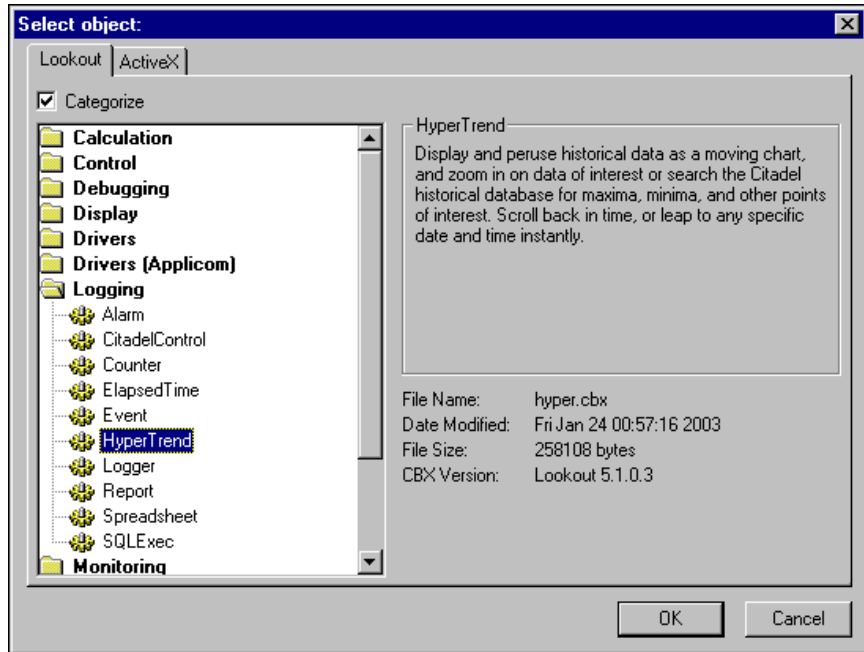
The first time you create a Lookout object, the Select object dialog box appears with the **Categorize** option checked. Some objects are contained in more than one category, depending on their versatility. To have access to all objects with this technique, disable the **Categorize** option by clicking the **Categorize** checkbox.

You create an ActiveX object on the ActiveX tab in basically the same way you create any other Lookout object. Refer to the *ActiveX* object description in the *Lookout Help* for details on ActiveX controls.

3. Select the object you want to create, and click **OK**. If you used the **Object** menu to create this new object, a dialog box appears for you to indicate where the new object should be created. Refer to the *Simplifying Your Folder Hierarchy* section of Chapter 2, *Developing an Application*, for tips on organizing your objects. Choose a location and click **OK**.

    A dialog box appears for you to define the parameters for the object.

4. Assign a unique name to the object. A name is how you identify a specific object. Lookout does not allow you to create or modify an object if the name is not unique or if it does not follow the proper naming convention. Rules for choosing names are as follows:

   • Names may include the characters A–Z, a–z, 0–9, and the underscore character (_).

   • Names must begin with a letter.

   • Names may be up to 32 characters long.

   • Spaces and tabs are not allowed.

   • Lookout recognizes uppercase and lowercase characters as the same character, so SW2 and sw2 are the same name.

   • Names cannot be Lookout keywords (object names and function names). For example, you cannot use terms such as Switch, DelayOn, STDEV, Modbus, nif, and Trend. You can, however, use Switch1, Switch2, and so on.

   • The following Windows color designations cannot be used as names: aqua, black, blue, blviolet, brown, dkblue, dkgreen, dkred, dkviolet, gray, green, hotpink, lime, ltblue, ltgray, ltgreen, ltviolet, ltyellow, magenta, orange, pastelblue, pastelgreen, pastelviolet, pink, red, skyblue, white, yellow.

   • The following terms cannot be used because of their internal use in Lookout: abs, acos, alias, and, asi, atan, atan2, avg, choose, cos, exact, exp, fact, find, fixed, folder, gchoose, getbit, getword, gif, if, int, lchoose, left, len, lif, ln, log, log10, lower, max, mid, min, mod, nchoose, new, nif, not, now, or, pi, product, proper, qbad, qget, qgood, qset, qtext, rand, replace, rept, right, round, search, sign, sin, sqrt, stdev, stdevp, substitute, sum, symlink, tan, tchoose, text, tif, today, trim, trunc, upper, var, varp, xor.

5. Define the object parameters. Each object has unique parameters, described in the *Lookout Help*.

   Ordinary parameters that can only be set in the object definition dialog box are indicated by white fields. Some parameters, however, allow *expressions*, as denoted by a yellow field. Expressions let you set a parameter value programmatically by connecting the parameter to a data member of another object. You can also manipulate data using the expression functions. Refer to Chapter 7, *Expressions*, for more information about expressions and expression functions.

6. Click **OK** to create the new object.

   If you create a **Display** object using the **Object** menu, the display properties dialog box appears as soon as you click **OK**, and when you

click **OK** again, the display appears on the control panel. If you create any other type of object, or you create a **Display** object using the Object Explorer, the new object appears in the Object Explorer but is not displayed on any control panel.

7. (Optional) To create a display of a control object, you can drag and drop it from Object Explorer to a control panel. You can create a control for a data member by right-click-dragging that data member from Object Explorer to a control panel, which creates the control with a remote source connection to the data member. Refer to the *Remote Source Controls* section for more information about this type of connection.

Choosing a Source Position for Control Objects

The Pot, Switch, RadioButtons, Pushbutton, and TextEntry controls have two different value source options: local and remote. You set this option when you create the control, in the **Position source** field of the dialog box (**Text source** for the TextEntry control).

For more information about the interactions of source positions and connection types, refer to the *Connecting Objects* section later in this chapter.

Local Source Controls

Local source controls can make only one-way connections, writing to the objects they are connected to. A control with the local option selected takes and maintains its value by being operated or by having its value data member written to. When you select local source, your control keeps the value you set, whether by turning a knob, flipping a switch, or entering text in the TextEntry field. The only way for the value of the control to change is by operating the control (unless you have converted the control to an indicator by connecting to its value data member). If the Lookout process is shut down and reopened, the initial value for a local source control is taken from the Lookout state file. A control configured with the local source option can be connected to another data member only with a direct connection.

Remote Source Controls

Remote source connections make two-way connections; they write to the object that they are connected to, but also adjust to reflect changes in that object that happen through some other means. A control configured for remote source takes its value from the data member it is connected to.

If something changes the value of the data member it is connected to, such as someone opening a valve manually in the field, the control will change to reflect the new hardware setting. If the Lookout process is shut down and reopened, the initial value for a remote source control is taken from the data member it is connected to, not from the Lookout state file. The two-way flow of data in this sort of connection requires a remote connection instead of a direct connection.

✏️ **Note**   While it is possible to make a remote source connection to write-only as well as to read/write data members, you must connect to a read/write data member in order for the two-way flow of data to function correctly. For that reason, if you make a remote connection to another Lookout control, you must connect to the value data member, not the (implicit) data member.

You cannot incorporate an expression into a remote source connection. If you need to modify the flow of data through such a connection, use a dynamic Symbolic Link as described in the *Symbolic Link Connections* section later in this chapter.

The hardware refresh rate can be an important consideration when using remote source connections with a control. For more information, refer to the *Lookout Control Sources and Refresh Value Operations* section of the *Lookout Release Notes*.

Setting the SnapDelay for Remote Source Controls

There is a delay built into Lookout between the time when a remote source control requests a change for a data member and when the control adjusts its own value to reflect that data member. Without this delay, any hardware operated by a remote source control might not have enough time to respond to the change before the control attempts to confirm the change. If the value read by the remote source control does not change within that delay time, the control snaps back to the last value reported by the hardware. In some cases, the control may snap back to the last value reported by hardware, then immediately jump to the new value when the hardware finally confirms the change. You can prevent this erratic behavior by adjusting the delay time for individual controls using the SnapDelay data member. For example, a dial-up connection may require a SnapDelay setting of 30 seconds or more. Refer to the control object's topic in the *Lookout Help* for more information on setting the SnapDelay data member.

Copying and Pasting Objects

Once you have made a Lookout object, you can copy it by right-clicking it in the Object Explorer, choosing **Copy**, then right-clicking again within the same process and choosing **Paste**. The pasted object has the original object's parameters, but not the original object's connections. You cannot copy or cut an object and paste it into another process.

Modifying an Object's Parameters or Display Properties

To modify an object's parameters, select **Object»Modify**, navigate to the object, and make the desired changes. Alternatively, you can right-click an object in Object Explorer and choose **Properties**. After you place a display on a Lookout panel, you can right-click it to access its parameters, change its display, or perform other editing tasks. You can select a group of controls, right-click them, and align, space, group, and ungroup them.

Using Custom Objects

If you plan to develop and reuse a set of objects in multiple processes, consider saving that set of objects as an Aggregate. If you require functionality that is not provided by any of the objects that ship with Lookout, you might want to create your own custom ActiveX object.

Aggregate Objects

Aggregates are useful if you have to create a number of identical or highly similar collections of connected objects. With Aggregates, you can use a Lookout process as a module in building other Lookout processes. You can create a Lookout process that contains all the elements you need to use repeatedly, and save that process as an *Aggregate definition file*. You can then use this Aggregate definition file when you create an Aggregate object within another process. Refer to the *Aggregate* topic in the *Lookout Help* for more information about designing and using Aggregates.

ActiveX Objects

An ActiveX control is a type of software object that can be used in developing applications or run inside of other applications. Lookout installs a set of National Instruments ActiveX objects that you can use in your process.

To create an ActiveX object in you Lookout process, select **Object»Create** then click the **ActiveX** tab. ActiveX controls that have been tested for use with Lookout are listed in the **Recommended** category. You may have a large number of ActiveX objects installed on your computer; many are installed by various software packages and a number come as a part of Microsoft Windows. These ActiveX controls appear in the **Other** category. The **Other** controls may or may not work satisfactorily with Lookout, and you should proceed cautiously in incorporating them into your processes.

The Lookout Object Developer's Toolkit contains an ActiveX Wrapper Toolkit that you can use to create ActiveX controls for use with Lookout. Contact National Instruments for more information about the Lookout Object Developer's Toolkit.

Editing Data Member Configuration

When you create an object, Lookout create a set of data members for that object. The online help for each object includes a list of these data members and information about possible values.

You can modify the configuration for any data member or create an alias for any data member. Data member configuration includes such things as alarm setpoints, logging deadband, deviation, scaling, historical logging, and alias names.

1. There are several ways to access the dialog box used to edit data member configuration.

    •   Select **Object»Edit Data Member Configuration**. In the dialog box that appears, choose the object whose data members you want to configure.

    •   In Object Explorer, right-click the object whose data members you want to configure, and select **Edit Data Member Configuration**. You can also right-click any writable data member to preselect it for editing.

   A dialog box similar to the following appears.

Note   Fields in this dialog box may vary, depending on the type of data member you select.
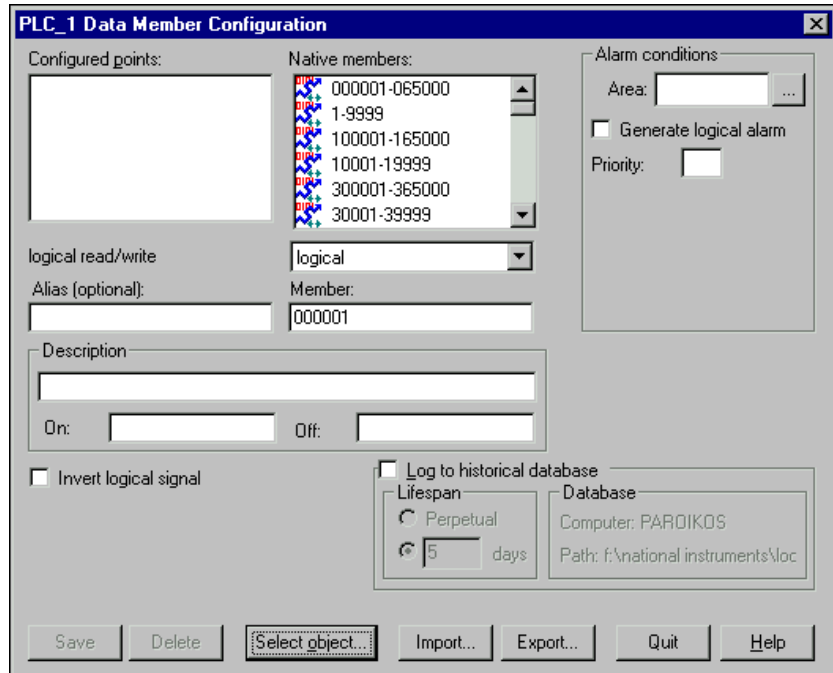
2. Follow these steps for each data member to be configured:

    a. Click the data member to edit in either the **Native members** or **Configured points** field. The data member appears in the **Member** field.

    b. Configure appropriate values for the data member. Refer to the following sections for each data member type, for more information about the configuration options.

    c. Click the **Save/Update** button. Lookout stores all the new configuration for the specified data member and adds the modified data member to the **Configured points** field. The changes take effect immediately.

3. Click **Quit** to exit the dialog box.

Editing Numeric Data Members

The following illustration and paragraphs describe numeric data member configuration options.
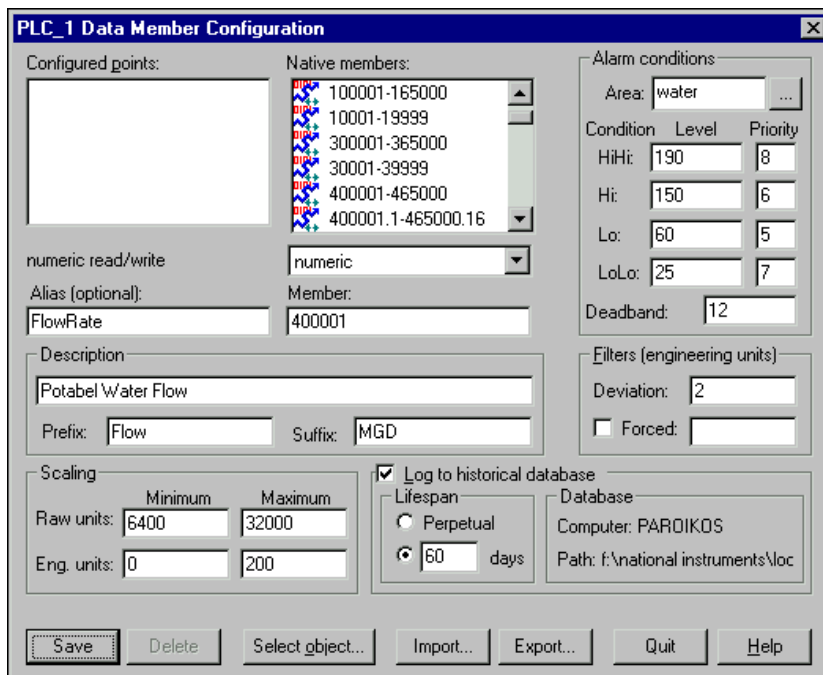
- **Alias (optional)**—If you enter an alias name for a data member and click **Save**, a new instance of that data member is created for the object. You can then configure that alias data member with a configuration different than the original data member from which the alias was created. For example, you could create an alias for the Modbus object's 40011 data member and name it FlowRate. Then, both 40011 and Flowrate are available as you develop your process, and you can configure the two data members differently, although they represent the same incoming value from your Modbus device.

  Aliasing data members is a good way to insulate your Lookout process from changes in your control hardware configuration. For example, suppose you create an alias called Flowrate for the Modbus 40011 data member, which represents a physical connection to a flow transmitter. You reference Flowrate multiple times throughout your process, but later the flow transmitter is rewired to the 40012 input. In that case, you need only modify the FlowRate data member to alias

40012 instead of 40011. Lookout instantly reflects this change everywhere FlowRate is used.

If you attempt to modify an existing alias name, the **Update** button changes to a **Save** button and you will only create a new alias.

**Note** You cannot use international character sets when creating alias names.

Note An alias is optional in some instances. You can apply alarming and logging parameters to any native data member and save it to the **Configured points** list without giving it an alias name. Scaling, however, operates somewhat differently. In order to read scaled data, you read it from the alias, not the native data member.

- **Description**—The description you enter here appears as message text in the Alarms Window. You do not have to use quotes in this field.
- **Prefix** and **Suffix**—These are part of the description, but do not appear in the Alarms Window.
- **Scaling**—Define linear scaling by entering **Raw units** and **Eng. units**. The raw numeric data member is converted (scaled) to an engineering unit value. In the previous illustration for example, the control hardware generates a raw value ranging from 6,400 to 32,000. Lookout will convert that raw signal linearly to range from 0 mgd to 200 mgd. Refer to your hardware specifications and calibration records for the minimum and maximum raw units associated with analog devices. If you leave the **Raw units** and **Eng. units** fields blank, Lookout performs no scaling on the signal.

    In order to scale this data, you can use an alias in combination with the native data member. Read or write the raw units by connecting to the native data member, and read or write out the engineering units from the aliased value.

- **Deviation**—This setting filters out insignificant variations of numeric signals. The following figure shows two values plotted on a trend. One line is the raw unfiltered value. The other, stair-stepped line, represents the filtered value after passing through a **Deviation** of 2.
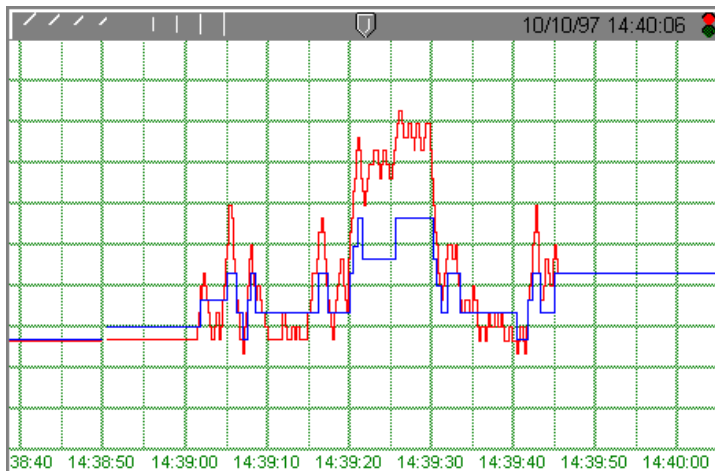
Citadel uses this **Deviation** when logging historical data. Refer to Chapter 8, *Logging Data and Events*, for more information on logging historical data.

• **Forced**—Use this field to manually enter a constant value for the data member. When you select the **Forced** checkbox and enter a value in the field, Lookout forces the engineering unit value to be equal to the value you entered, regardless of the actual value of the native data member. You might use this when a sensor fails or during sensor maintenance, or any time a PLC receives a bad signal from the transmitter.

• **Alarm conditions**—Use these parameters to define alarm limits and their associated priorities. Lookout compares the alarm setpoints to the engineering units value (that is, the post-scaled, post-filtered number). If you do not enter scaling parameters, Lookout applies the alarm parameters directly to the raw signal. Lookout generates an alarm the moment a value violates the **Hi** or **Lo** alarm setpoints. The alarm returns to normal when the value drops below the high alarm setpoint (minus the Deadband, if one is entered), or goes above the low alarm setpoint (plus the Deadband).

You can assign a data member to any existing alarm **Area** or you can create a new **Area** by entering the new area name in the field.

Use the alarm **Deadband** to prevent fluttering between alarm and normal states when the signal value hovers near an alarm limit. The Deadband applies to both **Hi** and **Lo** alarm setpoints, as wells as to **HiHi** and **LoLo** alarm setpoints.

Refer to Chapter 9, *Alarms and Events*, for more information on alarms.

- **Log to historical database**—These parameters define how long to store a value in the Citadel database. If you do not select this option, Lookout does not log the value. If any scaling or filtering parameters are defined, Lookout logs the scaled, filtered value (that is, the engineering unit value). When you log data, it is logged to the directory specified in the Create/Modify Process dialog box.

Tip If you intend to display data with a Lookout HyperTrend object, you must log that data.

Refer to Chapter 8, *Logging Data and Events*, for more information on logging historical data.

Editing Logical Data Members

With a few exceptions, logical data member configuration options are the same as those for numeric data members. Scaling of a logical signal consists of the **Invert Logical Signal** checkbox. When you choose this checkbox, an ON value is represented by an OFF value, and so on. When you do not select it, ON is ON and OFF is OFF.

A logical data member configuration is shown in the following figure.

Besides the alarm **Area** field, a logical data member include the **Generate Logical Alarm** checkbox. When this checkbox is selected, the data member generates an alarm whenever the value is ON and clears whenever the value is OFF. If the **Invert Logical Signal** checkbox is selected, the value used for alarming is the inverted value. Refer to Chapter 9, *Alarms and Events*, for more information about alarms.

Editing Text Data Members

Text data member configuration includes only **Alias**, **Member**, and **Description** fields. You can log text data to Citadel, but the only way to query for the logged text is though the Lookout SQLExec or DataTable objects.

Note   You cannot use international character sets when creating alias names.

Data Type Matching and Conversion

Lookout data is not strongly typed, which means that it does not need to be interpreted as the type it was originally cast in. *Data polymorphism* in Lookout means that data of one type is interpreted appropriately when connected to an input of another type. You can select the data type you want an expression to be displayed as in the **Display Type** field, shown in the following dialog box.
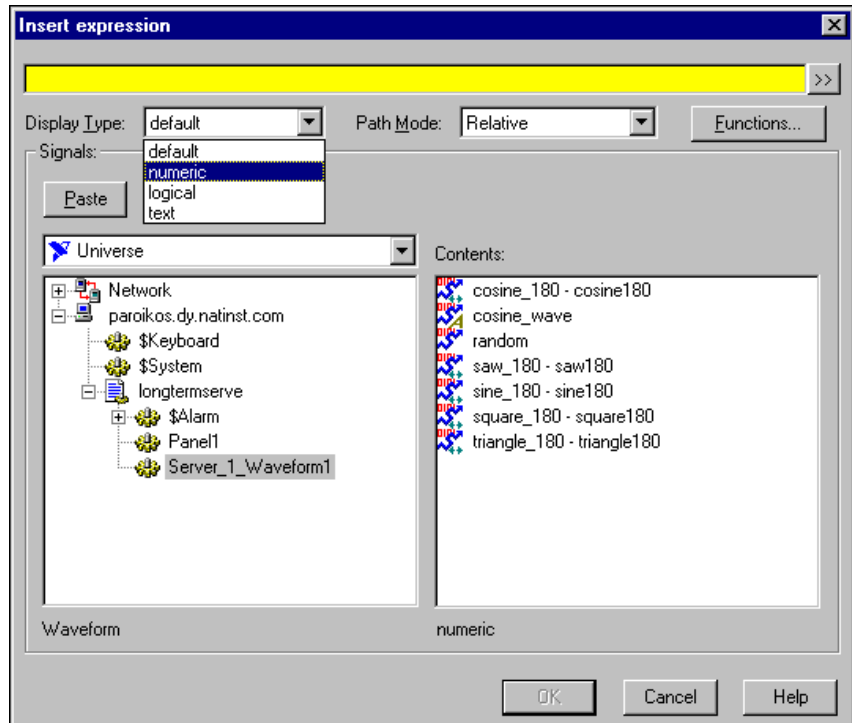
The following table describes how data types are used by inputs of a different type.

Table 5-1. Data Type Conversion

Data Type	Interpreted as Numeric	Interpreted as Text	Interpreted as Logical
Logical	0 or 1	ON or OFF	—
Numeric	—	digits of the number	0 displays as OFF. Any value other than 0 displays as ON.
Text	Appear as digits if the string consists only of digits in a valid Lookout display format, such as a decimal number or a scientific expression. In a time format, such as 10:05:30, the text is interpreted as a number (the fraction of one day represented by the time quantity) in scientific notation. If the text string does not consist of digits in a valid Lookout format, the text is interpreted as a 0, with the exception of ON or TRUE, which display as 1. The text strings OFF or FALSE are interpreted as 0, by default.	—	Interpreted as a 0 or as OFF, except when the text string consists of a 1, On, Yes, or True, all of which are interpreted as ON. By default, the text strings 0, Off, or False are interpreted as OFF.

Polymorphic data types affect the DataTable object. You must input numeric data to DataTable numeric data members, logical data to DataTable logical data members, and so on.

Importing and Exporting Data Member Configuration

You may want to import or export data member configuration for the following reasons:

- Export object data member definitions to Excel for the purpose of documentation.

- Export an object's data member configuration to Excel, perform global replacements on data member configuration settings, and then import the changes.

- Create a name list in Excel or in an application that exports to Excel, then copy that name list into Lookout.

- In a process using multiple duplicate driver objects (such as a gas pipeline or water distribution system), define the data member configuration settings in Excel. Then import that configuration for multiple driver objects.

Although you can import and export any object's data member configuration, you may find that it is most useful for objects with many data members, such as driver objects or the DataTable object.

Exporting Data Member Configuration

Follow these steps to export the data member configuration for an object's data members. Lookout will not export every possible data member (many driver objects have a capacity for thousands of members). Instead, Lookout exports configured points and data members that are connected to other objects.

1. Access the dialog box used to import/export data member configuration.

    - Select **Object»Edit Data Member Configuration**. In the dialog box that appears, choose the object whose data member configuration you want to export.

    - In Object Explorer, right-click the object whose data member configuration you want to export, and select **Edit Data Member Configuration**.

    A dialog box similar to the following appears. The following illustration shows a Modbus object that already has a number of configured points.
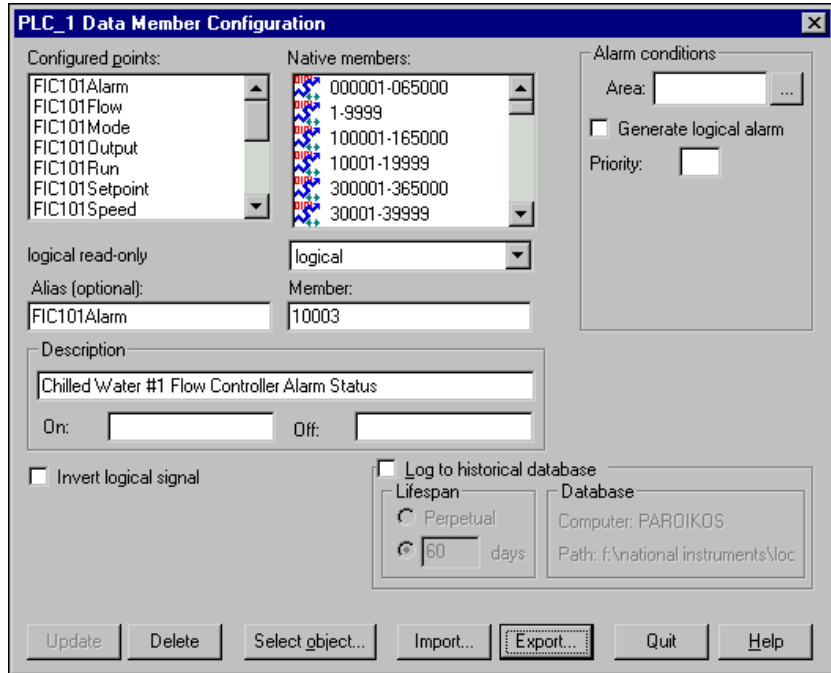
2. Click **Export** to export the object's data member configuration for **Configured points** and data members that are connected to other objects.

3. In the dialog box that appears, choose a directory path, enter a file name and click **Save**. The configuration information is exported to an Excel file. An example of the `.xls` file is shown in the following illustration.

Creating a Data Member Configuration Spreadsheet

The following steps explain how to create an Excel spreadsheet file for use in editing data member configuration and importing it into Lookout. This method can be much faster than working in the Lookout dialog boxes one at a time.

1. The easiest way to create a data member configuration spreadsheet for an object is to first export the configuration from Lookout as follows.

   a. Create the object in Lookout.

   b. Add one logical data member, one numeric data member, and one text member (if necessary) to the **Configured points** for that object. Refer to the *Editing Data Member Configuration* section for more information about creating configured points.

   c. Click **Export** to export to create a spreadsheet file.

   This creates a basic spreadsheet file with the **Configured points** and column labels for all possible data member configuration settings. Each row represent a data member.

2. Modify or copy the spreadsheet as needed, using the convenient tools of your spreadsheet program. You can add rows to define new data members, modify the rows to change data member configuration, and so on. Follow these guidelines to ensure that your spreadsheet will import successfully.

   • If you create or change the column labels manually, make sure to spell all column labels correctly. Lookout ignores white space and is case-insensitive. The following list contains all possible column labels, in the order in which Lookout inserts them when creating a spreadsheet file: Command, Alias, Member, Description, Prefix, Suffix, Eng min, Eng max, Raw min, Raw max, Invert?, Deviation, Force?, Forced value, Alarm area, Lolo level, Lolo priority, Lo level, Lo priority, Hi level, Hi priority, Hihi level, Hihi priority, Logical priority, Log data?, Lifespan.

   • Lookout requires the column labels Command and Member. All other column labels are optional. Command must be in cell A1. The order in which the other column labels appear makes no difference.

   • Each row in the Command column should contain either the keyword `insert` or the keyword `delete`. When you import a data member configuration file, Lookout adds information from rows with `insert` in the Command column, removes the information relevant to rows with `delete` in the Command column, and ignores rows that do not have either keyword.

> To determine exactly what to delete, Lookout uses the data member alias name; or if the record does not have an alias name, it matches the record member name.

- • Any time you import a data member configuration, Lookout reads the first 30 columns (A–AD) and ignores columns that do not have labels.

3. After you have edited and expanded your spreadsheet file, save it by selecting **File»Save**. If you are working with a version of Excel more recent that 4.0, the program asks you if you want to update your spreadsheet to a newer format, as shown in the following illustration. *Be sure to select* **No**. Lookout does not currently accept Excel spreadsheet files from versions later than 4.0.
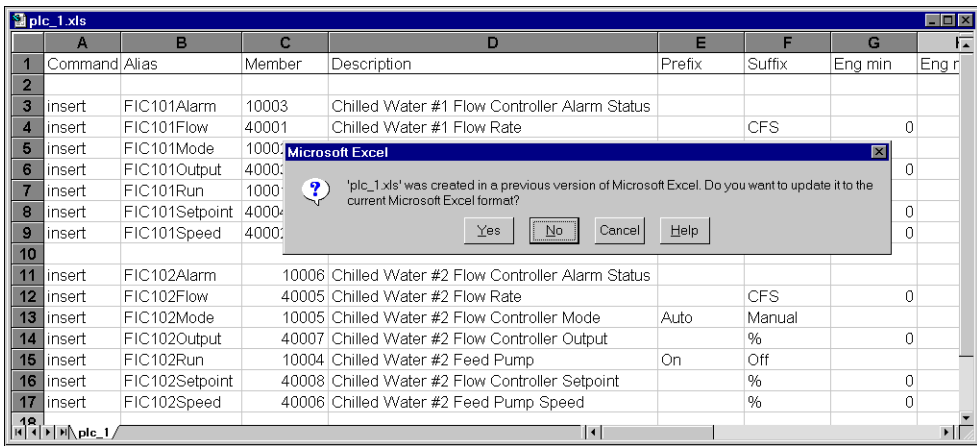
4. Import the new configuration values into Lookout. Refer to the next section for more information.
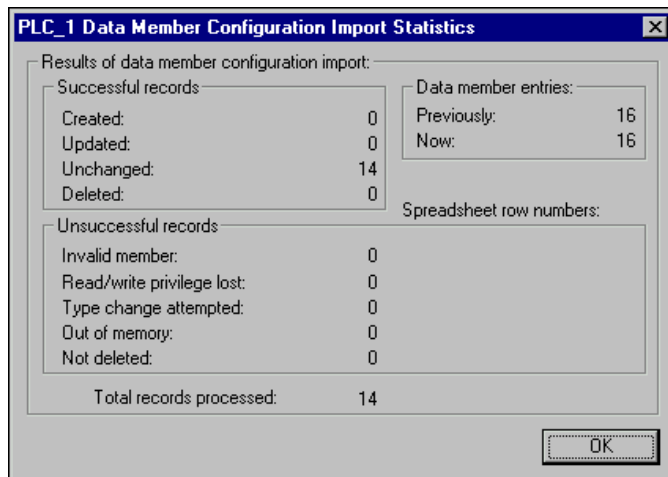
Importing Data Member Configuration

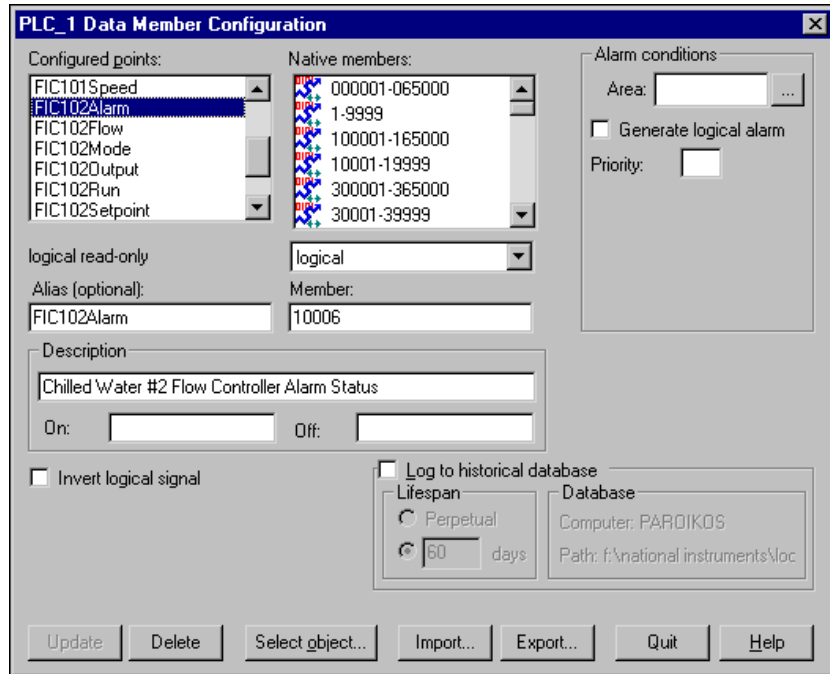Follow these steps to import data member configuration settings from an Excel 4.0 spreadsheet file.

1. Create a spreadsheet file, one per object, with the data member configuration you want to import into Lookout. Refer to the *Creating a Data Member Configuration Spreadsheet* section for more information about creating this file.

2. In Lookout access the dialog box used to import/export data member configuration.

    •   Select **Object»Edit Data Member Configuration**. In the dialog box that appears, choose the object whose data member configuration you want to import.

    •   In Object Explorer, right-click the object whose data member configuration you want to import, and select **Edit Data Member Configuration**.

3. In the dialog box that appears, click **Import**.

4. In the next dialog box that appears, choose a directory path, select the file name, and click **Open**.

    Lookout imports the data member configuration as detailed in the *Creating a Data Member Configuration Spreadsheet* section, adding the imported data members to the object's **Configured points**. When finished, Lookout displays the import statistics, as shown in the following illustration.

The following illustration shows the data member configuration from a spreadsheet successfully added to an object's **Configured points** list.

PLC_1 Data Member Configuration

Configured points:
- FIC101Speed
- FIC102Alarm
- FIC102Flow
- FIC102Mode
- FIC102Output
- FIC102Run
- FIC102Setpoint

Native members:
- 000001-065000
- 1-9999
- 100001-165000
- 10001-19999
- 300001-365000
- 30001-39999

Alarm conditions

Area: [   ] [...]

☐ Generate logical alarm

Priority: [   ]

logical read-only   logical

Alias (optional):   Member:

FIC102Alarm   10006

Description

Chilled Water #2 Flow Controller Alarm Status

On: [   ]   Off: [   ]

☐ Invert logical signal   ☐ Log to historical database

Lifespan
○ Perpetual
◉ 60   days

Database
Computer: PAROIKOS
Path: f:\national instruments\loc

[Update] [Delete]   [Select object...] [Import...] [Export...]   [Quit] [Help]

5.   Click **Update**.

Connecting Objects

In Lookout, you connect objects to pass control signals between them. You can connect one data member to another data member, a data member to an object parameter, or a data member to an (expression) object.

You can make the following kinds of connections in Lookout:

- Direct—Made with the Edit Connections dialog box, these connections work in one direction only. Direct connections should only be made from a server process to control hardware, or between objects contained within the same process. Direct connections can contain expressions. With expressions, you can use variables and logic tests to respond dynamically to changing needs and circumstances.

- Remote—This type of connection is used in Symbolic Links and consists of a fully-specified path, a URL. It is a two-way connection that requires both read access and write access on both ends of the connection. You cannot use an expression in a remote connection.

💡 **Tip**   Remember, when connecting two objects using the Edit Connections dialog box, always select the object you want to write to.

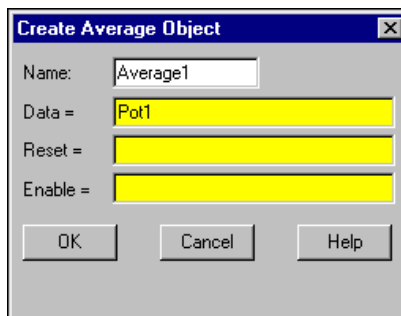## Connecting Data Members to Parameters

You connect object data members to object parameters by entering a data member name in a parameter field. If the parameter accepts an expression, the data member can be part of an expression.

1. When you create (**Object»Create**) or modify (**Object»Modify**) an object, you can set up connections to its parameters. You can reach the same dialog box by right-clicking an object in Object Explorer and choosing **Properties**. The fields that appear in the resulting dialog box are parameters for that object. If a field is yellow, you can use an expression in it, including a data member reference.

2. In the yellow expression field for the parameter you want to connect, enter the data member to connect to in the form *objectname.datamember*. Alternatively, you can right-click in the yellow field to launch the **Expression Editor**, navigate to the object data member you want, then click **Paste** and **OK**.

3. Click **OK**.

⚠ **Caution**   Because (implicit) data members are read-only, you cannot write to them or make a remote connection to them.

The following figure shows the **Data** parameter of an Average object connected to the (implicit) data member of an object called Pot1. In this case, the Average object uses the value of Pot1 as its **Data** input.
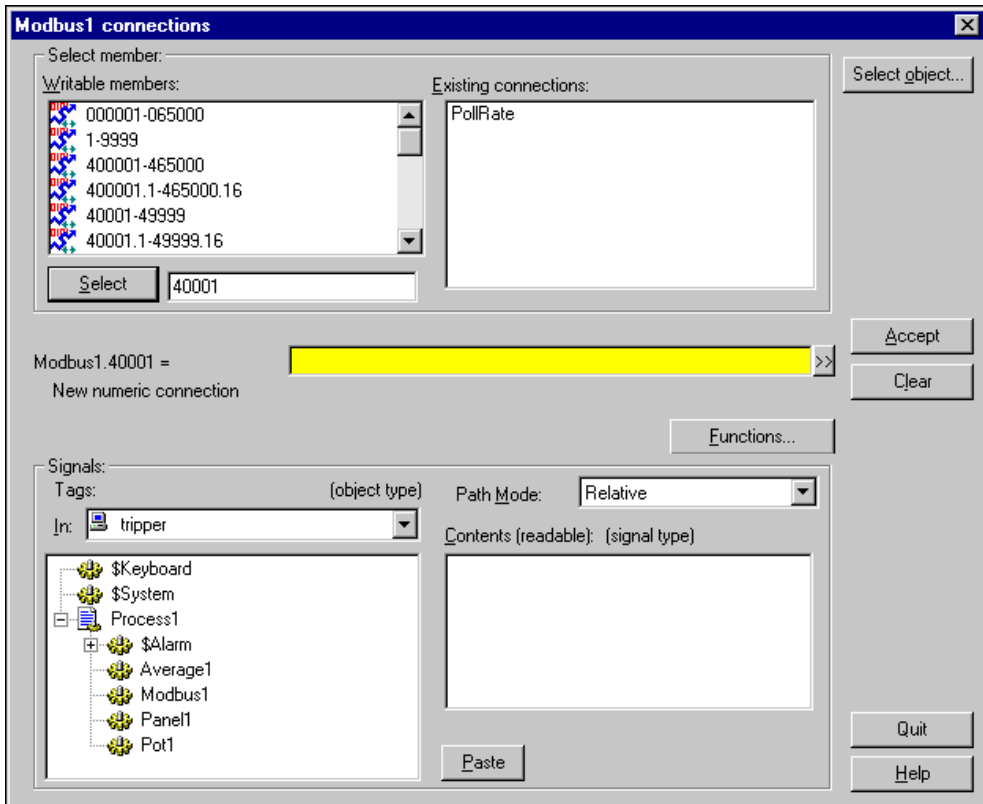
Connecting Data Members to Data Members

Complete the follow ing steps to connect a data member to a data member.

1. You can access the connections dialog box in two ways.

   • Select **Object»Edit Connections** and choose the object which will be written to by the connection.

   • In Object Explorer, right-click the object which will be written to by the connection, and choose **Edit Connections**.

A dialog box similar to the following appears.

The top part of this dialog box contains information on the object you are writing to. The lower part of the dialog box lets you navigate to the other data member you want to connect to.

2. To make a new connection, click a writable data member (or range of data members) from the **Writable members** list and click **Select**, or double-click a writable member. To edit an existing connection, click a data member in the **Existing connections** field.

    You can manually enter or edit a data member number in the selection box, for example to select a specific data member (233) from a range of data members (1–9999).

    The selected member appears to the left of the yellow expression field in the center of the dialog box.

3. In the **Signals** section of the dialog box, make sure that **Tags In** and **Path Mode** are set appropriately. For more information about path modes, refer to the *Path Modes in Lookout* section later in this chapter.

4. In the bottom left field, navigate to the object that contains the other data member you want to connect to.

5. In the bottom right **Contents** field, select the appropriate data member by clicking it.

6. Click **Paste**. The complete connection is displayed in the center of the dialog box: the data member in the yellow field will be writing to the data member on the left. Notice that the data type of the data members you have selected are displayed, to help you make sure you are connecting the same types of data members.

7. Click **Accept** to confirm the connection. You should see a message confirming the action under the yellow field. You will also see the data member on the left side of the connection equation added to the **Existing connections** list box.

8. Repeat these steps if you want to connect any other data members of this same object. You can click **Select object** to select from another object's data members.

9. Click **Quit** when you are done.

Making Remote Connections

A remote connection cannot contain any condition checking or mathematical operations. It is designed to permit a two-way flow of data through the connection. For example, a control connected to a PLC register using a remote connection changes the register value when the control is adjusted. Likewise, if the register value changes because of some other cause, the control changes to match the new register value. Remote connections are the preferred method for connecting a control in a client process to a server process in Lookout.

While you can make a remote connection to data members that are writable, or both readable and writable, you must connect to read/write data members in order for a two-way remote connection to function correctly.

⚠ **Caution** Because (implicit) data members are read-only, you cannot write to them or make a remote connection to them.

URLs in Connections

A URL (Uniform Resource Locator) in Lookout is essentially a path to any given data member, object, process folder, or process. Remote connections, Symbolic Links, and HyperTrends require URLs, as indicated by a green field in a dialog box. You can edit a URL by right-clicking in the green field, which opens the Lookout URL Editor dialog box, shown in the following illustration.
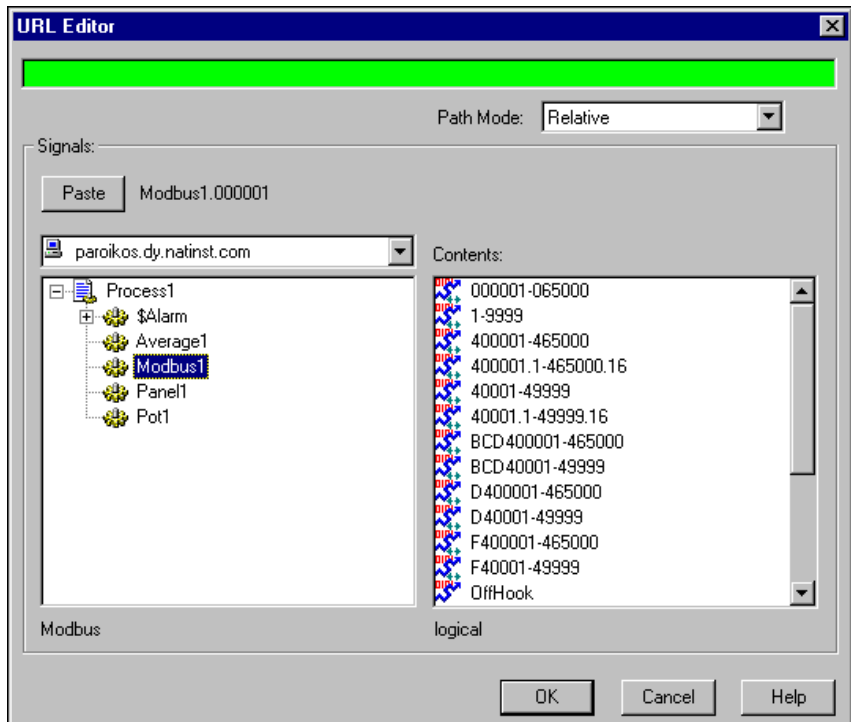
You can use the tree view in the left window to locate the object you want, and select the data member in the right window. Click the **Paste** button to insert the complete string into the URL field.

You control the path relativity of your URL with the **Path Mode** selection box. Refer to the *Path Modes in Lookout* section for more information on using the **Path Mode** selection.

Interaction of Remote and Direct Connections to the Same Data Member

If you create a remote connection from a control to a data member that already has a direct connection made to it, Lookout generates an alarm that the data member you are connecting to is not writable. Your remote-connection control will then function as an indicator instead of a control.

If you make a direct connection from a local source control to a data member to which you have previously made a remote connection from another control, the new direct connection takes precedence. Lookout generates an alarm that the data member you connected to is no longer writable for the remote connection. Your remote-connection control will then function as an indicator instead of a control.

HyperTrend Item Connections

Because the HyperTrend object trends data from the Citadel database, every item you display on a HyperTrend is accessed through a remote connection. If you need to trend an expression, you can create an (expression) object for the expression you want to trend, log it to Citadel, and then display it as a HyperTrend item.

Connecting Clients to a Server

Consider these general guidelines when connecting clients to a server process.

- Never make any connection *from* a server process *to* a client process. Connections between clients and servers should be remote connections that are created on the client side. Refer to the *Remote Connections from Clients to Server* section for more information.

- Avoid using control loops and process control logic in a client process. These elements belong in a server process, and clients can to connect to them remotely. Refer to the *Connections from Clients to Server Control Logic* section for more information.

- Use Symbolic Link connections when possible. If you make your client control connections through a Symbolic Link rather than to the server itself, then you can make any necessary path correction in the

Symbolic Link, once, rather than in a larger number of client objects. Refer to the *Symbolic Link Connections* section for more information.

Remote Connections from Clients to Server

All connections from a client to a server should be made so that changes in the server are read back to client controls properly. This requires the sort of two-way data exchange created by a remote source connection. For example, assume you have a Modbus object (Mod1) in the server process and a Switch (Switch1) in the client process. Now suppose you use the Edit connections dialog box to make the following connection in the server, as shown in the top of Figure 5-1:

```
Mod1.1 = \\Client1\ClientProcess\Switch1.value
```

You may intend for this connection to make the `Mod1.1` data member controllable from the Client1 process, but this method is not advisable for the following reasons:

- In this scenario, you could not use a switch in any other client process to control `Mod1.1`, as shown in Figure 5-1. The client switches would not be synchronized.

- In this scenario, you could not stop the Client1 process without causing problems. Server processes should run continuously, and the connection in Figure 5-1 forces the Server process to rely on the Client1 process for its `Mod1.1` value.

The correct design makes a remote source connection from the switch in either client process to `Mod1.1` in the Server process, as shown in the bottom of Figure 5-1.
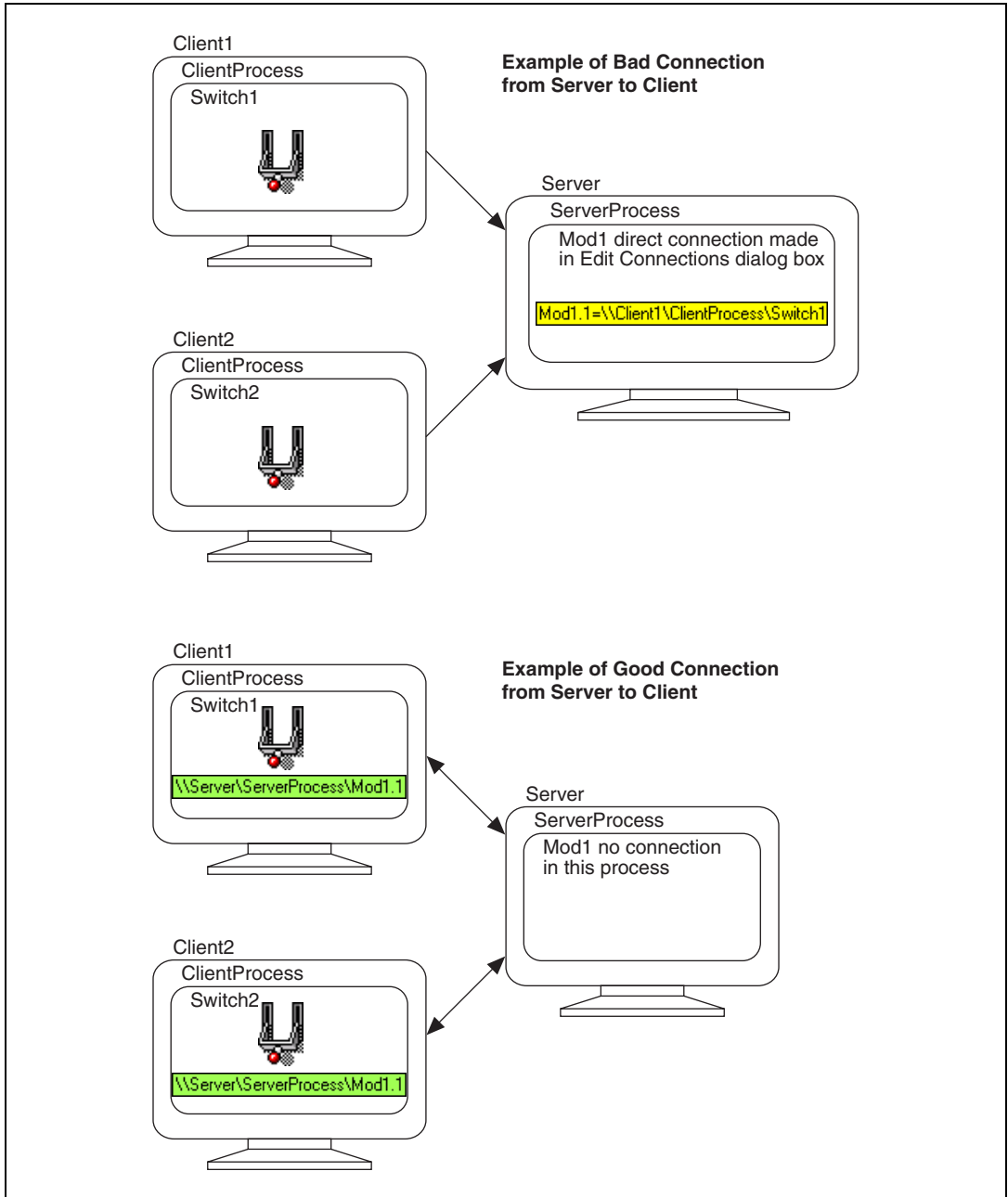
Figure 5-1. Examples of Bad and Good Connection from Server to Client

Connections from Clients to Server Control Logic

Server processes should implement control logic, while client processes use remote connections as necessary. For example, assume you have a Modbus object (Mod1) in the server process, and in the client process you have two Pots (Pot1 and Pot2) and a switch (Switch1) that selects between Pot1 and Pot2. As shown in the top of Figure 5-2, a third Pot object (Pot3) in the client process makes a remote connection to Mod1.1 in the server process, using the Edit connections dialog box.

You may intend for Pot3 to take the value of either Pot1 or Pot2, depending on the position of Switch1, then connect that value to Mod1.40001 in the server process, but this method is not advisable for the following reasons:

- In this scenario, you could not use a Pot in any other client process to control Mod1.40001. The client Pots would not be synchronized.

- In this scenario, you are attempting to write values to Pot3 from two different sources: Mod1.40001 and the If(Switch1, Pot1, Pot2) expression. You would get an alarm stating **Remote position source is not writable** because it is not clear which source Lookout should use for Pot3.

The correct design implements the control logic within the server process, with remote connections from the client, as shown in the bottom of Figure 5-2.
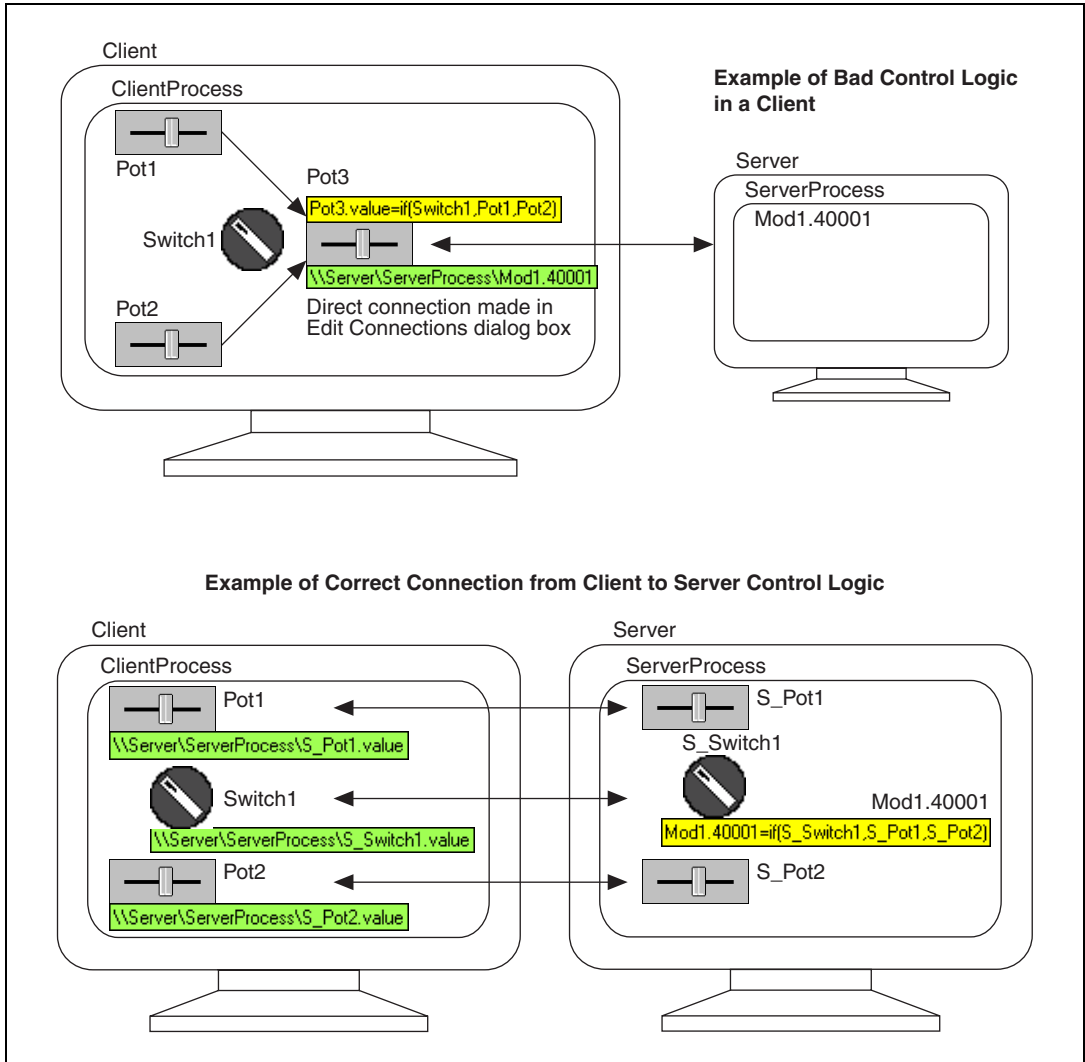
Figure 5-2. Example of Bad and Good Control Logic in a Client

Symbolic Link Connections

Lookout Symbolic Link objects have several important uses in Lookout. They are the best way to connect a client to a server, they are necessary to implement redundancy, you can use them to multiplex panels, and you can use them in conjunction with dynamic links to connect to complex expressions.

Once you create a Symbolic Link object, it can be accessed only from within the process where it exists. You cannot reference it from another process.

Symbolic Links with their **Source** set to **Static** can only refer to a single process, folder, or object. Symbolic Links with their **Source** set to **Dynamic** can incorporate complex expressions and are used for redundancy. The following sections describe both in detail.

Refer to the *(Symbolic Link)* topic in *Lookout Help* for more information about using this powerful Lookout object.

Static Symbolic Links

You can use a Symbolic Link with its **Source** set to **Static** to connect a client to a server process. The following illustration shows a Symbolic Link created using a static source, designating a basic server process.
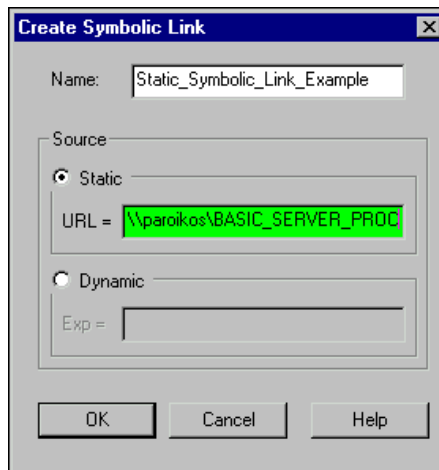
This Symbolic Link allows access to all the objects and their data members contained in the BASIC_SERVER_PROCESS, as shown in the following illustration.
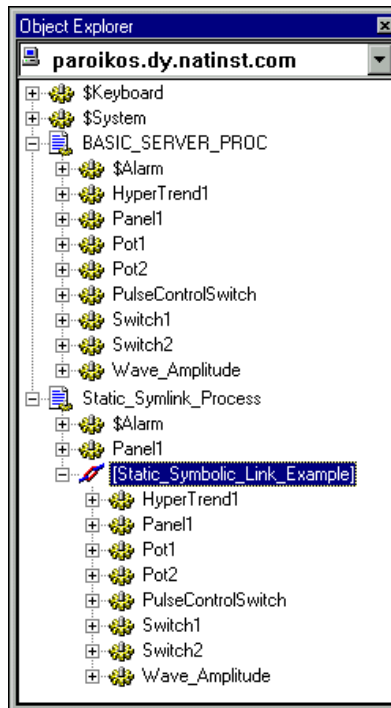
The main advantage of this connection is that if you move the server process to another computer, you can keep your links intact simply by changing the computer name in the Symbolic Link's **URL** field.

In the URL field you can specify an entire process, a folder inside a process, or even a single object.

Dynamic Symbolic Links

You can use a Symbolic Link with its **Source** set to **Dynamic** to implement redundancy, multiplex panels, and connect to complex expressions.

For more information about using a Symbolic Link to switch servers, refer to Chapter 14, *Setting Up Redundancy*. For more information about using a Symbolic Link to multiplex a panel, refer to the *Using Panel Templates* section.

You can use a dynamic Symbolic Link to make a remote source connection involving an expression, for example, if you want one control on your panel to adjust multiple hardware data members one at a time. To accomplish this task, follow these steps:

1. Create a selector control for switching between different connections. For example, you might create a Pot with three positions if you want to switch between three I/O points.

2. Create a Symbolic Link with its **Source** set to **Dynamic**. In the **Exp** field, enter an expression that produces the desired connection. For example, the expression shown in the following illustration could point to `Modbus1.40001`, `Modbus1.40002`, `Modbus1.40003`, and so on, depending on the current value of the `Selector_Pot`.
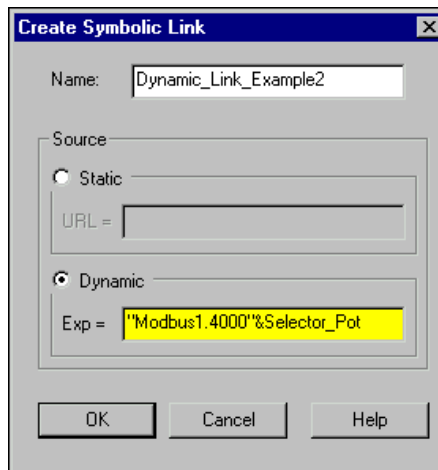
Path Modes in Lookout

When you insert an expression on a control panel or make a connection to a data member, you can use a value that comes from the process you are working in, from another process running on the same computer, or from a process running on some other computer on your network. Setting the *path mode* controls the way your process finds the value. The path modes used also determine how your process operates when copied or moved to another computer. The path modes in Lookout are Absolute, Relative (Lookout default), Process Relative, and Computer Relative.

*   A*bsolute mode* is indicated by two backslashes (\\) followed by a complete network path, as in

    `\\computer.place.com\process\folder\..folder\object`

Absolute mode is, in effect, a URL. Use Absolute mode when you want to refer to a particular process on a particular computer. Any references to a PLC, for instance, should be in Absolute mode so as to always refer to the computer physically connected to that PLC—no matter where the process making the reference is running.

- *Computer Relative* mode is indicated by a prefix consisting of two backslashes, a period, and a backslash (`\\.\`) followed by a process name, as in `\\.\process\folder\...folder\object`

  Use Computer Relative mode when you want to reference other processes that will always be running on the same computer.

- *Process Relative* mode is indicated by a single prefixed backslash (`\`) followed by a folder or object name, as in `\folder\...folder\object` or `\object`

  Use Process Relative mode when you want to make a connection between two objects in different folders in the same process.

- *Relative mode* instructs the computer to search for something in the directory relative to the currently active folder.

  In *Relative* mode, the form of the path is `object` or `folder\object` without any prefix.

  Relative mode is the default mode in Lookout. In this mode, all paths given are relative to the folder or process that contains the expression you are creating or the object you are connecting to. Connections between objects in the same folder should use Relative mode so that they can be copied and moved elsewhere while maintaining the same relationship with each other.

Connecting to Processes on Other Computers

Many Lookout dialog boxes that you use to create an object, display an expression, make a connection, and so on contain some variation of the following tree view. This tree view is used to locate any object in any Lookout process running on your network.

Each computer registered as running Lookout appears in this tree view. The pull-down combo box at the top indicates the root location. To display all the computers registered as running Lookout as well as displaying your own computer as local, set **Universe** as your root. You can set your local computer, the network of registered Lookout computers, any specific computer in the network, or any process as the root by right-clicking that node and selecting **Make Root**.

To edit a process, you must work from your computer's local node (or a process running on it). You can distinguish between Lookout's local and network representation of your computer in the root selection window by looking at the computer icon. The network representation of your computer name has red arrows representing the network I/O connection, as shown on the right in the following illustration. The local representation, on the left, has no I/O arrows.

Control Logic

This section briefly discusses how to implement control logic in Lookout processes. For more information and examples, refer to the NI Developer Zone at `ni.com/zone`.

Lookout Objects that Implement Logic

You can browse the object categories, especially Calculation, Monitoring, Statistical Process Control, and Timers, to determine ways you can implement control logic in Lookout. The following are a few objects that you might use to implement your control strategies.

- Use the Sequencer object for moving through batch process stages, for implementing loops, or for complex decision trees.
- Use the advanced features of the ActiveX Listbox control to implement queues or stacks.
- Use the Pulse object and other timers for ramping controls up and down.
- Use the DataTable object for lookup and staging.
- Use the Flipflop and Latchgate objects to implement on/off logic structures.

Comparisons in Expressions

When setting up process control strategies, you may want to compare two numeric values. You might want to check a current value against a set point or configure certain actions to occur based on the state of a particular object. You can make connections using expressions to incorporate this type of control logic or other features.

Example—Filling a Tank

As an example of using comparisons in your control logic, you could use the following expression to control a tank fill valve:

```
Tanklevel<50
```

where `Tanklevel` is the data member that reflects the actual level of fluid in the tank. The expression is TRUE when `TankLevel<50` and FALSE when `TankLevel` is greater than or equal to 50. You can use this expression in a PLC or RTU data member that controls whether the valve is open. Notice, however, that this expression could cause the valve to fluctuate between open and closed too often if the tank level hovers around 50.

Using a Neutralzone object might be a more appropriate method for controlling the tank fill valve. The following figure shows how you might set the high and low limits of the Neutralzone for the tank level.
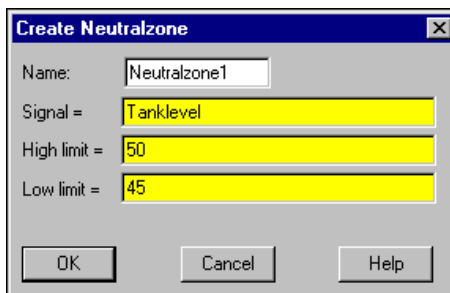
You could then use the following expression in the PLC or RTU data member:

```
Neutralzone1.below
```

`Neutralzone1.below` is TRUE when the tank level falls below 45. With the logical result of this expression connected to a PLC or RTU data member that controls the valve, the valve opens if the tank level falls below 45 and closes when the tank level reaches 50.

For even more flexibility in how the tank is filled, you could add a switch to control whether the valve is locked open, locked closed, or opened and closed according to your control logic. You can implement this strategy with a RadioButton or an HOA switch that you create using a Pot object, as shown in the following figure.
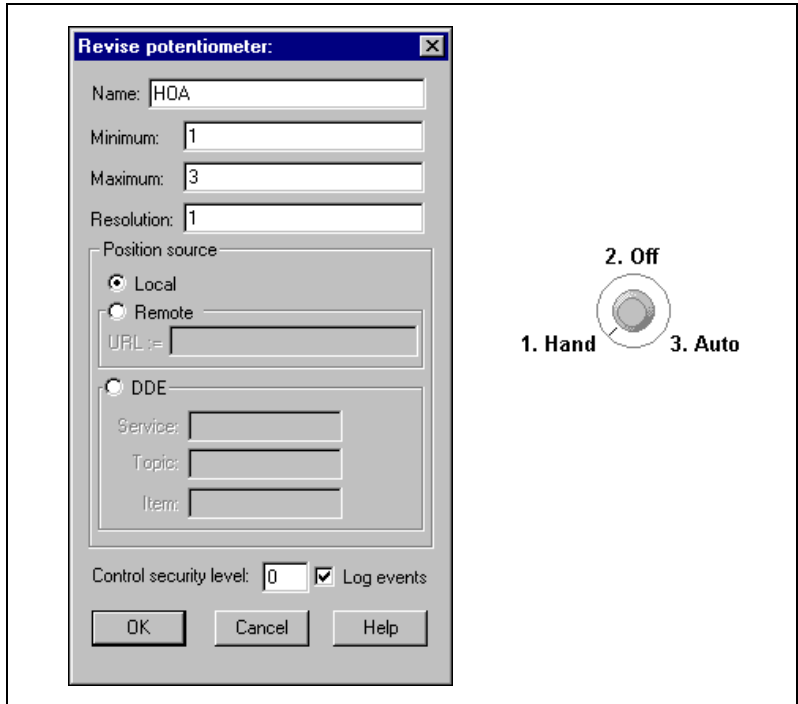
You could use the following expression in the PLC or RTU data member used to control the valve:

```
HOA=1 OR (HOA=3 AND Neutralzone1.below)
```

The valve would open if the HOA knob is in the **Hand** position, or if the HOA knob is in the **Auto** position and the tank level falls below 45.

Simulating I/O

If you want to simulate I/O, you can create a Modbus object communicating through COM1, connect COM1 to COM2 using a null modem cable, and create a ModbusSlave object communicating through COM2. This can be useful for developing control strategies when your hardware is not available during development time.

Creating Panels and Navigation

Control panels are where you display items such as switches, Pots, and trends. You can make them look like physical control panels, complete with pushbuttons, bezels, insets, lamps, gauges, and so on. Objects can be displayed on a single control panel or on multiple control panels. For example, you could insert the same switch on several panels. If you flipped the switch on one panel, it would immediately flip on all the other panels.

There is no limit to the number of control panels you can create in Lookout or the number of elements that you can display on any panel. Control panels can be any size and displayed in various states: maximized, normal, or minimized. Only one panel can be maximized at a time in Lookout. Maximizing a panel minimizes any other maximized panel.

Navigating Through Multiple Panels

It is common to have several or even dozens of control panels. In this case, you should designate one *home panel* and design a way for operators to switch between panels. A common way to allow panel switching is to use Pushbuttons on each panel that activate the other panels. Another option is to enable panel navigation arrows. These arrows, located in the right side of the status bar, activate control panels in the order in which you last accessed them. This feature is most convenient when you need to cycle through a subset of panels in a short period of time. To enable panel navigation arrows, select **Options»System**, and check the **Show panel navigation arrows in status bar** checkbox.

Using Panel Templates

If you have multiple control panels that with similar displays, using a panel template can simplify your user interface and improve system performance. Lookout panel templates allow you to *multiplex* your data, routing data from multiple sources to one location, where you can switch between sources. For example, suppose you are monitoring and controlling three similar pump stations. Your could have separate, dedicated control panels for each of these stations. A better solution might be to use a panel template, one control panel, where the data could be multiplexed from these different device sources.

A panel template can display data from, or write data to, only one device at a time. If you need the ability to read from or write to more than one device simultaneously, multiplexing to a panel template will not work. You can customize a panel template to include graphics that change with the

multiplexed data, as described in the *Multiplexing Graphics* section of Chapter 6, *Graphics*. However, you would need separate, dedicated panels for other differences in panel layout.

The following general steps explain how to set up multiplexing to a panel template.

1. Create a control panel to display the data. On the panel, include some method of switching between your various data sources, such as radio buttons (using the RadioButton object) or keyboard function keys (using the $Keyboard object).

2. Create a Symbolic Link to dynamically configure the path of the data. In the dialog box for the Symbolic Link, choose **Dynamic** as the **Source** and enter an expression function to point to the multiplexed sources. Be sure to use absolute paths that point to the server process. Your expression might look something like the following:

   ```
   choose(SelSource,"\\PC\ServerProc\Device1","\\PC\
   ServerProc\Device2","\\PC\ServerProc\Device3")
   ```

   If you have so many sets of data that the `choose` function becomes tediously long in the dialog boxes, you can use a DataTable instead. Refer to the *Using a DataTable to Manage Source Information* section for more information about using DataTables in this way.

3. Add your displays and controls to the panel and make your connections. You can even include a HyperTrend graph, graphics, or a text expression that indicates which data is being displayed. For more information about using HyperTrends and graphics with panel templates, refer to the *Using HyperTrends on Panel Templates* section and the *Multiplexing Graphics* section of Chapter 6, *Graphics*.

   • To read from a device—From Object Explorer, drag a data member to your panel and choose the display type that you want.

   • To write out to a device—Create your control and connect to a device through the Symbolic Link.

   • To include a title or text on the panel template—Select **Insert»Expression**, set **Display Type** to **text**, then enter an expression in the yellow field using the `choose` function.

Using a DataTable to Manage Source Information

For a larger number of possible choices of sources where a `choose` function becomes unwieldy, you can use a DataTable to manage the source information. For example, you could create a DataTable object called Table1 in your client process with the first column containing the path to the source. The second column could include the name of the device. Other columns could include additional information or notes about these stations. Refer to the *DataTable* topic in *Lookout Help* for more information about using the DataTable object.

Using HyperTrends on Panel Templates

You may want to use a HyperTrend to view live and historical data on a template panel. To do so, make sure that data logging is enabled for each data member in the server process. You may also want to create separate groups for the HyperTrend in the client process with the displayed traces from each device collected under one group.

Representing Data Members on Control Panels

Most objects are not automatically shown on a control panel when you create them. You can create a display or a control for a data member, from either Object Explorer or the **Insert** menu.

Displaying Data Members from Object Explorer

Follow these steps to display a data member on a control panel using Object Explorer:

1. Make sure Lookout is in edit mode and open the Object Explorer by selecting **Object»Object Explorer**.

2. Navigate to the object or data member you want to display on your panel. In the following illustration, a data member from a Waveform object running on a server process located on another computer on the network has been selected.
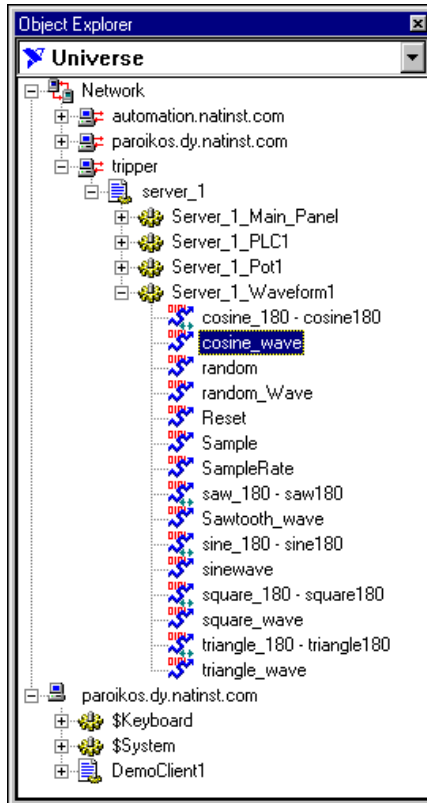
3. Drag and drop the object or data member onto the panel you want the
    data to display in, using either of the following methods:

    •   Left-click drag and drop to create a display for a readable data
        member or a control for an (implicit) data member. Lookout
        inserts an expression to display the data. You may have to adjust
        the size of the display for the data to appear properly.

- Right-click drag and drop to get more display options. The following dialog box appears.

Create Control ▶	Pot
Create Display	Switch
	PushButton
Insert Expression as Default	TextEntry
Insert Expression as Logical	RadioButtons
Insert Expression as Numeric	CWButton
Insert Expression as Text	CWKnob
Cancel	CWSlide
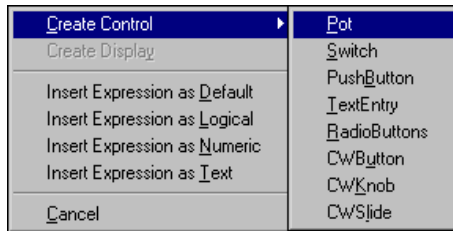
From this menu you can choose to create a control with a remote connection to the object or data member. You can also insert that value as an expression of the type you choose.

If you drag and drop an item from the local node of Lookout running on your computer, you will display the item using a relative path. If you drag an item from a process under the **Network** node (including the network node for your local computer), you will display the item with an absolute path. Refer to the *Path Modes in Lookout* section for more information on Lookout path modes.

4. Right-click the display to change its display characteristics, modify the expression, change the data format, or alter any other properties.

Displaying Data Members from the Insert Menu

Complete the following steps to display data members using the **Insert** menu.

1. Select either **Insert»Expression** or **Insert»Displayable Object**, as follows:

   - Select **Insert»Expression** and the Insert Expression dialog box appears as shown in the following illustration.
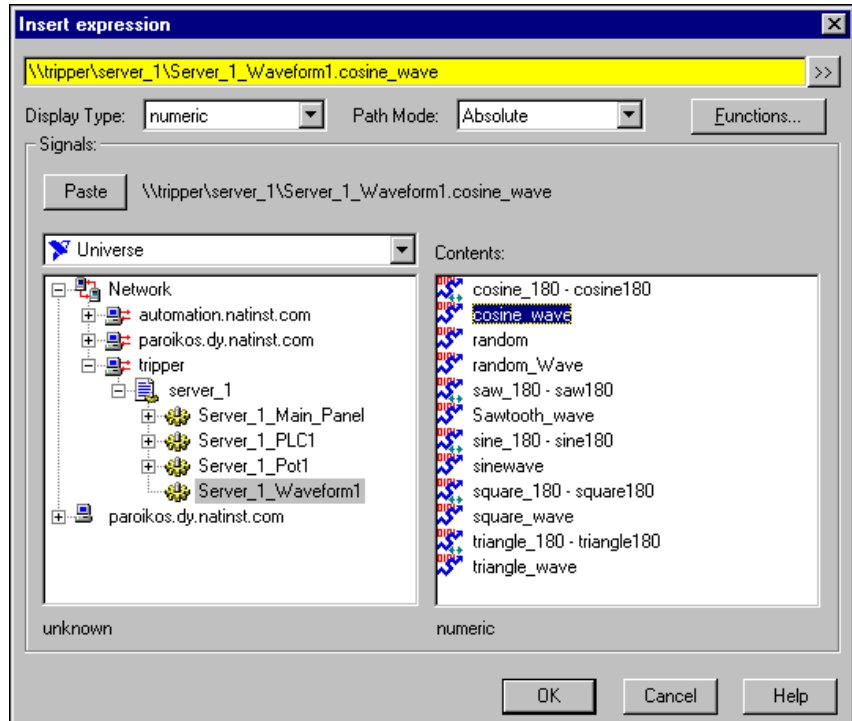
> Set the **Display Type** appropriately (you cannot change it later). Navigate to the data member you want to display, and set the **Path Mode**. Refer to the *Path Modes in Lookout* section for more information about path modes. Select the data member you want to display in the **Contents** list, and click **Paste**.

- Select **Insert»Displayable Object** and highlight the data member you want to display.

2. Click **OK**. The Display numeric signal or Display logical signal dialog box appears. The type of dialog box depends on the type of data member being inserted (numeric or logical). In this case the signal is numeric, a cosine wave generated by the Lookout Waveform object.
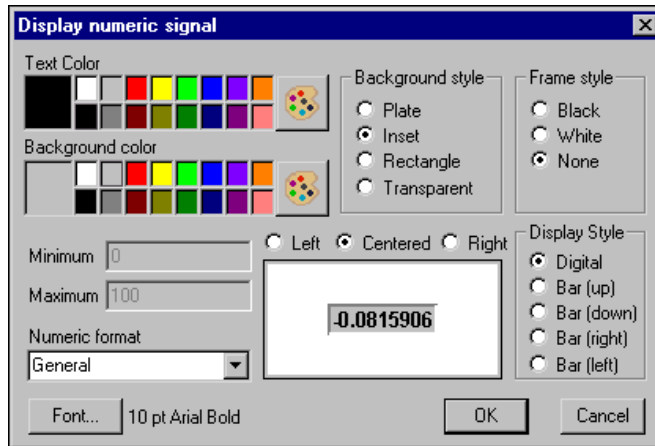
3. Select the color, style, and format parameters you want. To choose a custom color, click the paint palette button. Click **Font** to change the way your text looks.

4. Click **OK**.

Creating Custom Control Panel Elements

Besides using the standard Display objects available in Lookout, you can add other graphical elements as described of Chapter 6, *Graphics*. You can also use the ActiveX objects to create such displays as a Windows-style listbox or an embedded Web browser window on your control panel. Refer to the *ActiveX Objects* section for more information about using ActiveX objects in Lookout.

6

Graphics

This chapter explains the use of graphics in Lookout.

Lookout provides an extensive graphics library including switches, potentiometers, pushbuttons, bar graphs, valves, tanks, pumps, plates, insets, scales, and more. You can also use graphics from the Image Navigator utility, which comes with Lookout. If the standard Lookout and Image Navigator graphics do not exactly fit your needs, you can use any other drawing software to create your own custom graphics.

All graphics are either static or dynamic. *Static* graphics never change state, but *dynamic* graphics change state to represent process variations.

Displaying Static Graphics

Static graphics never change state. They exist on a control panel much the same way a picture hangs on your wall, never changing or moving.

Complete the following steps to display a static graphic on a control panel.

1. Select **Insert»Graphic**. The Select graphic dialog box appears and shows all bitmaps and metafiles located in the `Lookout\Graphics` subdirectory.
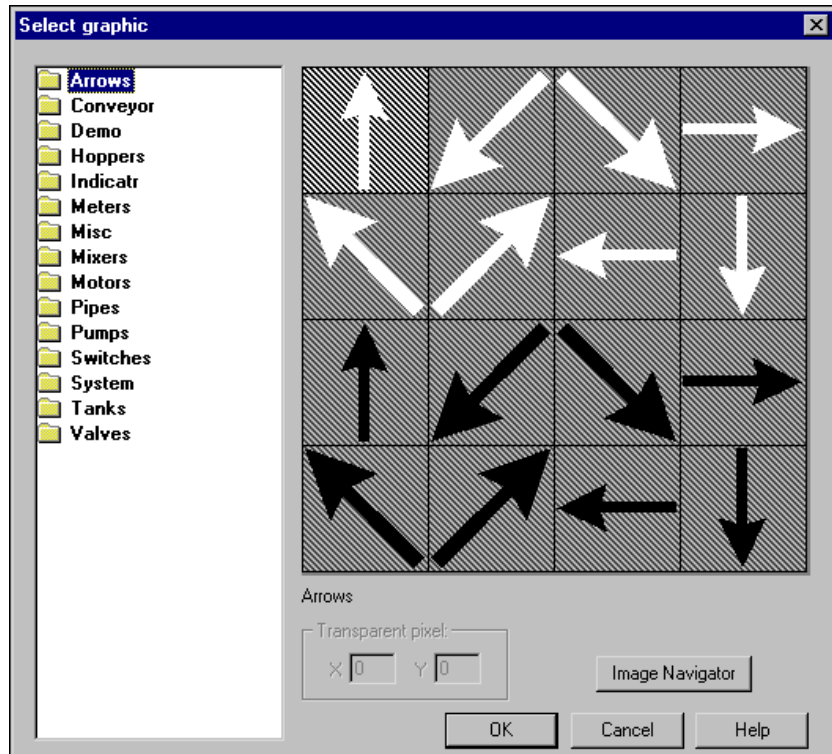
2. Select the graphic you want, and click **OK**, or click **Image Navigator** to launch the Image Navigator utility. Refer to the *Image Navigator* section for more information about this utility.

Windows metafiles are normally easier to use and manipulate than bitmap images. Refer to the *Graphic File Types* section for more information about the differences between bitmaps and metafiles. You can specify transparent pixels for bitmaps, as explained in the *Creating Transparent Bitmaps* section.

Displaying Dynamic Graphics

Many process control panels require some type of *animation*, such as a pump changing colors to represent on and off. Table 6-1 lists the Lookout elements that use animation or change states.

Table 6-1. Lookout Components that Use Dynamic Graphics

Lookout Component	Description
Animator object	An Animator object provides full graphical animation including horizontal and vertical motion, dynamic resizing and visibility, dynamic sequencing, and color changing.
Multistate object	A Multistate object displays up to six different custom graphics based on advanced if-then-else logic.
Pipe object	A Pipe object makes a rectangle or line (of any dimension) change colors or blink, based on advanced if-then-else logic.
DialGauge object	A DialGauge object displays a numeric signal as a sweeping needle on an analog gauge or dial.
Gauge object	A Gauge object makes a numeric expression (digital number or barchart) change colors or blink based on advanced logic.
Spinner object	A Spinner object is a small rotating disk. It can be turned on and off with a logical signal, and its rotation speed and direction are controlled by a numeric value.
Switch object	A Switch object can represent its two positions using standard switch symbols or custom graphics.
Pushbutton object	A Pushbutton object looks like a button that changes when depressed, but you can also make it transparent. You might use transparent pushbuttons over custom graphics.
Pot object	A Pot object can be displayed as a knob, vertical slider, horizontal slider, increment and decrement buttons, or a digital number.
Logical expression	When you create an expression that results in a logical value, you can represent that value using standard lights, text, or custom graphics.
Numeric expression	When you create an expression that results in a numeric value, you can represent that value using a digital number or a vertical or horizontal bar chart.
Text expression	When you create an expression that results in a text value, you can represent that value using any style and size font loaded on your computer.

Refer to the *Lookout Help* for information about a specific object. Refer to Chapter 7, *Expressions*, for information about expressions.

Displaying Dynamic Logical Signals

You can use various graphics and colors to represent the state of logical signals, or data members, on your control panel. The following illustration shows a switch in the ON and OFF positions. The graphics on the right side show three ways of graphically displaying the logical signal generated by the switch.
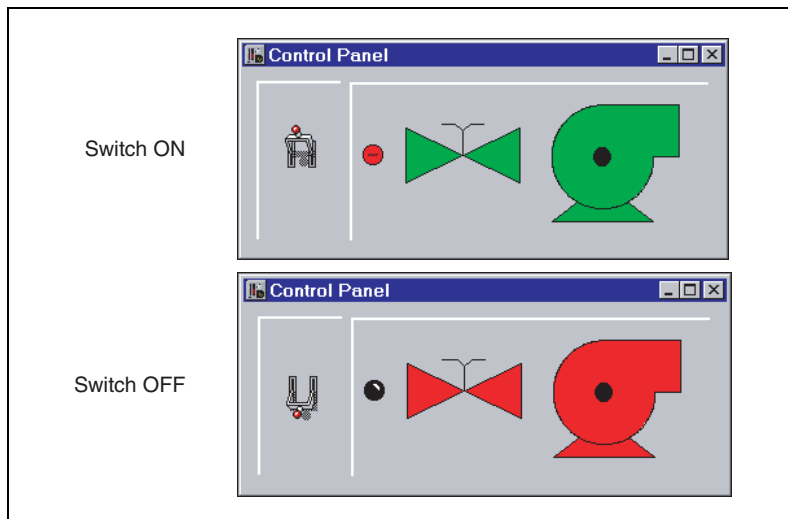
Complete the following steps to insert a display expression representing a logical signal (such as the on/off state of a switch):

1. Create an expression by selecting **Insert»Expression** or right-clicking the control panel and selecting **Insert Expression**. The Insert expression dialog box appears.

2. Select the logical value you want to display, click **Paste**, then click **OK**. The Display logical signal dialog box appears.
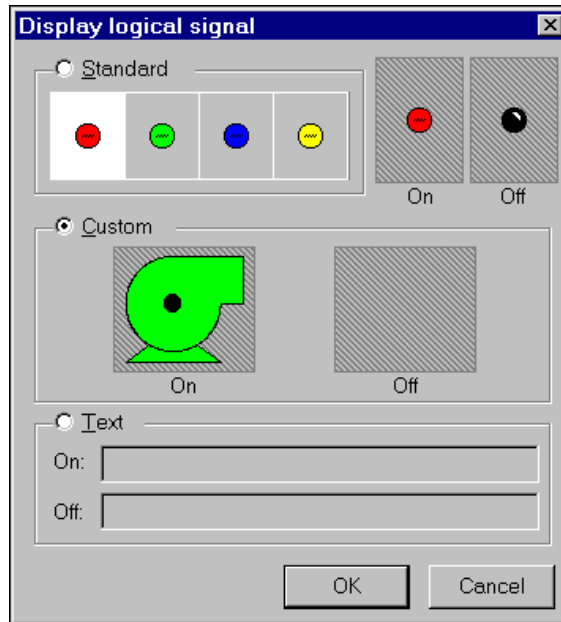
3. Select the display characteristics for your logical signal.

    •   To use a standard display, click the **Standard** radio button and
        click a color for the **On** state.

    •   To use a custom graphic, click the **Custom** radio button, click in
        the Custom **On** and **Off** boxes to select graphics for those states.

    •   To use a text display, click the **Text** radio button and type the text
        that you want displayed for each state.

4. Click **OK**.

You can revise this display at any time by right-clicking it and choosing
Display Properties.

If you want to represent more than two conditions with dynamic graphics,
use a Multistate object. Similarly, you can use a Pipe object to make a line
or rectangle dynamically change colors and blink.

Displaying Dynamic Numeric Signals

You can use a Pot object to display dynamic numeric signals, or data members. Figure 6-1 shows the Pot as a slider, a bar graph, a digital entry, and increment/decrement buttons—all are graphical representations of the same Pot object.
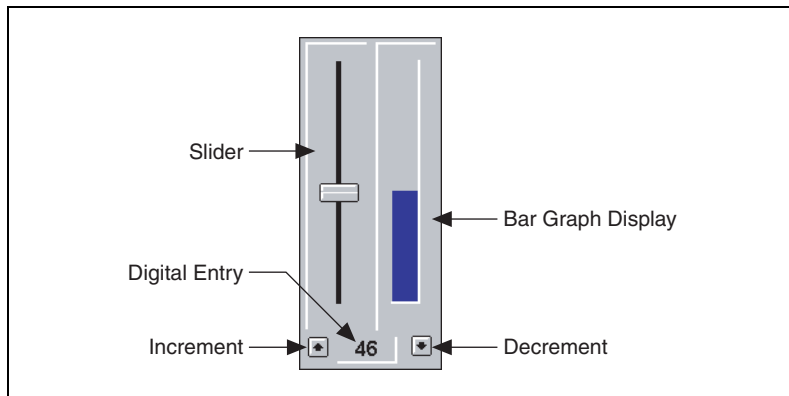
Figure 6-1. Pot Object Used to Display a Dynamic Numeric Signal

Complete the following steps to display a numeric value.

1. Create an expression by selecting **Insert»Expression** or right-clicking the control panel and selecting **Insert Expression**. The Insert expression dialog box appears.

2. Select the numeric value you want to display, click **Paste**, then click **OK**. The Display numeric signal dialog box appears.
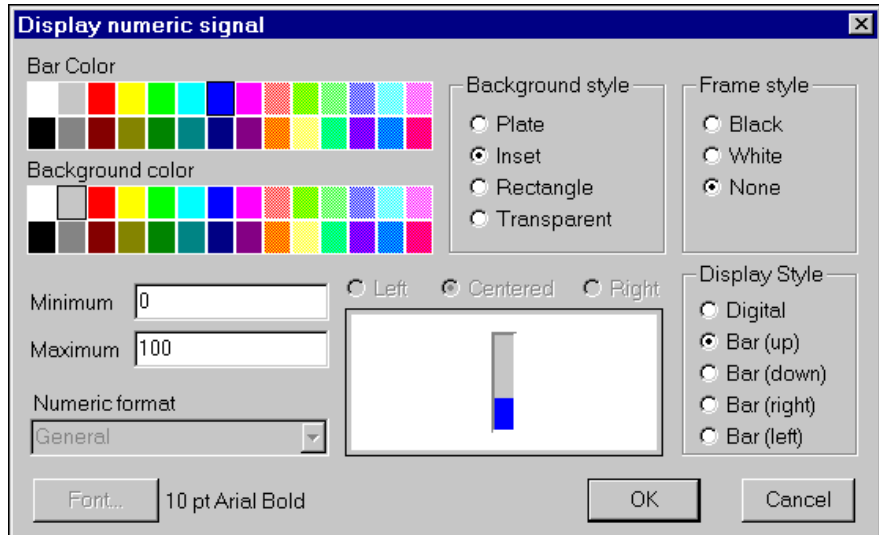
3. Select the display characteristics for your numeric signal.

4. Click **OK**.

You can revise this display at any time by right-clicking it and choosing **Display Properties**.

If you want to represent the numeric signal with moving custom graphics, use the Animator object.

Displaying Dynamic Text Signals

You can display dynamic text messages on a control panel with text expressions. You can easily display up to two separate dynamic text messages using a logical display, as described in the *Displaying Dynamic Logical Signals* section. You can use a numeric signal to display more than two different text messages, as follows.

1. Create a numeric control, such as a Pot or RadioButton, to generate numbers that will correspond to your text messages (1–4 for four text messages).

2. Select **Insert»Expression** and enter an appropriate expression in the yellow field, using the CHOOSE expression function. The first variable of the expression function is the object that you created in step 1. The other variables are the text messages. The following illustration shows an example. Refer to Chapter 7, *Expressions*, for information about the CHOOSE function.

Insert expression

CHOOSE (Pot1, "Mixing", "Melting", "Cooling", "Reacting"

Display Type: text   Path Mode: Relative   Functions...

Signals:

Paste   Pot1

paroikos.dy.natinst.com   Contents:

- $Keyboard
- $System
- Demonstrations
  - $Alarm
  - Demo_Panel_1
  - **Pot1**

- (implicit)
- value

Pot   numeric

OK   Cancel   Help
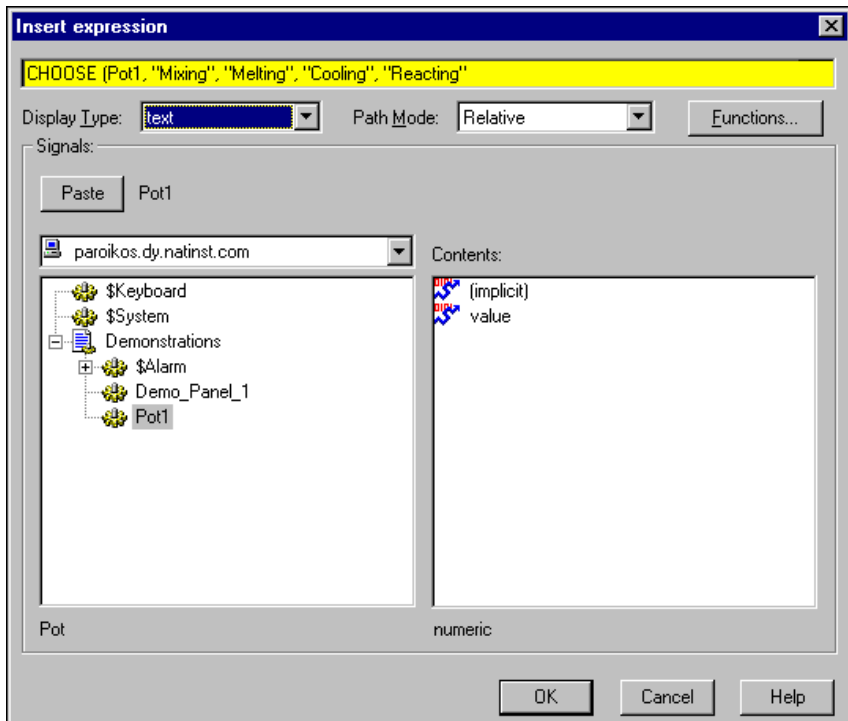
3. Click **OK**. The Display text signal dialog box appears.

4. Select display characteristics for the text signal expression and click **OK**.

You can revise this display at any time by right-clicking it and choosing **Object Properties** or **Display Properties**.

Displaying Text, Plates, Insets, Rectangles, and Lines

Effective use of text, plates, and insets make control panels more intuitive and usable. The following example demonstrates how plates and insets organize information about two pumps.
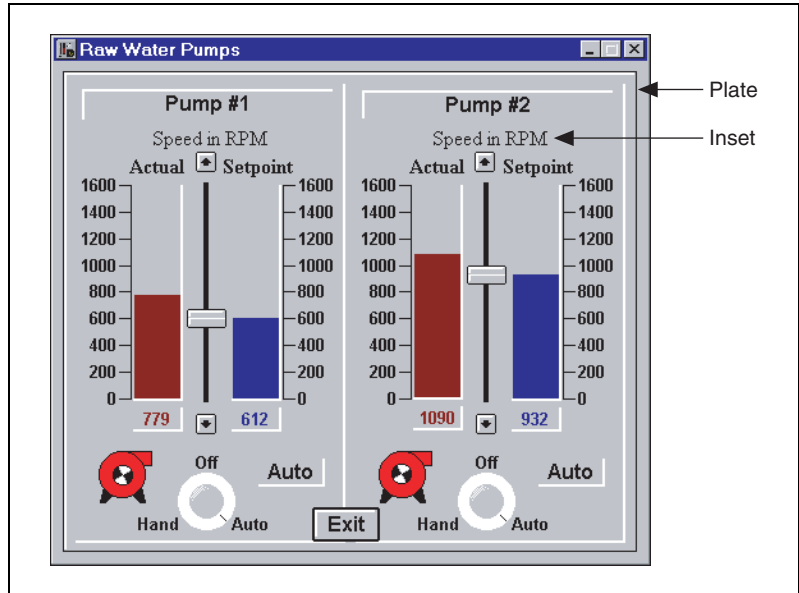
To create static text, plates, insets, rectangles, or lines in Lookout, select
Insert»Text/plate/inset. The Insert text/plate/inset dialog box appears.
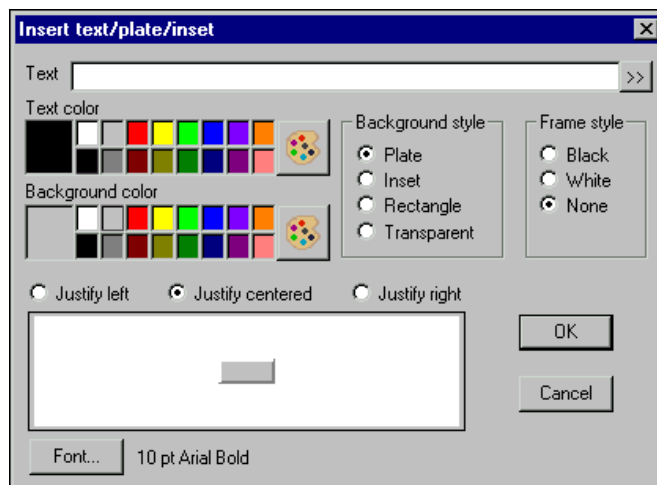
- Click the **»** button to the right of the **Text** field to expand the field for
  multiple line entries. You can enter text using international character
  sets such as Chinese, Korean, and Japanese.

- If you want to create a rectangle, line, plate, or inset, leave the **Text**
  field blank and choose a **Background style**.

- To create a vertical or horizontal line, choose **Rectangle**. After clicking **OK**, size the rectangle to the desired length, and reduce the width to the thickness you want your line to be. If you need the rectangle an exact size, use the yellow status bar, which indicates dimensions.

✎ **Note** Lookout uses color grids in many dialog boxes. The grids make selecting colors quick and visually helpful. However, they might look slightly different from computer to computer.

Image Navigator

In addition to the standard graphics library that comes with Lookout, you can also choose from a collection of symbols, drawings, and line art provided by the Image Navigator utility.

The following general steps explain how to use Image Navigator graphics on your Lookout control panel.

1. (Optional) Any graphic that you choose to use from Image Navigator will be placed into the `Lookout\Graphics` directory, so you may find it convenient to create a subdirectory for your Image Navigator graphics before you begin using the Image Navigator.

2. Select **Insert»Graphic** and open the folder in the left tree view where you want to store the Image Navigator graphic.

3. Click the **Image Navigator** button.

4. Browse through the categories to select a graphic to use. Your choice is shown in the **Preview** area. You can click the **Options** button to change some aspects of the graphic's appearance.

5. Click **OK** to choose the graphic. When possible, Image Navigator gives you a choice between `.bmp` and `.wmf` format. If you want to resize or animate colors with the graphic in Lookout, choose metafile format.

   Image Navigator saves the graphic in the folder that was selected earlier, with the name `in000N.bmp` or `in000N.wmf`, where *N* is the number of images exported to that directory. To change the name of this file, locate it in the `Lookout\Graphics` directory and rename it there.

For more information about customizing graphics in Image Navigator, adding images to existing categories, and using graphics from Image Navigator in your application, refer to the Image Navigator online help.

Creating Custom Graphics

There might be times when none of the Lookout standard graphics exactly fits your needs. You can use any drawing software or Image Navigator to create your own custom graphics. Save your graphic in either Windows Device-Independent Bitmaps (`.bmp`) or Windows Metafiles (`.wmf`) format. Then place a copy of the graphic file into the `Lookout\Graphics` subdirectory of your choice. Lookout cannot use graphics unless they are stored in the `Lookout\Graphics` directory.

Creating Transparent Bitmaps

If you choose to insert a bitmap (`.bmp`) file, you can specify a color in that graphic to be transparent. The remainder of the bitmap then becomes a sort of mask or frame on the control panel, as demonstrated in the following illustration.
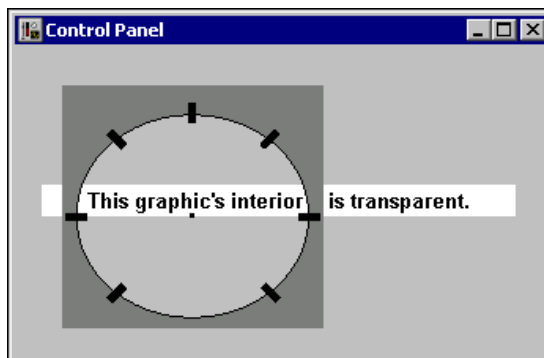
Complete the following steps to specify a transparent color for a bitmap.

1. Select **Insert»Graphic** and select a bitmap graphic, or right-click a bitmap on the control panel and choose **Display Properties**. The Select graphic dialog box appears.

2. Enter **X** and **Y** coordinates in the **Transparent pixel** fields, and the corresponding color becomes transparent. The top-left corner of the bitmap is **X**=0, **Y**=0; the center is **X**=–1, **Y**=–1; the bottom-right corner is **X**=–2, **Y**=–2. To get the result you want, try entering a value for **X**, then a value for **Y**, then click in the **X** field to see how the bitmap is affected.
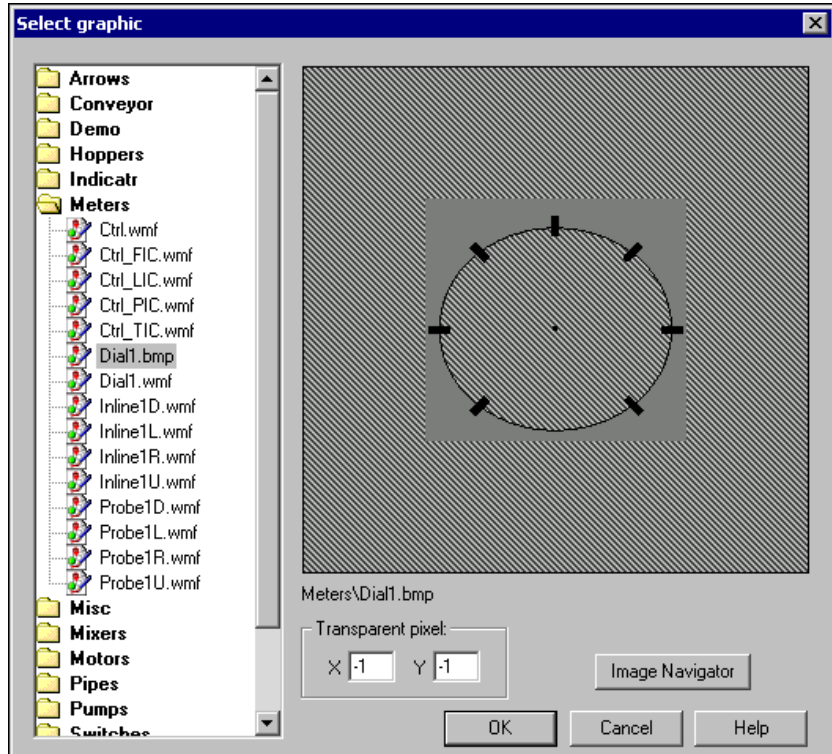
Windows Metafiles (.wmf) have transparency built into them. They inherently draw themselves on top of what is already there, so if the graphic does not contain instructions to draw over a certain pixel, that pixel appears to be transparent.

Graphic File Types

Lookout accepts both Windows Device-Independent Bitmaps (.bmp) and Windows Metafiles (.wmf).

Bitmaps

Windows bitmap files typically have a .bmp file extension. They are always rectangular and never resizable in Lookout. You can define transparent pixels for bitmaps in Lookout, for the parts of the rectangle you do not want to show. Refer to the *Creating Transparent Bitmaps* section for more information.

Bitmaps effectively handle large background schematics or system overviews. Bitmaps usually display faster than metafiles. A complex drawing rendered as a bitmap can display 100 times faster than the corresponding metafile. If you want to display scanned photographic quality images with hundreds of colors, use bitmaps.

Metafiles

Windows metafile files typically have a `.wmf` file extension. Metafiles are not necessarily rectangular. They can be resized and stretched to any size or aspect ratio in Lookout.

Metafiles are typically smaller than bitmap files, so they take up less disk space and consume less memory than bitmap files. You can use one file that contains a metafile drawing of a pump to display several pumps of various sizes on the screen.

Multiplexing Graphics

When you multiplex data sources to a panel template, you may want to use a small graphic for the picture of the equipment, or the plan of the physical layout at each station. You can use a Multistate object to switch between six cases. If you have more than six possible sources, and therefore more than six possible graphics that you have to switch, you can use multiple Multistate objects, carefully stacked on top of each other. Also make sure that the `Else` condition is left blank on each Multistate, which makes no graphic appear when no conditions apply, and choose the conditions so that two states on two Multistates do not simultaneously become true. Refer to the *Multistate* topic in *Lookout Help* for more information about using that object.

Arranging Graphics

You can move anything on a control panel by clicking and dragging it. You can move a display one pixel at a time by selecting it, placing the mouse cursor over it, holding down the space bar, and pressing the arrow keys on your keyboard.

Align and group control panel displays using the commands in the **Arrange** menu. You can activate the snap to grid feature by selecting **Edit»Snap to grid**. This feature aligns objects you are positioning to a grid on the control panel. You can adjust this grid by selecting **Edit»Options**.

Finding a Lost Graphic File

If a lost graphic icon appears on your control panel, right-click it in edit mode and choose **Display Properties**, as shown in the following illustration.
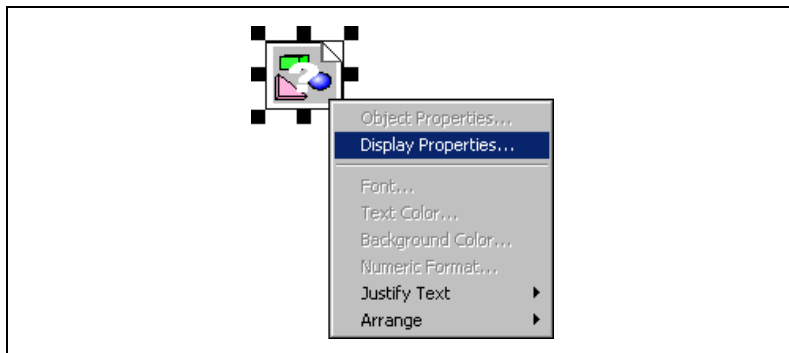
The Select graphic dialog box appears, and the expected path and name of the graphic is shown below the graphic preview.

Graphics and Memory Considerations

Graphics can be memory intensive. If your control panels are graphic intensive, and you experience performance degradation, you can either simplify your graphics or add memory to your computer.

7

Expressions

This chapter explains the features and uses of Lookout expressions, which can act as variables, capable of real-time math statements, condition testing, and other complex operations.

Expressions Overview

You can use expressions in many places, and in many ways, in Lookout. You can create expressions as independent (expression) objects, enter expressions in yellow object parameters fields, connect object data members to expressions, or insert expressions on control panels.

Lookout expressions can be short and simple, such as a constant or the name of an object, such as `True` or `Pot1`. They can also be extremely complicated with several signal inputs, functions, operators, and parentheses, such as `if(HOA=1,"Hand",if(HOA=2,"Off","Auto"))`.

If any value referenced in an expression changes when an event occurs, the expression automatically recalculates.

Table 7-1 shows some sample expressions.

Table 7-1. Sample Expressions

Expression	Description
`True`	Always returns the value true (on).
`Pot1`	Returns the current value, the `(intrinsic)` data member, of Pot1.
`879.03`	Always returns the value 879.03.
`1:04:33`	Returns the value 0.0448264. If formatted in hours, minutes, and seconds, it is displayed as 1:04:33.
`Pressure * 2.31`	Multiplies Pressure by 2.31.

Table 7-1. Sample Expressions (Continued)

Expression	Description
`(Switch1 or Switch2) and TankLevel > 65.2`	If Switch1 or Switch2 is true, and TankLevel is greater than 65.2, return true. Otherwise, return false.
`Pressure >= Setpoint and !ReliefValve and TimeOccuring > 0:30`	If Pressure is greater than or equal to Setpoint, and `ReliefValve` is false, and `TimeOccuring` is greater than 30 seconds, return true. Otherwise, return false.
`if(PLC1.AutoPos, AutoTemp, if(PLC1.ManPos, ManualTemp,0))`	If the alias member `AutoPos` of PLC1 is true, return the value of `AutoTemp`. If the alias member `ManPos` of PLC1 is true, return the value of `ManualTemp`. Otherwise, return the value 0.
`if(HOA=1,"Hand",if(HOA=2,"Off","Auto"))`	`if` is the text version of the `if` function. If the HOA switch is at position 1, return `Hand`. If the HOA switch is at position 2, return `Off`. Otherwise, return `Auto`.
`choose(HOA,"hand","Off","Auto")`	If the HOA switch is at position 1, return `Hand`. If the HOA switch is at position 2, return `Off`. If the HOA switch is at position 3, return `Auto`.

Expression Data Types

A single expression can incorporate any number of numeric, logical, and text signals within its calculation. However, the *result* of the expression can be only one of three types: numeric, logical, or text. The leftmost function or operator in the expression returns a variable type that determines the overall signal type of the expression.

While an expression produces a single type of data, keep in mind that most Lookout objects interpret data according to the data type that they need for a particular data member. For example, you can connect a logical output to a numeric input, and the arriving data will be interpreted as a 0 or a 1.

Expressions used as parameters in Lookout objects do not permit you to select the data type because the object interprets input according to the data type required by that object. Making Lookout data polymorphic permits

you to use output of a data type different from that required by an object, as long as the output can be interpreted meaningfully by the object.

After you have placed an expression on your panel, you cannot change the data type in that expression. You have to delete that expression and place a new one if you want to change data types.

Refer to the *Data Type Matching and Conversion* section of Chapter 5, *Creating Lookout Processes*, for more information about how Lookout interprets data types.

Illegal Operations in Expressions

Make sure that your expressions do not attempt to calculate illegal operations, such as dividing by zero or finding the arc cosine of a number greater than 1. If a signal can assume a value that could cause Lookout to attempt an illegal mathematical operation, you should specifically test for that condition in the expression. Expressions trap illegal mathematical operations and generate alarms in the Math alarm area. The alarms do not reset until you correct the expression. The following list includes some of the illegal conditions that generate Lookout alarms:

- Attempting to divide by zero
- Taking the square root of a negative number
- Numeric underflow
- Numeric overflow

Expressions and Path Modes

Expressions depend on paths to function properly. Refer to the *Path Modes in Lookout* section of Chapter 5, *Creating Lookout Processes*, for more information about path modes.

Tabs and Spaces in Expressions

Because Lookout ignores white space characters in expressions, you can use tabs or spaces between the elements of an expression to make it easier to read. Lookout removes extra spaces when you close the dialog box.

Creating Expressions

Creating a Panel Expression

You can graphically display the result of an expression on a control panel with the following steps. This type of panel expression does not create an output signal that can be used by other expressions or objects.

1. Select **Insert»Expression**. The Insert Expression dialog box appears, as shown in the following illustration.
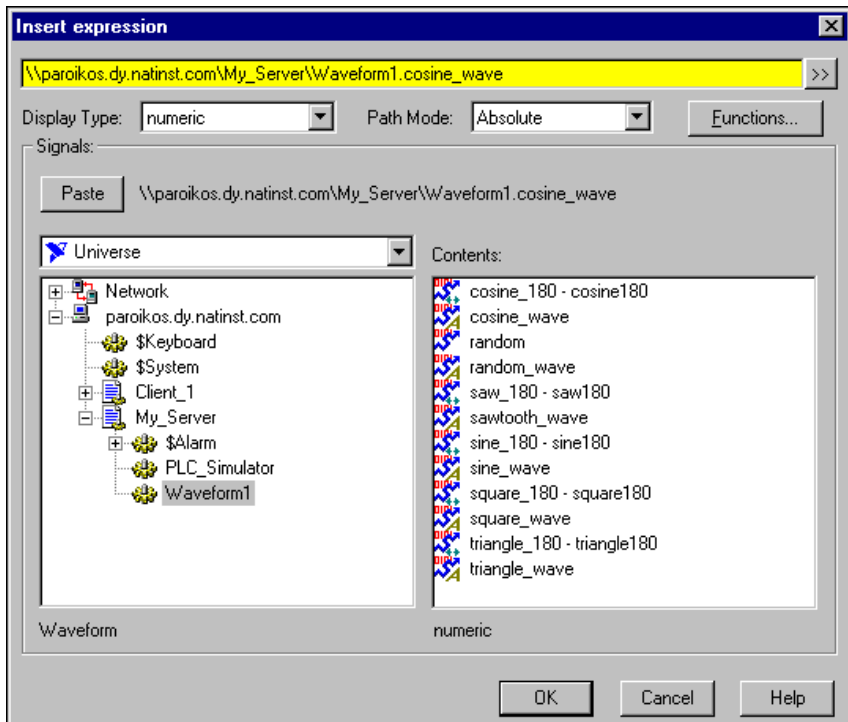
2. At this point, you can manually type the expression in the yellow field, being careful to use the correct spelling, syntax, and path mode prefix, or you can construct the expression as described in the following steps.

3. Set the **Display Type** and **Path Mode** appropriately. Refer to the *Data Members* section of Chapter 1, *Introduction to Lookout*, for more information about data types. Refer to the *Path Modes in Lookout* section of Chapter 5, *Creating Lookout Processes*, for more

information about path modes. After you have created an expression, you cannot modify the data type, so be sure you select the correct type for your display.

4. Construct the expression that you want, as follows.

    a. In the left tree view, navigate to the object that contains a data member that you want as part of your expression.

    b. In the **Contents** field, click the data member, then click **Paste**.

    c. Repeat steps a and b to add any other data members that you want in your expression.

    d. Add any constants or expression functions that you want, to complete the expression in the yellow field. Click the **Functions** button to choose an operator or function to insert.

5. Click **OK**. A dialog appears with display options.

6. Set the display options that you want for your expression and click **OK**.

Creating an Expression Object

When you need to define a unique condition that your application uses multiple times, use an *(expression)* object. Other expressions and/or objects can reference the output of an (expression) object by referencing its name.

Create an (expression) object just as you would any Lookout object. Enter the expression in the Create expression dialog box. You can right-click in the yellow field to launch the expression editor to assemble a long expression. Refer to the *(expression)* object topic in the *Lookout Help* for more information about the (expression) object.

Using an Expression for an Object Parameter

Many objects accept expressions as parameters. When they do, the parameter field is yellow, and the parameter expects the expression to return a certain signal type—numeric, logical, or text. Refer to the specific object topic in the *Lookout Help* for more information about object parameters.

If you right-click in a yellow parameter field, the Expression Editor dialog box appears. Using this dialog box, you can navigate to object and data members and construct your expression to paste back into the parameter field.

The following illustration shows a DelayOn object with expressions entered for two parameters.
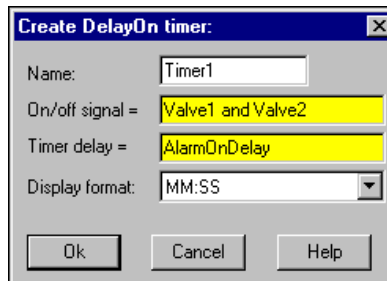
The parameter **On/off signal** expects a logical expression and **Timer delay** requires a numeric expression. Both `Valve1` and `Valve2` are names for switches. Connecting them with `and` produces a logical result that is TRUE if both valves are on and FALSE if either valve is off. `AlarmOnDelay` is the name of a Pot, which creates a numeric signal.

You could use the following options for the **Timer delay**:

- Enter the name of an output signal from another object such as a Pot, as in the illustration. In this case, the operator adjusts the delay.

- Enter a constant. In this case, the delay never changes.

- Enter a complex expression that automatically calculates the delay based on multiple inputs.

Similar configuration solutions exist for the **On/off signal** parameter or any other parameter that accepts expressions.

Using Expressions in Connections

You can connect object data members with expressions. Writable data members accept expressions as inputs. To connect an expression to a data member, follow the steps given in the *Connecting Data Members to Data Members* section of Chapter 5, *Creating Lookout Processes*. In the yellow field of the connections dialog box, you can construct an expression that includes constants, data members, operators, and functions. Click the **Functions** button in the dialog box to choose an operator or function to insert.

Modifying Expressions

To modify a panel expression, right-click it and choose either **Object Properties** or **Display Properties**, depending on what you want to modify.

✏️ **Note**   After you insert an expression as a display, you cannot change the data type of the expression. The Display Type dialog box will be disabled if you open the expression to edit it. To change the data type, you have to insert a new expression.

To modify an (expression) object, right-click it in Object Explorer and select **Properties**.

To modify an expression used for an object parameter, right-click that object in Object Explorer, and select **Properties**.

To modify an expression used in a data member connection, select **Object»Edit Connection**, double-click the object containing the data member, and double-click the **Existing connection** you want to modify.

Expression Operators

In an expression, *operators* are instructions to perform an operation on a value or to combine values to form a new value. You can type in an operator or click the **Functions** button to choose one. The following sections describe the operators you can use in Lookout expressions.

Arithmetic Operators

The following operators perform basic arithmetic operations. They require numeric operands and produce numeric results.

Table 7-2. Arithmetic Operators

Operator	Description
+	Addition
–	Subtraction
*	Multiplication
/	Division
%	Percentage (divides preceding value by 100)

Table 7-2. Arithmetic Operators (Continued)

Operator	Description
^	Exponentiation
–	Negation (additive inverse of the following value)

Text Operator

The text operator is the ampersand character (`&`). An ampersand joins two text strings. For example, the expression "`Flow rate is`" & `TEXT(Flow, "0.00")& "gpm"` produces a text signal of `Flow rate is 141.23 gpm`, assuming `Flow` is a numeric signal whose value rounds to 141.23. Typical uses for the text operator include the following:

- Imbedding a numeric value within a text string

- Using it in an expression to verify an action

- Using it in an alarm message

- Sending text out to a remote PLC display panel

Comparison Operators

Comparison operators compare two numeric values and produce a logical result of either TRUE or FALSE (on or off).

Table 7-3. Comparison Operators

Operator	Description
<	Is less than
>	Is greater than
<=	Is less than or equal to
>=	Is greater than or equal to
=	Is equal to
<>	Is not equal to

Avoid using the *is equal to* (=) and *is not equal to* (<>) comparison operators to compare an analog value from a PLC or the result of mathematical calculations in an expression. Because analog (floating point) values can have about 17 significant digits, `TankLevel=40` might never be exactly true. If you know the signals are integer values, you can use the = and <> operators.

Expression Functions

There are over 50 built-in expression functions, generally classified as follows:

- Logical functions
- Lookup functions
- Mathematical functions
- Statistical functions
- Text functions
- Trigonometric functions
- Date/time functions
- Quality functions

You can use any of these functions as part of an expression. You can type in a function or click the **Functions** button to choose one. This section describes each function, specifies the syntax, and provides an example on how to use the function.

Logical Functions

AND	Syntax	*logical1* AND *logical2* AND *logical3...*
	Example	Switch1 AND Pot1 > 50.0
	Description	If Switch1 is on *and* Pot1 is greater than 50.0, return TRUE, otherwise return FALSE. Returns TRUE if *all* logical parameters are TRUE.
FALSE	Syntax	FALSE
	Example	False
	Description	Returns the logical value FALSE. Accepts no parameters.
IF	Syntax	IF(*logical*, *result1*, *result2*)
	Example	IF(Switch1, 99.9, Pot1)
	Description	If *logical* is TRUE, return *result1*, otherwise return *result2*. In the example, if Switch1 is on, it outputs the value 99.9. Otherwise, it outputs the value of Pot1.

LIF	Syntax	`LIF(`*`logical`*`, `*`logical result1`*`, `*`logical result2`*`)`
	Example	`LIF(Switch1, False, Pot1 > 50.0)`
	Description	Included for compatibility with early versions of Lookout. If *`logical`* is TRUE, return *`logical result1`*, otherwise return *`logical result2`*. In the example, if Switch1 is on, it outputs the logical value `False`. Otherwise, it returns the logical result of `Pot1 > 50.0`.
NIF	Syntax	`NIF(`*`logical`*`, `*`numeric result1`*`, `*`numeric result2`*`)`
	Example	`NIF(Switch1, 99.9, Pot1)`
	Description	Included for compatibility with early versions of Lookout. If *`logical`* is TRUE, return *`numeric result1`*, otherwise return *`numeric result2`*. In the example, if Switch1 is on, it outputs the numeric value `99.9`. Otherwise it outputs the value of `Pot1`.
NOT	Syntax	`NOT(`*`logical`*`)` or *`!logical`*
	Example	`NOT(`*`Switch1`*`)` or *`!Switch1`*
	Description	If *`logical`* is TRUE, return FALSE, else return TRUE. In the example, if Switch1 is on, return FALSE, but if Switch1 is off, return TRUE. Do not use ! in front of parentheses; use `NOT` in that case.
OR	Syntax	*`logical1`* OR *`logical2`* OR *`logical3`*`...`
	Example	`Switch1 OR Pot1 > 50.0`
	Description	Returns TRUE if at least one logical parameter is TRUE. In the example, if either Switch1 is on or Pot1 is greater than 50.0, or both, the expression returns TRUE. If neither condition is true, it returns FALSE.
TIF	Syntax	`TIF(`*`logical`*`, `*`text result1`*`, `*`text result2`*`)`
	Example	`TIF(Switch1, "Text Message", TextInput1)`
	Description	Included for compatibility with early versions of Lookout. If *`logical`* is TRUE, return *`text result1`*, otherwise return *`text result2`*. Notice the use of quotation marks. In the example, if Switch1 is on, it outputs `Text Message`; otherwise, it outputs the current text value of `TextInput1`.

TRUE	Syntax	TRUE
	Example	TRUE
	Description	Returns the logical value TRUE. Accepts no parameters.
XOR	Syntax	*logical1* XOR *logical2* XOR *logical3...*
	Example	Switch1 XOR Pot1 > 50.0
	Description	Returns TRUE *if only one* logical parameter is TRUE. In the example, if Pot1 is greater than 50 and Switch1 is on, it outputs FALSE because both logical parameters are TRUE. If Pot1 is less than 50 and Switch1 is on, it outputs TRUE because only one logical parameter is TRUE.

Lookup Functions

CHOOSE	Syntax	CHOOSE(*numeric*, *value1*, *value2*, *value3*,...)
	Example	CHOOSE(Pot1, "Switch1", "TRUE", "Pb1")
	Description	Returns the value parameter corresponding to the integer portion of the numeric parameter. If the *numeric* parameter is less than 2.0, CHOOSE returns *value1*. If the *numeric* parameter is greater than the number of value parameters, CHOOSE returns the last logical parameter listed. In the example, if the value of Pot1 is .5, 1, or 1.4, it returns the value of Switch1. If Pot1 is 2.0 or 2.8, CHOOSE returns the logical value TRUE. If Pot1 is 3.0 or higher, CHOOSE returns the value of Pb1.
LCHOOSE	Syntax	LCHOOSE(*numeric*, *logical1*, *logical2*, *logical3*,...)
	Example	LCHOOSE(Pot1, Switch1, TRUE, Pb1)
	Description	Included for compatibility with early versions of Lookout. Returns the logical parameter corresponding to the integer portion of the numeric parameter. If the *numeric* parameter is less than 2.0, LCHOOSE returns *logical1*. If the numeric parameter is greater than the number of logical parameters, LCHOOSE returns the last logical parameter listed.

NCHOOSE	Syntax	NCHOOSE(*numeric*, *numeric1*, *numeric2*, *numeric3*,...)
	Example	NCHOOSE(Pot1, Pot2, Pot3, 14.3)
	Description	Included for compatibility with early versions of Lookout. Returns the *numericx* parameter corresponding to the integer portion of the numeric parameter. If the *numeric* parameter is less than 2.0, NCHOOSE returns *numeric1*. If the *numeric* parameter is greater than the number of *numericx* parameters, NCHOOSE returns the last *numericx* parameter listed.
TCHOOSE	Syntax	TCHOOSE(*numeric*, *text1*, *text2*, *text3*,...)
	Example	TCHOOSE(Pot1, "Auto", "Manual", "Local", "Locked")
	Description	Included for compatibility with early versions of Lookout. Returns the text parameter corresponding to the integer portion of the *numeric* parameter. If the *numeric* parameter is less than 2.0, TCHOOSE returns *text1*. If the *numeric* parameter is greater than the number of text parameters, TCHOOSE returns the last logical parameter listed.

Mathematical Functions

ABS	Syntax	ABS(*numeric*)
	Example	ABS(Pot1 – 50.0)
	Description	Returns the absolute value of *numeric*. In the example, if Pot1 is zero, the function returns 50.0.
EXP	Syntax	EXP(*numeric*)
	Example	EXP(Pot1)
	Description	Returns the base of the natural logarithm, *e* (2.71828182), raised to the power of *numeric*. In the example, if Pot1 is 2.0, the function returns 2.71828182^2 or approximately 7.38906. EXP is the inverse of the function, LN. To calculate the values of other powers, use the exponentiation operator (^).

FACT	Syntax	`FACT(`*numeric*`)`
	Example	`FACT(4.2)`
	Description	Returns the factorial of the integer portion of *numeric*. In the example, the function returns the factorial of 4, or 1×2×3×4, or 24. The factorial of zero, FACT(0), or any number less than zero is one (1).
INT	Syntax	`INT(`*numeric*`)`
	Example	`INT(4.2)`
	Description	Rounds *numeric* down to the nearest integer. In the example, the INT function returns 4. Notice also that INT(-8.5) = -9. See also *TRUNC*.
LN	Syntax	`LN(`*numeric*`)`
	Example	`LN(PLC.AI1)`
	Description	Returns the natural logarithm of *numeric*. In the example, if PLC.AI1 = 1000.0, the function returns 6.90776.
LOG	Syntax	`LOG(`*numeric1*`, `*numeric2*`)`
	Example	`LOG(Pot1,2)`
	Description	Returns the logarithm of *numeric1* to the *numeric2* base, where *numeric1* is a positive real number. In the example, if Pot1 = 8.0, the function returns 3.0.
LOG10	Syntax	`LOG10(`*numeric*`)`
	Example	`LOG10(Pot1)`
	Description	Returns the base-10 logarithm of *numeric*, where *numeric* is a positive real number. In the example, if Pot1 = 100.0, the function returns 2.0.
MOD	Syntax	`MOD(`*numeric1*`, `*numeric2*`)`
	Example	`MOD(50,8)`
	Description	Returns the modulus (remainder) of *numeric1* divided by *numeric2*, where *numeric2* is not equal to zero. In the example, 50 divided by 8 equals 6 with a remainder of 2. Therefore, the function returns 2.

PI	Syntax	`PI()`
	Example	`PI()`
	Description	Returns an approximation of PI: 3.1415927. Accepts no parameters.
PRODUCT	Syntax	`PRODUCT(numeric1, numeric2, numeric3,...)`
	Example	`PRODUCT(2,4,6,10)`
	Description	Returns the product of all the numerics. In the example, the function returns 2×4×6×10 or 480.
RAND	Syntax	`RAND()`
	Example	`RAND() * Pot1`
	Description	Generates a new random number between zero and one every time the formula that this function is a part of is recalculated. Accepts no parameters. In the example, every time the value of `Pot1` changes, the function generates a new random number and multiplies it by the value of `Pot1`.
ROUND	Syntax	`ROUND(numeric1, numeric2)`
	Example	`ROUND(Pot1,2)`
	Description	Rounds the value of `numeric1` to `numeric2` decimal places. In the example, if `Pot1` equals 15.745, the function returns 15.75. If `numeric2` equals zero, the function returns an integer. If `numeric2` is less than zero, the function rounds off the result to the corresponding place to the left of the decimal.
SIGN	Syntax	`SIGN(numeric)`
	Example	`SIGN(Pot1)`
	Description	Returns 1 if `numeric` is positive, 0 if `numeric` is 0, –1 if `numeric` is negative. In the example, if `Pot1` is –0.00001, the function returns –1.
SQRT	Syntax	`SQRT(numeric)`
	Example	`SQRT(ABS(Pot1))`
	Description	Returns the square root of `numeric`, where `numeric` is a positive real number. In the example, if `Pot1` is –25.0, the absolute function first converts the Pot value to positive 25 and the square root function calculates $\sqrt{25} = 5$.

SUM	Syntax	`SUM(numeric1, numeric2, numeric3,...)`
	Example	`SUM(2,4,-6,12)`
	Description	Returns the sum of all the numeric values listed. This function requires at least two numeric values. In the example, the function returns 12.
TRUNC	Syntax	`TRUNC(numeric)`
	Example	`TRUNC(8.9)`
	Description	Truncates `numeric` to its integer component by removing its fractional part. In the example, the TRUNC function returns 8. Notice also that `TRUNC(-8.9) = -8`. See also *INT*.

Statistical Functions

AVG	Syntax	`AVG(numeric1, numeric2, numeric3,...)`
	Example	`AVG(2,4,-6,12)`
	Description	Returns the arithmetic mean (average) of all the numerics listed. This function requires at least two numeric parameters. In the example, the function returns $(2+4-6+12)/4$ or 3.0.
MAX	Syntax	`MAX(numeric1, numeric2, numeric3,...)`
	Example	`MAX(2,4,-6,12)`
	Description	Returns the highest value of all the numeric values listed. This function requires at least two numeric values. In the example, the function returns 12.
MIN	Syntax	`MIN(numeric1, numeric2, numeric3,...)`
	Example	`MIN(2,4,-6,12)`
	Description	Returns the lowest value of all the numeric values listed. This function requires at least two numeric values. In the example, the function returns –6.

STDEV	Syntax	STDEV(*numeric1*, *numeric2*, *numeric3*,...)
	Example	STDEV(2.9,4.5,5.0,4.3,3.8)
	Description	Returns the sample standard deviation of the numeric values listed. (Standard deviation is a measure of dispersion, calculated as the positive square root of the variance.) This function calculates the standard deviation using the non-biased or $n-1$ method. This function requires at least two numeric values. In the example, the function returns 0.796869.
STDEVP	Syntax	STDEVP(*numeric1*, *numeric2*, *numeric3*,...)
	Example	STDEVP(2.9,4.5,5.0,4.3,3.8)
	Description	Returns the standard deviation of a full population of numeric values. This function calculates the standard deviation using the biased or n method. This function requires at least two numeric values. In the example, the function returns 0.712741.
VAR	Syntax	VAR(*numeric1*, *numeric2*, *numeric3*,...)
	Example	VAR(2.9,4.5,5.0,4.3,3.8)
	Description	Returns the sample variance of the numeric values listed. (Variance is a measure of dispersion.) This function calculates the variance using the non-biased or $n-1$ method. This function requires at least two numeric values. In the example, the function returns 0.635.
VARP	Syntax	VARP(*numeric1*, *numeric2*, *numeric3*,...)
	Example	VARP(2.9,4.5,5.0,4.3,3.8)
	Description	Returns the population variance of the numeric values listed. This function calculates the variance using the biased or n method. This function requires at least two numeric values. In the example, the function returns 0.508.

Text Functions

EXACT	Syntax	EXACT(*text1*, *text2*)
	Example	EXACT(*TextEntry1*, "batch a")
	Description	Returns TRUE if *text1* exactly matches *text2*, otherwise returns FALSE. This function is case sensitive. In the example, the function returns TRUE if the text result of TextEntry1 exactly matches batch a (notice that the entry in this example is all lowercase).
FIND	Syntax	FIND(*text1*, *text2*, *numeric*)
	Example	FIND("ON", "Reactor A is ON", 1)
	Description	Searches for *text1* within *text2* starting after *numeric* characters, and returns the position of the first character where the match begins. This function is case sensitive. The output of this function is numeric. It returns 0 if no match is found. In the example, the function searches the entire string for the word, ON and returns 14, indicating the position of the first character of the word. See also *SEARCH*.
FIXED	Syntax	FIXED(*numeric1*, *numeric2*)
	Example	"The flow is" & FIXED(Pot1,2)
	Description	Rounds the value of *numeric1* to *numeric2* decimal places and then converts *numeric1* to a text string. In the example, if the value of Pot1 equals 123.456, the FIXED function returns the text value 123.46. Thus, the entire text string would read The flow is 123.46. If *numeric2* is omitted, *numeric1* is rounded to the nearest whole number. For example, FIXED(123.588) returns 124. If numeric2 is negative, Lookout rounds numeric1 to the corresponding place to the left of the decimal. See also *TEXT*.
LEFT	Syntax	LEFT(*text*, *numeric*)
	Example	LEFT("Reactor A is ON", 9)
	Description	Retrieves the specified number of characters from the left end of *text* and outputs it as a text value. In the example, the function returns Reactor A. Use positive integers. If you set numeric less than 1, Lookout returns an empty string.

LEN	Syntax	LEN(*text*)
	Example	LEN("Reactor A is ON")
	Description	Returns the length of the text string (the number of characters in *text*). In the example, the function returns the numeric value 15.
LOWER	Syntax	LOWER(*text*)
	Example	EXACT(LOWER(*TextEntry1*), "batch a")
	Description	Converts *text* to all lower case. In the example, the LOWER function ensures that a match is found whether TextEntry1 content reads Batch A, BATCH A, or batch A.
MID	Syntax	MID(*text*, *numeric1*, *numeric2*)
	Example	MID("Reactor A is ON",9,7)
	Description	Retrieves *numeric2* characters from *text*, beginning at character *numeric1*. In the example, the function returns the text value A is ON. Use positive integers. If you set numeric1 less than 1, Lookout parses the input as if you entered 1. If you set numeric2 less than 1, Lookout returns an empty string.
PROPER	Syntax	PROPER(*text*)
	Example	PROPER("reactor A is ON")
	Description	Capitalizes the first character of each word in *text*. In the example, the function returns the text value Reactor A Is On.
REPLACE	Syntax	REPLACE(*text1*, *numeric1*, *numeric2*, *text2*)
	Example	REPLACE("Reactor A is ON", 9, 1, "B")
	Description	Replaces *numeric2* characters with *text2* beginning with character *numeric1* in *text1*. In the example, the function returns the text value Reactor B is ON. Use positive integers, with numeric2 greater than numeric1. If numeric1 is larger than the length of text1, Lookout appends text2 to the end of the text1 string. If you set numeric1 to a value less than 1, Lookout parses the value as if you entered 1, and the replacement string starts at the beginning of the text1 string.

REPT	Syntax	`REPT(`*text*`, `*numeric*`)`
	Example	`REPT("Lookout", Pot1)`
	Description	Repeats `text` `numeric` times. In the example, if the value of `Pot1` is 8, the function returns the text value, `Lookout Lookout Lookout Lookout Lookout Lookout Lookout Lookout`.
RIGHT	Syntax	`RIGHT(`*text*`, `*numeric*`)`
	Example	`RIGHT("Reactor A is ON", 7)`
	Description	Retrieves the specified number of characters from the right end of `text` and outputs it as a text value. In the example, the function returns the text value `A is ON`. Use positive integers. If you set `numeric` less than 1, Lookout returns an empty string.
SEARCH	Syntax	`SEARCH(`*text1*`, `*text2*`, `*numeric*`)`
	Example	`SEARCH("on", "Reactor A is ON", 1)`
	Description	Finds `text1` within `text2` beginning at `numeric` character, and returns the position of the first character where the match begins. This function is not case sensitive. The output is numeric. It returns 0 if no match is found. In the example, the function searches the entire string for the word `ON`, `on`, `On`, or `oN` and returns 14, indicating the position of the first character of the word. See also *FIND*.
TEXT	Syntax	`TEXT(`*numeric*`, `*text*`)`
	Example	`TEXT(Pot1, "0.00")`
	Description	Included for compatibility with early versions of Lookout. With polymorphic data, this function becomes unnecessary. Like `FIXED`, the `TEXT` function converts `numeric` to a textual value. While `FIXED` allows you to specify the number of decimal points, `TEXT` allows you specify a desired numeric format in the text parameter. If `Pot1` equals 12.3456, the function would return the textual value `12.35`. See also *FIXED*.

TRIM	Syntax	`TRIM(text)`
	Example	`TRIM("Reactor   A   is ON")`
	Description	Trims multiple spaces from between words. In the example, there are multiple spaces on either side of the word, *A*. This function returns `Reactor A is ON`, eliminating all repeated spaces.
UPPER	Syntax	`UPPER(text)`
	Example	`UPPER(text)`
	Description	Converts `text` to all capital letters (upper case). In the example, the `UPPER` function ensures that a match is found whether `TextEntry1` content reads `Batch A`, `BATCH A`, or `batch A`.

Trigonometric Functions

ACOS	Syntax	`ACOS(numeric)`
	Example	`ACOS(Pot1)`
	Description	Returns the arccosine of `numeric` (that is, it returns the angle of the cosine you specify as `numeric`). *Numeric* must range between –1 and 1. The result is output in radians and ranges from 0 to π. In the example, if `Pot1` = 0.5, the function returns 1.0472 radians (60 degrees). You can express the arccosine in degrees by multiplying the result by $180/\pi$.
ASIN	Syntax	`ASIN(numeric)`
	Example	`ASIN(Pot1)`
	Description	Returns the arcsine of `numeric` (that is, it returns the angle of the sine you specify as `numeric`). `numeric` is the sine of the angle and must range between –1 and 1. The resulting value is given in radians and ranges from $-\pi/2$ to $\pi/2$. In the example, if `Pot1` = –1, the function returns –1.5708.

ATAN	Syntax	`ATAN(numeric)`
	Example	`ATAN(Pot1)`
	Description	Returns the arctangent of `numeric` (that is, it returns the angle of the tangent that you specify as `numeric`). The resulting value is given in radians and ranges from $-\pi/2$ to $\pi/2$. In the example, if `Pot1` = 180, the function returns 1.56524 radians (about 90 degrees).
ATAN2	Syntax	`ATAN2(numericX, numericY)`
	Example	`ATAN2(5, 5)`
	Description	Returns the arctangent of the specified x- and y-coordinates (that is, it returns the angle of a line extending from the origin (0,0) to a point specified by the `numericX`, `numericY` coordinate pair that you specify). The resulting value is given in radians and ranges from greater than $-\pi$ to π. In the example, the function returns 0.785398 radians (45 degrees).
COS	Syntax	`COS(numeric)`
	Example	`COS(Pot1)`
	Description	Returns the cosine of `numeric` where `numeric` is the angle in radians. In the example, if `Pot1` = 1.047, the function returns 0.500171. You convert degrees to radians by multiplying by $\pi/180$.
SIN	Syntax	`SIN(numeric)`
	Example	`SIN(Pot1)`
	Description	Returns the sine of `numeric` where `numeric` is the angle in radians. In the example, if `Pot1` = 3.14159, the function returns 1.2246E–16 (effectively zero). You convert degrees to radians by multiplying by $\pi/180$.
TAN	Syntax	`TAN(numeric)`
	Example	`TAN(Pot1*PI()/180)`
	Description	Returns the tangent of `numeric` where `numeric` is the angle in radians. In the example, the value of `Pot1` is 45, but it is in degrees, not radians. Convert it to radians by multiplying it by $\pi/180$. In this example, the function returns 1.

Date/Time Functions

NOW	Syntax	`NOW(logical)`
	Example	`NOW($Keyboard.F1)`
	Description	Returns a numeric value representing the current system date and time when any signal within the parentheses changes. The result is a floating point number in which the integer represents the date and the fraction represents the time of day. In the example, when you press the function key <F1>, the date and time are output as a single number, like 34738.3. If you change the **Numeric Format** of the number to `mm/dd/yy hh:mm:ss`, the same number would be shown as `02/08/95 07:49:02`.
TODAY	Syntax	`TODAY(logical)`
	Example	`TODAY($Keyboard.F1)`
	Description	Returns a numeric value representing the current system date when any signal within the parentheses changes. The result is an integer that represents the number of days that have passed since Jan. 1, 1900. In the example, when you press the function key <F1>, the date is output as a single number, such as `34738`. If you change the **Numeric Format** of the number to `mm/dd/yy`, the same number would be shown as `02/08/95`.

Note If you want to display the current time only, subtract the TODAY function from the NOW function.

Example: `NOW($Keyboard.F1) – TODAY($Keyboard.F1)`

In this example, if you want the result to update itself every second, replace `$Keyboard.F1` with a one second pulse timer.

Quality Functions

When you display data on a control panel directly through an expression, or when an object with a displayable element is connected to some other object and receives bad data, the quality problem is indicated by a red X superimposed over the display.

Some data members might not be displayed or may not connect to an object on some particular panel, but you may still need to monitor the quality of that data. You can use the qgood and qbad expression functions to check

for quality. You can use the qtext function to report more specific information about the quality problem.

qtext	Syntax	qtext(*expression, [delimiter]*)
	Example	qtext(modbus.40001, "AND")
	Description	Returns a text string specifying what elements of data quality have gone bad.
qgood	Syntax	qgood(*expression*)
	Example	qgood(modbus.40001)
	Description	Returns TRUE while the expression is good.
qbad	Syntax	qbad(*expression*)
	Example	qbad(modbus.40001)
	Description	Returns TRUE while the expression is bad.

8

Logging Data and Events

This chapter describes how to log data and events from Lookout.

Refer to Chapter 9, *Alarms and Events*, for more information about how alarms are logged to disk.

Logging to the Citadel Database

You can use the Citadel database to log historical data to disk in a compressed format. You can then view or retrieve the information using the HyperTrend object, SQL, or the Historical Data Viewer in Measurement and Automation Explorer (MAX).

How Lookout Logs Data

Lookout logs data only when the value of a point changes, *not* at a timed interval. The logged data accurately reflects the actual behavior of the point being logged because the same value of a point is not recorded over and over.

Citadel stores historical information using traces. A *trace* refers to the line of continuity for a specific data source. A trace connects all the historical values for a given I/O point, which can be displayed as a continuous line on a HyperTrend. If Lookout is unexpectedly interrupted or a data member is temporarily modified to *not* log to disk, a trace can be broken. If the trace is broken, the HyperTrend plots the trace as a continuous line with void sections to represent gaps in the data. You cannot remove individual traces from Citadel, but you can add new traces by configuring a new point.

Because Citadel uses Coordinated Universal Time (UTC) to time stamp the data, you do not need to compensate for data logged in different time zones. The HyperTrend object automatically converts from universal to local time before displaying a data history.

Note When your operating system switches in or out of Daylight Savings time, Lookout corrects for the change relative to universal time so that there is no data discontinuity or loss in the Citadel database.

When you install Lookout, the installer creates an ODBC data source for the default location of your Lookout data in Citadel. If you install Lookout in a custom location, or log data to a location other than the installation default location, you will have to create an ODBC data source for that location. Refer to the *Creating a Citadel ODBC Data Source* section of Appendix B, *Structured Query Language*, for more information about creating an ODBC data source.

Choosing a Default Data Logging Location

Choose **Options»System** to view or change the default Citadel logging location, which is shown in the **Citadel Database** fields of the System Options dialog box. The default is for Citadel to log to the database subdirectory in the Lookout directory. This subdirectory holds your historical data files.

If you move or rename the historical data subdirectory or files, you must redirect Lookout processes and external tools to that new path in order to ensure data continuity.

If you change the default directory from within Lookout, the change does not take effect until you restart Lookout.

Unlike spreadsheet files, only one set of Citadel files exists for each Lookout process. However, you can set up separate Citadel locations for multiple processes on one computer. Refer to the next section for more information.

Changing a Process Data Logging Location

When you create a process in Lookout, you have an opportunity to set the default data logging location for that process in the Create Process dialog box. To access this dialog box at a later time, select **File»Modify Process**.

By default, the **Use default values** checkbox is selected. This option directs Lookout to log data from this process to the default location set in the System Options dialog box. With this option, you can create a process that logs to the default data location of any copy of Lookout running on any computer that might later run this process.
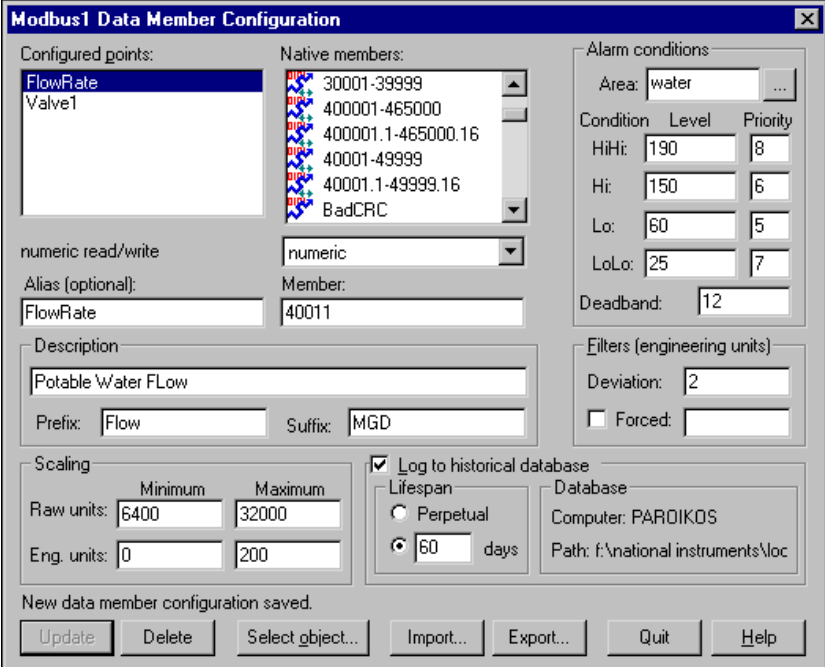
To log data to a different location from your process, deselect the **Use default values** option and enter the computer and path to which you want data logged. You can also direct data to a location on a remote computer, as long as that computer is running the Lookout Citadel service.

Although you can direct data from different processes to different logging directories, proliferating a variety of databases with large numbers of processes running in one instance of Lookout can degrade performance, depending on the number of data points being logged.

Configuring Data Members for Logging

You can log any data member of any object to the Citadel database. By default, data members are *not* logged. Use the following procedure to configure a data member to log to Citadel.

1. Select **Object»Configure Data Members**, navigate to the object that contains data member you want to log, and click **OK**. A dialog box similar to the following appears.

2. Select the data member you want to log from the **Native members** field. If the value was previously configured, select it from the **Configured points** list. The data member then appears in the **Member** data field.

   If **Alias** is not specified, Lookout uses the native data member name as displayed in the **Member** field. If an alias is assigned to the native data

member, Lookout uses the **Alias** name instead. The name used for the trace is the same name saved to the **Configured points** list box.

3. Check the **Log to historical database** checkbox and choose an appropriate **Lifespan**. Lookout needs to know how long you want to maintain this trace of data in Citadel. Lookout maintains the data for the time span specified in this field.

4. Enter a value in the **Deviation** field. If a change in a numeric value surpasses or equals the **Deviation** setting, Lookout saves a new value to Citadel. If you implement **Scaling** parameters, Lookout compares the **Deviation** setting to changes in the **Eng. units** values to determine when to log a new value. If you leave the **Scaling** blank, Lookout compares the **Deviation** setting to the raw (unscaled) signal. The **Deviation** parameter is not available on logical signals, because Lookout logs all state transitions of logical values.

   If **Deviation** is not specified, *any* change of a numeric value results in a new value being logged to disk, and you might log more data than you need. If the **Deviation** setting is too large, very little information might be saved to disk.

5. Click the **Save/Update** button.

6. Repeat these steps to configure logging for any other data members. You can click the **Select object** button in this dialog box to configure logging for another object's data members.

7. Click **Quit** when you are done.

Logging Intermittently or on an Event

You can use the Logger object to log data on command or at set times, or to start and stop continuous logging. Refer to the *Logger* object topic in the *Lookout Help* for more information about logging under special circumstances with this object.

Disk Space Required for Logging

Because Lookout creates and maintains the historical data files in 1 MB units, you must have a least 1 MB of available disk space to begin logging data. When the first 1 MB file fills up with data, Lookout creates a second 1 MB file in the same directory. When the second fills up, Lookout continues to create 1 MB files until all traces can be maintained on disk for the data lifespan you have specified. Refer to the *Configuring Data Members for Logging* section for more information about data lifespan settings.

Lookout monitors the status of your hard drive and generates an alarm when your disk falls below 500 KB of available space.

You might be able to reduce the amount of logging space needed by adjusting the **Deviation** setting on each logged data member. Refer to the *Configuring Data Members for Logging* section for more information about deviation settings.

Refer to the *Preventing Event Information Overload* section later in this chapter for information about using less disk space for logging events.

Copying, Moving, Deleting, and Archiving Historical Data Files

You can use the Historical Data Viewer in MAX (National Instruments Measurement and Automation Explorer) to copy, move, delete, or archive Citadel historical data files. MAX comes with Lookout, and you must have it installed to use the Historical Data Viewer. Select **Options»Historical Data Viewer** in Lookout, to launch MAX. Then select the **Historical Data** category in the left tree view, and refer to the online help in MAX for more information.

You can use the CitadelControl object to archive and restore your Citadel data files. You can automate archiving so that it takes place on a regular basis. You can use CitadelControl to archive data from several locations into one archive location or restore data to active logging locations where data has been corrupted or lost. Refer to the *CitadelControl* topic in the *Lookout Help* for more information.

Retrieving Logged Historical Data

After historical data is logged to disk, you can view it with a HyperTrend object, SQL, HyperTrend ActiveX object, or with the Historical Data Viewer in MAX. Logged text can be accessed only through an SQL query.

A HyperTrend serves as windows into your historical data. Refer to the *HyperTrend* topic in *Lookout Help* for more information about the HyperTrend object.

SQL, an industry-standard language supported by most database packages, enables other applications to directly retrieve data from Citadel. Refer to Appendix B, *Structured Query Language*, for more information about using SQL queries.

You can also view historical data with the Historical Data Viewer in MAX. To launch the Historical Data Viewer, select **Options»Historical Data Viewer** in Lookout.

Logging to a Spreadsheet

You can use a Spreadsheet object to store real-time data to disk. The Spreadsheet object creates standard ASCII text files in comma separated value (.csv) file format that you can later open with software packages such as Excel, Lotus, and Foxpro. Refer to the *Spreadsheet* object topic in the *Lookout Help* for more information.

Logging Events

When you want to create a chronological schedule of events or an audit trail, log Lookout events. You can log all events: operator commands, Event objects, and system events such as running Lookout, closing Lookout, opening a process file, or closing a process file.

Lookout system events, such as logging on and logging off, or entering and exiting edit mode, are automatically logged as events by Lookout. Other events, such as adjusting the value of a Pot object, are only logged if you select the **Log events** option when you create or modify the object.

When Lookout logs an event, it also logs the account name (the operator), date and time of the event, name of the object adjusted, and the previous and subsequent settings of the object. Because all of this information is logged, you can create an exhaustive audit trail for any specific workstation.

Use the Event object to define and log your own event messages based on a user-defined trigger. Refer to the *Event* topic in the *Lookout Help* for more information about using the Event object.

Retrieving Event Information

You can access logged event information in several ways such as printing them by selecting **Alarms»Print** or by using the Historical Data Viewer in MAX.

Preventing Event Information Overload

Because every time you flip a Switch, adjust a Pot, or press a Pushbutton, Lookout logs the event to disk, you might collect more data than you need. By selectively modifying certain control objects to disable the **Log events** checkbox, you cut down on the amount of information logged to the event file.

You can also change certain default parameters to prevent the **Logging buffer is full** error message, by modifying your `Logos.ini` file. If you do not have this file on your computer, you can create it and save it in `C:\WINNT` (Windows 2000/NT/XP) or `C:\WINDOWS` (Windows Me/98).

Using a text editor, make the following entries to the file:

```
[Citadel]
MaxQueueSize=### in points
LogPeriod=### in milliseconds
```

`MaxQueueSize` is the number of points to hold in the buffer. The default is 500, but setting it up to as high as 5000 can prevent overload.

`LogPeriod` is how often the logging buffer is flushed to the database. The default is 200 ms. If you are logging many data members, this can be very slow, because Lookout must get the timestamp from each point before logging to the database. Setting this to as high as 2000 (2 seconds) or more will help free up the processor.

✎ **Note**   You can also get the **Logging buffer is full** error if Lookout is unable to connect to the Citadel database. This can happen if the Citadel service is not running on the computer on which the database resides (usually the local computer), or if that computer is not accessible from the local computer.

9

Alarms and Events

This chapter describes Lookout alarms and events.

An *event* can be anything that happens within the Lookout environment, such as adjusting a control, entering or leaving edit mode, or logging in and out. An *alarm* in Lookout denotes an abnormal condition and must be acknowledged by the operator. You can define alarm conditions for objects and data members. For the purposes of logging and retrieval, events and alarms are combined.

Lookout keeps track of error messages and any alarm conditions that you have defined for your process elements. You can define alarm conditions for most Lookout objects and data members. You can filter, display, log, and print alarms, as well as organize alarms into hierarchal areas. All Lookout alarms are automatically logged to Citadel. You can view alarms and events in the Alarms Window.

Defining Alarm Conditions

Alarm conditions can be generated automatically by an object, they can occur due to alarm settings you have defined for a data member, or they can be generated by an Alarm object.

Alarms Triggered by Data Member Values

If values in your Lookout process exceed allowable conditions, you can have Lookout generate an alarm by defining alarm conditions for certain data members. This method offers the most efficient way to define standard or simple alarm conditions. Furthermore, this method is especially useful when you configure alarms for data that originate from objects with a large number of data members, like driver objects and DataTable objects.

Complete the following steps to view or change the alarm settings for a data member.

1. Select **Object»Edit Data Member Configuration**, navigate to the object whose alarm settings you want to modify, and click **OK**. Or, you can right-click the object in the Object Explorer. A dialog box similar to the following appears.

PLC_Simulator Data Member Configuration

Configured points:
- amplitude
- offset
- period
- phase

Native members:
- A.txt.1-IV.txt.64
- A1-IV10000
- A1.logical-IV10000.logic
- A1.txt-IV10000.txt
- cursor.1-cursor.64
- enable

numeric read/write   numeric

Alias (optional):   Member:
Tank1Level   A1

Alarm conditions

Area: Tank1

Condition	Level	Priority
HiHi:	95	8
Hi:	88	6
Lo:	20	7
LoLo:	10	9

Deadband: 4

Description

Prefix: Suffix:

Filters (engineering units)

Deviation:

☐ Forced:

Scaling

	Minimum	Maximum
Raw units:		
Eng. units:		

☐ Log to historical database

Lifespan
- ○ Perpetual
- ◉ 60   days

Database
Computer: PAROIKOS
Path: f:\national instruments\loc

[Save] [Delete] [Select object...] [Import...] [Export...] [Quit] [Help]

2. Set the alarm conditions appropriately for any numeric or logical data member. Refer to the *Editing Data Member Configuration* section of Chapter 5, *Creating Lookout Processes*, for more information about modifying alarm settings for data members.

3. Click the **Save/Update** button when you are done.

Alarms Triggered by Events or Conditions

You can create an Alarm object to define an alarm condition based on an event or condition. For example, you can create alarms that are triggered by any event or combination of events you specify in the Create Alarm dialog box using as simple or complex an alarm criterion as needed. You do not have to use the traditional set of alarm conditions. Refer to the *Alarm* topic in the *Lookout Help* for more information about using the Alarm object.

Alarms Generated by Objects Automatically

Most objects generate alarm signals through their data members. Some objects, such as the Spreadsheet object, can generate an alarm without having a data member configured. Driver objects can generate communication alarms and pass alarms back from their hardware as well.

Several objects generate alarms warning about circular references.
A circular reference defines a condition where the signal generated by an object is sent back to that object as a parameter, either directly or indirectly. Circular references are always priority 10 alarms, and you should correct them during process file testing.

Configuring Alarm Areas

For organizational purposes, you can classify alarms into areas. Alarm areas allow operators to filter unwanted alarms and acknowledge alarms on an area-by-area basis.

You can create any number of alarm areas and assign any number of alarms to any area. However, you should carefully plan your alarming structure so operators can filter alarms by meaningful categories. With a carefully planned structure, you can handle specific areas and/or priorities if your system experiences a large number of alarms. Also, if you move an alarm from one area to another at a later time, you would have to search both the old and new alarm areas for historical information about that alarm.

When you create an alarm area, Lookout creates a folder in the global $Alarm object to hold those alarms. These folders make browsing for alarm areas more convenient when setting filters.

When you first start Lookout, it has one default alarm area called Lookout. There are also a number of alarm areas built into Lookout that become visible when you create or open a process that uses one of the objects associated with those areas. Table 9-1 lists the various alarm areas.

Table 9-1. Lookout Alarm Areas

Alarm Area	Description
Lookout	Lookout default alarm area. Includes alarms not directed to a specific alarm area.
Serial Ports (comm)	Alarms involving serial ports and serial communication.
Citadel	Alarms concerning the Citadel database.
DDE	Alarms generated by any DDE object or action.
Disk	Disk alarms.

Table 9-1. Lookout Alarm Areas (Continued)

Alarm Area	Description
Math	Mathematical errors, such as an attempt to divide by zero.
Print	Printer alarms.

You can create your own alarm areas by entering a new alarm **Area** name in the **Alarm conditions** section of the dialog box used to edit data member configuration. That new alarm area becomes a part of your Citadel data files and appears in the alarm area hierarchy.

Setting Alarm Priorities

Each alarm has an associated priority level ranging from 1 to 10, where 1 is the lowest priority and 10 is the highest priority. You can set priority levels using your own standards of severity, though certain priority levels have some predefined characteristic in Lookout.

Priority 1 alarms do not require operator acknowledgment. When they are deactivated, priority 1 alarms are removed from the Alarm Window by Lookout, regardless of acknowledgment. You should assign priority 1 to only minor, inconsequential alarms. Often, priority 1 alarms do not exist in control strategies. If an occurrence is significant enough to be classified as an alarm condition, you should probably assign a priority level that requires operator acknowledgment.

Priority 2 and higher alarms require operator acknowledgment before they are removed from the alarm list.

Selecting Processes to Monitor for Alarms

Before you can monitor alarms or events from processes running on other computers, you must add them to the list of processes that your monitoring computer is tracking. Complete the following steps to select other processes to monitor for alarms.

1. Select **Alarms»Select Processes**. The following dialog box appears.
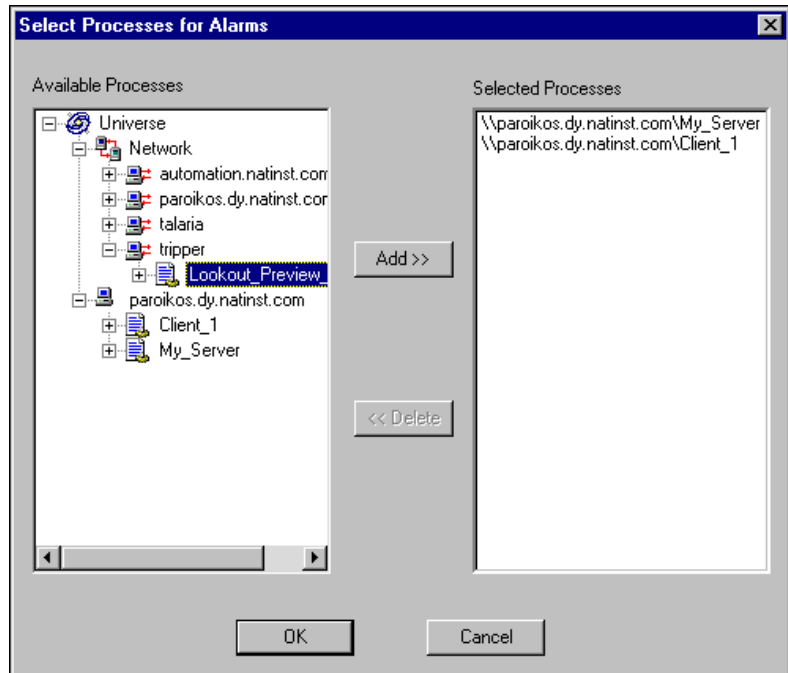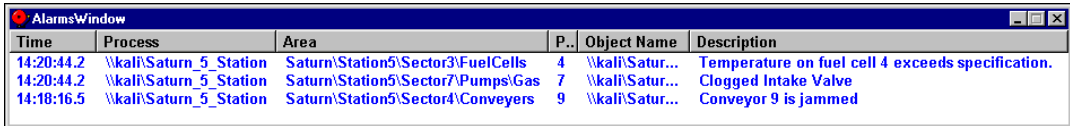
2. Navigate to the process you want to monitor in the **Available Processes** list, click the process, and click the **Add** button. (If you do not see the computer that is running the process you want to monitor, right-click the Network node, and select **Register Computer**.)

3. To stop monitoring a process, select the process in the **Selected Processes** list and click the **Delete** button.

4. Click **OK**.

Viewing Alarms and Events

You can view recent or active alarms and events in the Alarms Window in Lookout. You can also view historical alarm and event information that has been logged, using the Historical Data Viewer in MAX (National Instruments Measurement and Automation Explorer).

Viewing Alarms and Events in the Alarms Window

The Alarms Window lists all active and unacknowledged alarms and/or events. Alarms are listed in chronological order with the most recent alarm at the top of the list. If there are too many alarms to see at once, you can use the scroll bar at the right of the **Alarms Window** to scroll through the list.

If you have your Alarms Window minimized, you can view it by pressing <Ctrl-A>, selecting **Alarms»Show**, or clicking the alarm indicator box on the far right side of Lookout status bar, at the bottom of the screen. The number of alarms and events currently shown in the Alarms Window is displayed in the alarm indicator box.

To quickly identify alarm status, use the Lookout Alarms Window color scheme, as defined in Table 9-2.

Table 9-2. Alarms Window Color Scheme

Color	Alarm Status
Red	Active
Blue	Unacknowledged, Inactive
Red and Black	Acknowledged, Active (Alarm information, including time, description, priority, and name, appears in black and the area appears in red)
Green	Event

Because a new line is added to the list every time an alarm activates or event occurs, Lookout might list the same item multiple times.

You can sort the alarms and events by clicking the column headers in the Alarms Window. To view detailed information about a particular alarm or event, right-click it and select **Properties**. The following dialog box appears, listing specific information about the alarm or event.

You can scroll through alarms and events using the **Previous** and **Next** buttons.

Right-clicking in the **Alarms Window** displays the **Alarms** menu, which includes options you can use to acknowledge alarms and access all other alarm properties.

Setting Alarm Display Options

With the **Alarms»Display Options** command, you can change the display style of the Alarms Window. The Alarm Display Options dialog box appears, as shown in the following figure.
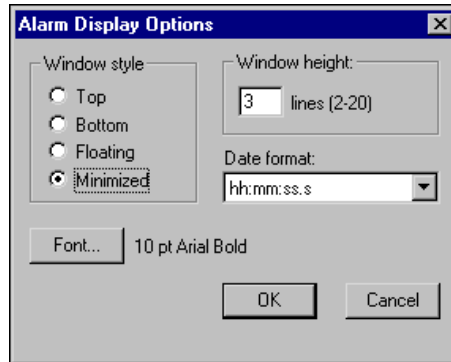
- **Window style**—Determines the position of the Alarms Window in the Lookout workspace. If you select **Floating**, it appears as a pop-up style control panel that you can resize, move around, and minimize at any time. If you use either the **Top** or **Bottom** window style, the **Window height** specifies the number of items Lookout can display in the Alarms Window. The actual height of the window adjusts automatically depending on the selected font and **Window height** setting. If more alarms occur than can be displayed in the window at once, a scroll bar appears along the right side of the window.

- **Date format**—Set the date format using the **Date format** selection box. Refer to the *Time and Date Values* section of Appendix A, *Numeric, Time, and Date Formats*, for more information about time and date formats in Lookout.

Filtering Alarms in the Alarms Window

To filter the alarms and events that appear in your Alarms Window, select **Alarms»Filter Options**, or right-click in the **Alarms Window** and select **Filter Options**. The following dialog box appears.

- **Priority**—To monitor alarms with specific priorities, check this box and set the **Min** and **Max** criteria.

- **User Name**—Check this box and enter a user account name, to restrict your display to alarms and events generated while that particular user is logged on. You can select only one user name at a time, but you can use wildcard characters (*, ?, #) to widen the scope.

Tip The wildcard characters supported are * (substitutes for zero or more characters), ? (substitutes for any single character), and # (substitutes for any single numeral).

- **Object Name**—Check this box and enter an object name, including the full path, to restrict your display to alarms and events involving that particular object. You can only enter one name at a time, but you can use wildcard characters (*, ?, #) to widen the scope of those objects reported.

- **Description**—Check this box and type a description to restrict your display to alarms and events that meet your criteria. You can filter on only one description at a time, but you can use wildcard characters (*, ?, #) to widen the scope of the alarms reported.

- **Area Name**—Check this box and enter an area name to restrict your display to only alarms in the alarm area you choose. You can enter only one alarm area at a time. Click the **Browse Areas** button to locate and select the alarm area you want to use as a filter.

- **Show**—You can choose to have the Alarms Window show alarms only, events only, or both alarms and events.

- **Old Alarms**—This option allows you to display alarms after they have been acknowledged. **Number** specifies the number of acknowledged alarms to show. Any alarms older than **Start time** are not displayed. You can enter **Start time** using any time format supported in Lookout.

- **Audible Alarms**—Check this box to enable a sound alert when an alarm takes place. The sound depends on your Windows system setting for error sounds.

✏️ **Note** If you use international characters in any filter options other than **Description**, no alarms will match.

Printing Alarms and Events

You can print alarms and events manually, or you can set up your process to print them as they happen.

Printing Alarms and Events Manually

You can print alarms and events in Lookout, based on your filtering.
To print alarms, select **Alarms»Print**. The following dialog box appears.
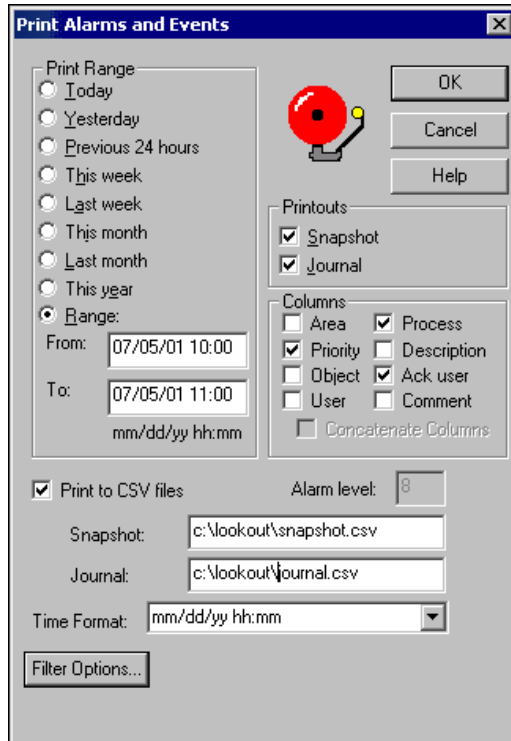
- **Print Range**—Select the time range you want to print alarms and events from with the items in the **Print Range** section of the dialog box. Notice that when you define your own range, you use month, day, and year, followed by hour and minute.

- **Printouts**—Determines the exact alarm information included in your printout. **Snapshot** prints only the status of alarms at the beginning of the specified **Range** but does not indicate what happened during the time span. **Journal** creates a printout of everything that happened during the time span from the beginning of the **Range**.

- **Columns**—Specifies which columns you want printed. Specific information about each alarm is presented in columnar format, and Lookout prints only the information you designate.

- **Print to CSV file**—Prints the results to a comma separated file (.csv). Enter the file names for your **Snapshot** and **Journal** files, including a complete path to where you want the files written. If you enter a file name only, Lookout will create the file in the Lookout directory.

- **Alarm level**—Sets the alarm level for the error generated if there is a problem printing to a `.csv` file. You must be in edit mode with **Print to CSV files** selected in order to change this setting.

- **Time Format**—Sets the format for printing times.

- **Filter Options**—You can adjust your filter settings by clicking this button. The New Alarm Filters dialog box appears with the same options available when filtering alarms, plus the following two additional **Filter Criteria** options:

  – **Ack User Name**—Check this box and enter a user account name to restrict your display to alarms and events acknowledged by that particular user. You can only enter one user name at a time, but you can use wildcard characters (`*`, `?`, `#`) to widen the scope.

  – **Ack Comment**—Check this box and type a comment to restrict your alarms displayed to those with the specified acknowledgement comment.

📝 **Note**   If you use international characters in any filter options other than **Description**, no alarms will match.

Printing Alarms and Events as They Happen

You can print alarms as they happen. Complete the following steps to print the contents of the Alarms Window using the filter criteria you have configured:

1. Select **Options»System**. The System Options dialog box appears.

2. Specify a printer port in the **Log alarms to** field of the dialog box.

3. Click **OK**.

This method works well for a printer directly connected to your computer that has a line-by-line option. Some printers may queue an entire page of data before printing. To print alarms directly to a network computer, you must capture a port in the network printer driver and link it to your networked printer. Refer to your operating system documentation for detailed instructions on how to capture a port for a printer driver.

Acknowledging and Clearing Alarms

Complete the following steps to acknowledge an alarm or event in the Alarms Window.

✏️ **Note** The security level for alarm acknowledgement is set in the System Options dialog box, accessed by selecting **Options»System**. Refer to the *System Security Settings* section of Chapter 10, *Security*, for more information about setting security levels.

1. Select an alarm in the Alarms Window. <Ctrl-click> to select multiple alarms; <Shift-click> to select blocks. Right-click in the **Alarms Window** and choose **Select All** or **Acknowledge All** to select all alarms for acknowledgment.

2. Right-click in the **Alarms Window** and choose **Acknowledge**. A dialog box appears for you to enter a comment.

3. (Optional) Enter a comment, as shown in the following illustration. Comments are optional, and you can click **OK** to finish acknowledging alarms without entering a comment. However, you can search historical data for alarms, or print out the alarms, based on comments. So, having your operators use certain standard comments or comments on specific circumstances can be beneficial.
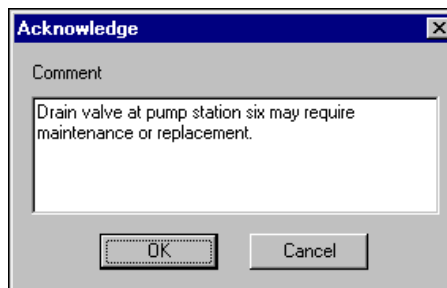
When an alarm is inactive and has been acknowledged, Lookout removes it from the Alarms Window.

Acknowledging Alarms Programmatically

By connecting to the appropriate data members of the global $Alarm object, you can acknowledge alarms programmatically with Pushbuttons or logical expressions. This way, you can bypass the menu commands, or acknowledge alarms from remote Lookout network nodes. For instance, you could create a single Pushbutton that acknowledges all alarms, or you

could create multiple Pushbuttons for acknowledging specific areas of alarms. Refer to the *$Alarm* topic in the *Lookout Help* for more information about using the $Alarm object.

Silencing Audible Alarms

To permanently silence alarms, select **Alarms»Filter Options** and uncheck the **Audible Alarms** checkbox. If the **Audible Alarms** checkbox is checked, you can temporarily silence alarms by selecting **Alarms» Silence** or pressing <Ctrl-S>, but new alarms will produce an audible beep.

Data Quality Problems

When a *red X* appears over a display or control on a panel, there is a data quality problem. You can hover over the red X in run mode to see a description of the problem in the middle of the status bar. You can also use the data quality functions to monitor data quality and report problems. Refer to the *Quality Functions* section of Chapter 7, *Expressions*, for more information about these functions.

10

Security

This chapter describes the types of Lookout security: network security, control security, and system security.

Network security determines how other computers on the network can read from or write to your process. Network security is implemented using permissions and should be implemented in your server process.

Control security uses access levels defined in the User Account Manager to determine who can manipulate a control in your Lookout process. Control security can be implemented in both server and client processes.

System security uses access levels defined in the User Account Manager to determine who can perform certain tasks within the Lookout environment. System security settings are made in the System Options dialog box.

Creating and Editing User Accounts

You use the User Account Manager to create and edit the properties of groups, create or edit the properties of user accounts, assign users to one or more groups, and otherwise manage security accounts for Lookout applications. Only an Administrator or someone whose account is a member of the Administrator group can create, revise, or delete system user accounts.

Note User Account Manager does not support international character sets, so user names, group names, and passwords must not include international characters.

Note For user accounts to work consistently across your network, you must use the same `lookout.sec` file for all installed copies of Lookout. Refer to the *Lookout Security Files* section later in this chapter for more information.

Creating User Accounts

✎ **Note**   If you are logged on to a Windows 2000/NT/XP computer as Guest or Restricted User, certain Lookout tasks will not work correctly, because they require writing to the registry or `WINNT` folder. These tasks include creating a user account, adding a client license, and registering a Logos computer.

1. Select **Options»User Manager** to open the User Account Manager.

2. Select **User»New User**.

3. Enter the domain name of the new user in the **Username** field.

4. Enter the **Full Name** of the user.

5. Enter job titles or other relevant information in the **Description** field.

6. Enter the user password in the **Password** field.

7. Enter the password a second time in the **Confirm Password** field to make sure there was no typing error in the first entry.

8. Set the **Security Level** for the new user. Security levels range from 0 to 10, with 10 being the highest possible security authorization.

   You should carefully consider users and the security level you assign each one. Assign level 10 access only to those people responsible for system security. By default, users with security levels of 8 and higher can close process files and exit Lookout. Only users with security level 9 or higher can edit process files in development versions of Lookout. Refer to the *Lookout Menu Commands* topic in the *Lookout Help* for more information.

9. **Minutes idle until logoff** sets how long Lookout runs with no operator interaction before logging the current user off automatically. Setting this value to 0 (the default) means there is no timeout in effect.

10. You can set an expiration time for passwords. Users cannot reset their own password; a member of the Administrator group must set the password for them. The default is for passwords never to expire.

11. Place a checkmark in the **Account Disabled** checkbox if you want to disable a user account without removing the user from the system.

12. Click the **Groups** button to add this user to various local security groups. The Group Memberships dialog box appears.

    The default groups are Administrators, Guests, Operators, and System Operators. Any groups you have created are also shown.

13. To enter a user in a group, highlight the group in the **Not Member of** field and click the **Add** button. To remove a user from membership in a group, highlight a group in the **Member of** field and click the **Remove** button.

Note   When you add an individual user whose individual account has a security level different than that of the group, that user has the higher of the security levels.

14. Click the **OK** button.

Creating Groups

1. Select **Options»User Manager** to open the User Account Manager.

2. Select **User»New Local Group**.

3. Assign a name to the group in the **Group Name** field.

4. Enter a description of the group in the **Description** field.

5. Assign the security level for members of this group in the **Security Level** field.

   You should carefully consider groups and the security level you assign each one. Assign level 10 access only to those groups responsible for system security. Groups with security levels of 8 and higher can close process files and exit Lookout. Groups with security level 9 or higher can edit process files in development versions of Lookout. Refer to the *Lookout Menu Commands* topic in *Lookout Help* for more information.

Note   When you add an individual user whose individual account has a security level different than that of the group, that user has the higher of the security levels.

6. To add **Members**, click the **Add** button. The Add Users and Groups dialog box appears.

7. The **List Names From** listbox selects the domain to list user names from. At this time, you are restricted to your local domain.

8. Highlight the names you want to add in the **Names** field, and click the **Add** button to add those users to your group.

Modifying User and Group Accounts

The dialog boxes for editing users and groups are essentially the same as those for creating users and groups. Complete the following steps to modify user and group accounts.

1. Select **Options»User Manager** to open the User Account Manager.

2. Either double-click the user or group you want to edit, or highlight the user or group and select **User»Properties**. The User Properties dialog box appears and displays information about user activity.

3. Use the User Properties dialog box as you would the new user dialog box. Refer to the *Creating User Accounts* section for more information about the fields in this dialog box.

4. Click the **OK** button.

Special Pre-Defined User and Group Accounts

The National Instruments User Account Manager comes with several user accounts and groups built-in. The built-in user accounts include *Administrator*, *Guest*, and *(nobody)*. The built-in groups include *Administrators*, *Everyone*, *Guests*, *Operators*, and *System Operators*. You cannot delete any of these accounts, though you can edit the properties of some of them.

The Administrator account overrides all other security settings and has access to everything in Lookout. This override extends to all individual accounts added to the Administrators group.

You cannot delete the Administrator account or change its security level. You can set the password and enter the name and a description of the Administrator. You can add or remove individual user accounts from the Administrator group.

The (nobody) account cannot be edited or deleted, and does not actually appear as an account in the User Account Manager. This account is what Lookout defaults to when no authorized user is logged on. It always has a security level of 0.

The Everyone group cannot be edited or deleted, and does not actually appear as an account in the User Account Manager. By default, the Everyone group has full read and write permissions to your entire process.

You can edit all the properties of the Guest user account and of the Guests, Operators, and System Operators groups.

User Account Manager Timeout

You can set the time in minutes that the User Account Manager utility can remain completely idle (including no mouse movement over the active Lookout window until a dialog box appears, requiring you to log on as an Administrator again. A value of 0 allows the User Account Manager to run indefinitely. Select **Options»User Manager Timeout** within the User Account Manager to change this setting.

Ensuring Overall System Security

Lookout provides security for processes, but Lookout security features are not a substitute for proper system security for your computer network. For example, if your computers are connected to the internet, a firewall is a good idea. Lookout does not provide this type of security for your computers.

When security is important, it is also a good idea to put your process files in a secure directory, which is possible if you use Windows 2000/NT/XP instead of Windows Me/98.

Security Setting Interactions

If you find yourself with complex security setting interactions, the following principles should help you sort out how your interactions will work.

- In cases when you have both control security and network permissions set for an object, it is best to make sure that the control security is consistent with permissions.

- Group permissions are cumulative. For example, if user_A is a member of both Operators (read access) and System Operators (write access), user_A has both read and write access. The exception is that the **no access** setting in the Network Security dialog box overrides all other permissions.

- User and group permissions are cumulative. For example, if user_A's user account has a security access level of 5, and user_A is also a member of the Operators group, which has a security access level of 7, user_A effectively has an access level of 7. The exception is that the **no access** setting in the Network Security dialog box overrides all other permissions.

Logging On and Off

The first time you start a development version of Lookout, it opens with the Administrator user account logged in, without requiring any login. You should assign a password to the Administrator account before allowing others who should not have Administrator privileges to use your copy of Lookout. Once a password has been assigned to the Administrator account, a login is required anytime Lookout is launched.

Server and client run-time versions of Lookout open with the (nobody) user account logged in, no matter what the password is for the Administrator account.

Select **File»Log Off** or press <Ctrl-D> to log off. To log back on, select **File»Log on**, press <Ctrl-L>, or click the account name in the status bar.

When no one is logged into Lookout, the (nobody) account is automatically logged in.

Note   If the (nobody) account is logged on, any functions of a process that require a security level greater than zero do not receive or report data until someone logs on using an account with a high enough security level.

Note   If yours is the only account that is a member of the Administrators group, and you forget your password, there is no way to access the **System»User Manger** command, and there is no way to modify account settings. Contact National Instruments for assistance.

In run mode the name of the user account logged in to Lookout is shown in the status bar at the bottom of the screen. You can programmatically access the name and security level of the user account currently logged in to Lookout using the $System object's `username` and `seclevel` data members.

Control Security

Control security for a Lookout user interface object, such as a Pot or a Switch, is stored in the process file and implements security at the operator level. This security level is compared to the security level assigned to a user account or a group to prevent or enable access. If you want to keep certain operators from doing certain tasks, configure the control security accordingly.

Several objects in Lookout support control security, including Pots, Switches, Pushbuttons, RadioButtons, and TextEntry objects, and a few driver objects. These objects have a **control security level** parameter, which is compared to the user's security level anytime the user attempts to interact with it. If the user is allowed access, the mouse cursor changes into a hand when positioned over the object and the operator can adjust and control the object. If the user is not allowed access, the cursor changes into the international symbol for forbidden, and the operator cannot control the object.

Control Is Accessible

Control Access Is Denied
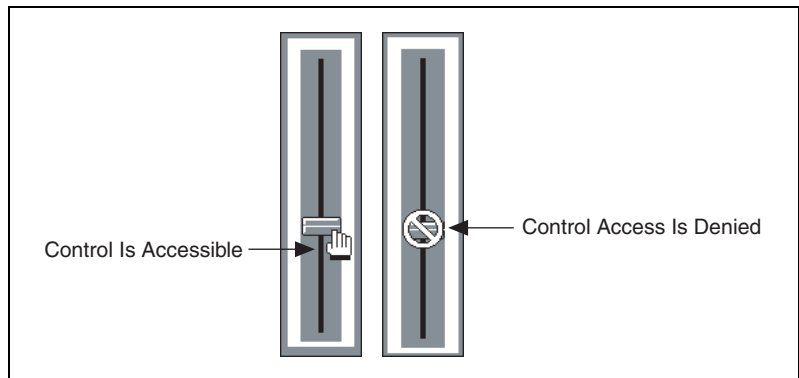
You can implement control security on an object-by-object basis, through the individual security level set in the object properties dialog box.

Lookout globally applies the Control Panel object security setting to all individual objects on that panel. If an individual item on that panel has a higher security level than the Control Panel, then that higher level is enforced.

Controlling the Visibility of Objects

Lookout provides visibility control for panels, controllable objects, and system settings. With these security options, you can restrict access to control panels, objects, and Windows system resources.

Control Panel Visibility

Control Panel objects have a **Viewing** security parameter that sets the visibility control for the entire panel. For example, if you set **Viewing** security to level 6 on a particular panel, operators with level 5 or lower cannot view that control panel and might not even know that panel exists. If a level 6 (or higher) operator logs on, the control panel instantly becomes

available for display. This feature is useful for hiding panels that are rarely used or that contain sensitive information.

Controllable Objects Visibility

Controllable objects such as Pots, Switches, Pushbuttons, and so on have a writable data member called `visible`. When `visible` is true, you can see the object on a control panel. When `visible` is false, you cannot see or adjust the object. To ensure that the object is always visible when it is first created, `visible` defaults to true.

You can connect the `visible` data member of a controllable object (for example, a Pot object) to a logical input signal. When the incoming logical signal is true, the `visible` data member of the Pot would be true, allowing the operator to see the Pot and adjust it. But when the logical signal is false, the `visible` data member might be false, hiding the Pot from the operator and prohibiting operator control.

You can also use the `username` or `seclevel` data members of the $System object to control the visibility of a control object, depending on the name or security of the person logged on to Lookout at any given time.

System Security Settings

You can define system options to keep Lookout maximized, the menu bar invisible, title bars invisible, and pop-ups to a minimum.

1. Select **Options»System**. The following dialog box appears.
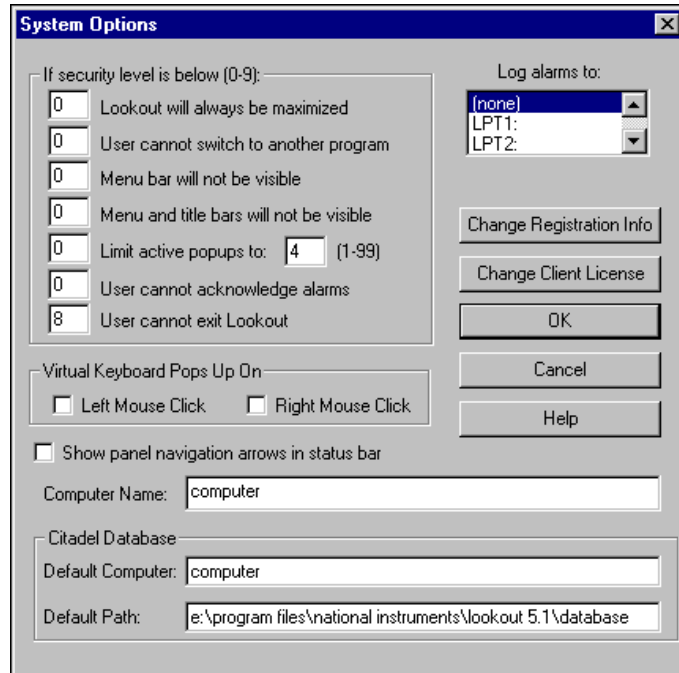
2.   Set the options appropriately.

- **Lookout will always be maximized**—When you enter a security level, Lookout prohibits users below that security level from minimizing Lookout.

- **User cannot switch to another program**—This prevents an operator from using <Alt-Tab> to switch from Lookout to some other program running on the computer. For this feature to work properly under Windows NT, you must install the Lookout NT keyboard driver when you install Lookout.

- **Menu bar (and title bars) will not be visible**—When you enter a security level, users below that security level cannot view the menu bar or the title bar. On computers running Windows 2000/NT/XP, users may still be able to press <Alt-Tab>, <Ctrl-Esc>, or <Ctrl-Alt-Delete> to switch tasks or access the operating system. You can disable this ability under Windows NT by installing the Lookout NT keyboard driver.

- **Limit active popups to**—This option requires two values—a security level and the number of pop-ups. Users below that security level can view up to the specified number of pop-ups at one time. This feature can help keep users from becoming lost.

- **User cannot acknowledge alarms**—This option prevents an unauthorized operator from acknowledging alarms.
- **User cannot exit Lookout**—This option prevents an unauthorized operator from exiting Lookout.

Network Security

Lookout processes can pass data and commands back and forth across a network. Network security prevents or enables this communication based on who is logged in on each instance of Lookout running on the networked computers. Lookout network security works between different instances of Lookout running on one computer or different computers on the network. Network security is based on user and group permissions configured for processes, collections of objects grouped in a folder, or individual objects.

With network security, Lookout checks the user account permissions configured for an object or process to determine if an operator can control (write to) a particular object. If a user does not have proper access, the user can adjust a control, but the process does not accept the input and the control returns to its original value.

Note For user accounts to work consistently across your network, you must use the same `lookout.sec` file for all installed copies of Lookout. Refer to the *Lookout Security Files* section later in this chapter for more information.

Configuring Security Permissions for Processes, Folders, and Objects

With permissions, you can set access privileges for a process, a folder holding a collection of objects, or an individual object.

Lookout objects inherit the permission status of the process or folder in which they are created. When you first create a process in Lookout, it has full read and write permission granted to the Everyone group, by default. Any folder or object you create in the process has the same permission.

If you change the permission of the process or of one of the folders, any objects you create after the change have the permission status of the parent process or folder. Changing the permissions of a process or folder does not always change the permissions of an object or folder that already exists in that process, depending on how you set the Permissions dialog box options.

If a process has one set of permissions, and a folder under that process has a different set, the objects created under the folder will inherit the permission status of the folder only.

Complete the following steps to configure security permissions for a process, folder, or object:

1. In Object Explorer, Connection Browser, or **Insert Expression** dialog box, right-click the process, folder, or object you want to configure security permissions for and select **Configure Network Security**. The Security Properties dialog box appears.

✎ **Note**   You cannot configure security for a network node, for your local computer, or for any Lookout global objects such as $Keyboard or $System.

2. Click **Permission**. The Permissions security properties dialog box appears, as shown in the following illustration.
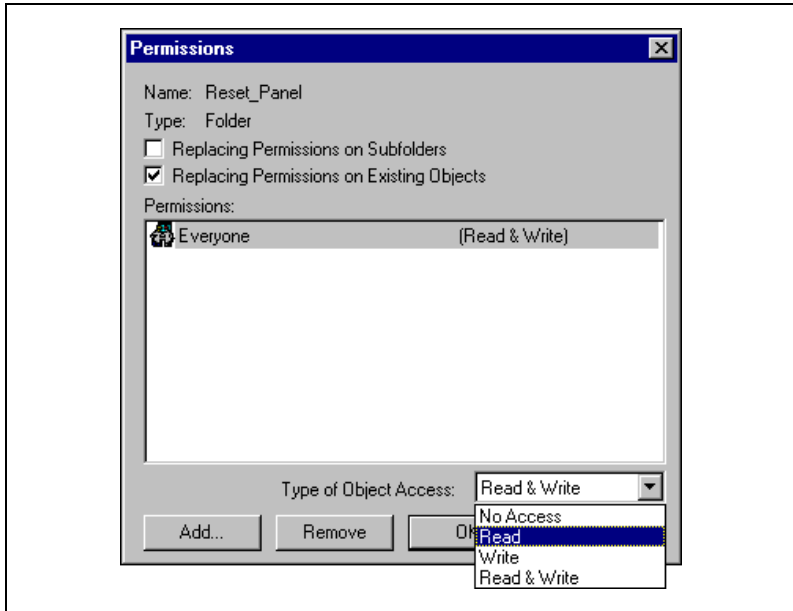
3. Configure the permissions appropriately. You can refuse access, permit reading or writing only, or allow both reading and writing. Because permissions are cumulative, you must remove the **Everyone** group to assign more specific permissions.

   The dialog box in the previous illustration shows that everyone has access to Lookout, with permission both to read and write for all the controls in the `Reset_Panel` folder.

   • **Replacing Permissions on Subfolders** and **Replacing Permissions on Existing Objects**—You can restrict permission to the process, folder, or object you selected, or extend the permissions as shown in the following table.

Table 10-1. Options for Replacing Permissions

Options Selected	Result
neither	Only the selected process, folder, or object has its security configuration changed.
Replacing permissions on subfolders	Changes permissions on the selected process or folder, all the folders under it, and any subfolders under them. This option is disabled when an individual object is selected.

Table 10-1. Options for Replacing Permissions (Continued)

Options Selected	Result
Replacing permissions on existing objects	Changes permissions on all the individual objects contained immediately under the selected process or folder, but not on any folder or subfolder under the selected process or folder. Disabled when an individual object is selected.
both	Changes permissions on all the individual objects contained immediately under the selected process or folder, as well as on any folder or subfolder, and all the objects in them.

- **Permissions**—Make sure that the users and groups for which you want to assign permissions are listed in this field. To add a user or group, click **Add**, and in the dialog box that appears select the user or group, click **Add** again, and **OK**. To remove a user or group, select it and click **Remove**. For your individual user and group permissions to have any effect, you must remove the **Everyone** group from this field.

- **Type of Object Access**—Select the appropriate access rights.

4. Click **OK**.

Advanced Security for Processes

You can set a number of advanced network security options in Lookout, but only at the process level. Complete the following steps to configure security permissions for a process:

1. In the Object Explorer, Connection Browser, or the **Insert Expression** dialog box, right-click the process, and select **Configure Network Security**. The Security Properties dialog box appears.

2. Click the **Advanced** button. The Advanced security properties dialog box appears, as shown in the following illustration.

3. Select the appropriate advanced security settings for your process. The default Lookout setting is to have both **Proxy user** and **Basic authentication** enabled and the other option turned off.

- **IP setting**—Check this checkbox to configure Lookout to grant or deny access to any computer operating at specific IP addresses. Click the **IP setting** button. In the **IP Setting** dialog box that appears, configure your selections using the **Grant access to** and **Deny access to** radio buttons, along with the **Add** and **Remove** buttons.

   If you choose to grant access to one or more computers using the IP setting option, those will be the only computers able to access the process you have applied the restriction to. If you choose to deny access to one or more computers using the IP setting option, all other computers using Lookout will be allowed to access the process you have applied the restriction to, subject to the other security settings in place.

   You can enter either the IP number itself, or the simple name of the computer. Whether you use the IP number or the computer name, the IP address for the computer appears in the list after you accept the entry.

- **Proxy user**—By defining a proxy user, you can designate a specific local security account whose security level applies for any user accessing processes in your local instance of Lookout from another computer (or another instance of Lookout running on the local computer). If this option is enabled, the security level of the operator logged into the external instance of Lookout is ignored.

To enable the proxy option, select the **Proxy user** checkbox, enter the **Username** of the account you want to serve as the proxy security account. You can specify any defined account as the proxy user account, but the Guest account is specifically provided for this purpose (as well as for providing a visitor with a user account). You must also enter the valid **Password** for the proxy user account.

- **Basic authentication**—When you select this option, Lookout checks the account information of a user logging in to that instance of Lookout. Security responds to the security level, individual account, and group permissions of that user account.

  To process the security status of a person logged on to another computer, you must have an identical `lookout.sec` file on each computer. Otherwise, if you enable **Basic authentication** but disable proxy access for your process, a person attempting to read from or write to your process from a remote computer cannot access it unless you have configured that permission for the Everyone group.

As with other Lookout security settings, the effects of multiple selections in this dialog box are cumulative. For example, if a user had permission to read under **Basic authentication** and to write under **Proxy user**, the user would be able both to read and write.

Protecting Process Files from Editing

You can protect your process files from being edited by unauthorized users, without using the security accounts, as follows.

⚠ **Caution** You cannot open an encrypted file with an earlier version of Lookout, even if you create an account with the same account name and password in that version.

⚠ **Caution** When you protect a process file, Lookout does not save the `.lks` file. Because the `.lks` file serves as a backup file during application development, you should not use this encrypted-save feature until after you have completed your application and made a backup copy of all of your development files on a separate archive disk.

1. Log in with an account name and a non-empty password that should have authorization to edit the process file.

2. Select **File»Save As** from the menu. The following dialog box appears.
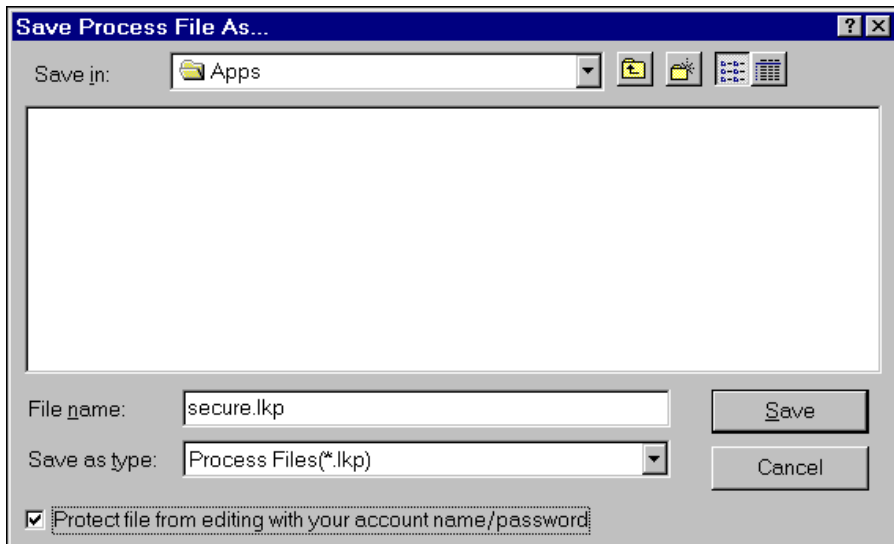
3. Check the **Protect file from editing with your account name/password** box at the bottom of the dialog box to save the file with your password as protection. To edit the file again, you must log in under the same account with the same password.

4. Click **Save**.

Implementing Action Verification

The Switch and Pushbutton objects support action verification. When you define action verification for an object and a user attempts to operate the control, Lookout displays a message box stating your **Verify** message and prompts the user to select either **Yes** or **No**.

To define action verification for a Switch or Pushbutton, enter values for the verification parameters when you create or modify the object. These action verification parameters accept text expressions, which can contain dynamic data. Refer to the object topic in the *Lookout Help* for more information about the object's parameters. Refer to Chapter 7, *Expressions*, for more information about creating expressions.

As an example, consider a switch that controls a pump that fills a storage tank unless the water level is already too high. You might enter an expression similar to the following for the switch **Verify On** parameter:

```
"Are you sure you want to turn on sludge return pump #2?
Holding tank #2 is currently " & DATA_VARIABLE & "
percent full."
```

The warning message appears every time you turn on the switch. Notice the water level is dynamic—it changes to reflect the value of *DATA_VARIABLE* when the switch is flipped.
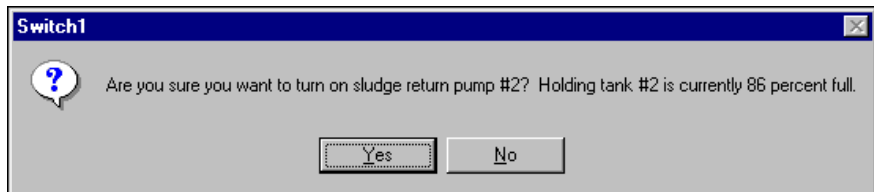
When you turn off the switch, no warning message appears because the verify **Off** parameter was not specified. If you want to disable the warning message, delete the entire expression from the **On** parameter field.

Note Pushbutton verification works in much the same way. However, when you select **Yes**, the pushbutton creates only a momentary output signal. When action verification is enabled, it is impossible to hold the button down for any length of time.

Importing Old Security Files into Lookout

You can import the user account information from your Lookout 3.*xx* processes into Lookout 4 or greater using the User Account Manager.

Note If you edit or import a lookout.sec or .lka file from an earlier version of Lookout, it will work properly. However, you will not afterwards be able to use the file with earlier versions of Lookout.

1. In edit mode, select **Options»User Manager** to launch the User Account Manager.

2. Select **User»Import Lookout 3.x Security File**. The following dialog box appears.
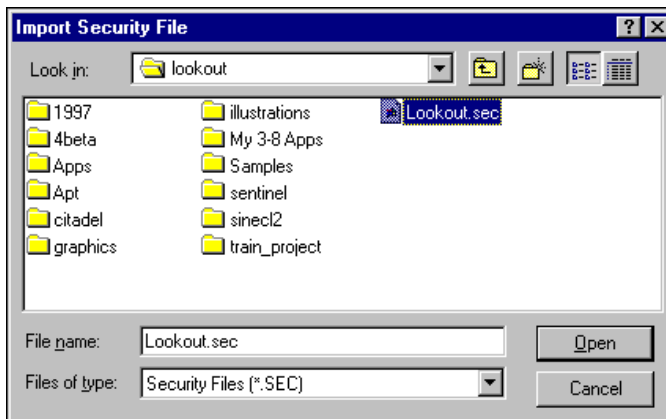
3. Navigate to your old `lookout.sec` security file and select it. Lookout 3.*xx* kept the `lookout.sec` security file in the `Lookout` directory.

4. Click **Open**. If you have already created any user accounts in Lookout 4 or greater that are the same as accounts you used in Lookout 3.8, you will receive a message informing you that a user account with that name already exists. You may replace your recently created account, or choose not to use the old account information.

5. Exit the User Account Manager.

Note   When converting a Lookout process from Lookout 3.*xx* or earlier, leave the control security in place when possible.

Lookout Security Files

Several Lookout files are important for the implementation of security.

*   The *processname*.lka file is used to store the security information relevant to making network connections to an objects data members. This is the security that you are setting when you configure permissions. You must keep the .lka file in the same directory as your process file. If you misplace it, all users will have complete access to all parts of the process.

*   The security level for control objects stored in the *processname*.l4p file.

- The `lookout.sec` file contains user and group account information relevant to both local/user interface and network security. If you want basic authentication to work between different computers running Lookout on your network, you must have an identical `lookout.sec` file installed on each computer. Copy your `lookout.sec` file to the Windows `System` directory of every computer on which you intend to run Lookout.

- Make sure unauthorized users cannot access the User Account Manager and copy over the `lookout.sec` file with their own version of the file. Do this by making sure Lookout is the only application in the Windows startup folder, and selecting the **User cannot switch to another program** in the Lookout System Options. You must install the NT keyboard driver to enable this option in Windows NT. This prevents an operator from toggling out of Lookout and modifying `lookout.sec` or deleting *processname*`.lka`.

- Passwords for user accounts are encrypted, so even if someone attempting to breach security acquires the `lookout.sec` file, they will have difficulty decrypting your passwords. Without the password, an unauthorized user cannot log on under someone else's account.

11

DDE

This chapter describes how you can use DDE with Lookout processes.

Dynamic data exchange (DDE) is the Microsoft message-based protocol used by applications like Microsoft Excel and Lookout to link to data in other applications. When the data in a source application changes, it dynamically updates all linked data (in real-time).

With DDE, you can dynamically link other Windows applications to Lookout. If Lookout is getting data from another application, Lookout is the client. But if another application is getting data from Lookout, Lookout is the server. Lookout can be both server and client in a peer-to-peer configuration.

The client application is responsible for establishing a DDE link with the server. When Lookout is a client, it first tries to establish an XlTable DDE connection (because this is the most efficient). If the server application does not support this format, Lookout uses the CF-TEXT DDE format.

To establish a DDE link, the client application must identify the location of the desired data. A three-tier address identifies the location of the data: Service, Topic, and Item. Look in the application documentation to determine its service, topic, and item.

Service specifies the name of the server application the client is linking to. Each application that supports DDE has a unique service name. For example, LOOKOUT is the service name of Lookout, and EXCEL is the service name of Microsoft Excel.

Topic is the second level in the three-tier address. For many server applications like Excel and Lookout, topic specifies a particular file. In Lookout, the topic is the process file name.

Item identifies the specific data or value being linked between the server and the client. A Lookout item is the object name, followed by a specific data member (such as Name.datamember) if needed. An item in a spreadsheet, such as cell B3 in Microsoft Excel, would be r3c2.

Linking Lookout to Other Applications with DDE

Lookout can act as a DDE client, DDE server, and both DDE client and server. Therefore, there are three basic ways to link Lookout to another application using DDE:

- Lookout as the server
- Lookout as the client
- Lookout as both client and server (peer-to-peer)

✎ **Note**   All readable numeric, logical, and text values in Lookout are automatically available to any other application through DDE. No special setup is required.

Because Microsoft Excel is widely used and accepted, it is used in the Lookout DDE examples.

DDE Server Example

In this example, you can send information from Lookout to another application, making Lookout the server. First, create a potentiometer in Lookout so you can link its value in real-time to a cell in Excel. Any time the Pot is adjusted, the value in the spreadsheet cell automatically changes.

1. Make sure Lookout is not in edit mode.

2. Hold down the <Ctrl> key and click the object you want to link to. In this case, select the Pot object you just created.

3. Lookout beeps when it successfully copies the object value to the clipboard. The object can be a slider, bar graph, switch, pushbutton, digital display, text entry object, knob or almost anything else in Lookout that contains a value.

4. Start Excel and select the cell you want to link to.

5. If your version of Excel is 5.0 or later, select **Edit»Paste Special**, then click **Paste Link**. If you have an earlier version of Excel, select **Edit»Paste Link**.

You have just created your first DDE link. Repeat this process as many times as you need. If you are linking large numbers of objects to Excel, you might want to use the Excel copy and edit tools to speed up the process.

Not all applications support the Windows clipboard shortcut method as described above. Therefore, you might have to manually enter the appropriate Lookout service, topic, and item in the other application to create a DDE link to that package. The format in which you enter this

information varies from one package to another. For this reason, you should refer to the documentation of the client application for instructions.

DDE Client Example

In this example, you import information from another application into Lookout. For instance, you might want to use a value calculated inside a spreadsheet as a process control setpoint for a Lookout application. In this kind of DDE link, Lookout is the client and the spreadsheet application is the server. Because Lookout is the client, it is responsible for establishing the link to the server data. Therefore, you must identify the service, topic and item in Lookout. These are object parameters in the DdeLink object.

1. Select **Object»Create** and select the DdeLink object.

2. In **Service**, enter the name of the software package (`Excel` in this example).

3. In **Topic**, enter the name of the spreadsheet file.

4. In **Item**, enter the address of the cell you want to read a value from.

    Notice that the entered cell address is `r1c1`. This translates to row1/column1 (cell A1) in Excel. The Excel DDE structure requires this format.

5. Click **OK**, and then select **OK** again when Lookout prompts you to insert the expression DDE1. Finally, pick the desired display format and click **OK**.

To test your link, enter a numeric value into cell A1 of your spreadsheet. Whatever value you enter into the spreadsheet is immediately written to the DDE expression on your panel. You can also connect the DdeLink object you just created to other Lookout objects. Refer to the *DdeLink* topic in the *Lookout Help* for more information about connecting a DdeLink object.

Note   The DdeTable object is another way of linking data to Lookout using DDE. This object links large quantities of data though the more efficient XlTable format. Refer to the *DdeTable* topic in the *Lookout Help* for more information about linking data to Lookout using DDE.

DDE Peer-to-Peer Example

Assume you want to take the Lookout as a DDE server example one step farther. Suppose you want to adjust the Pot to change a value in Excel and also be able to enter a different value in Excel to adjust the Pot. That is, you want to send data both ways through a DDE link. You can easily create such two-way links for user-controlled objects (that is, Switches, Pots, and Pushbuttons).

1. Select **Object»Create** and define a new Pot; or select **Object»Modify** and select the existing Pot object.
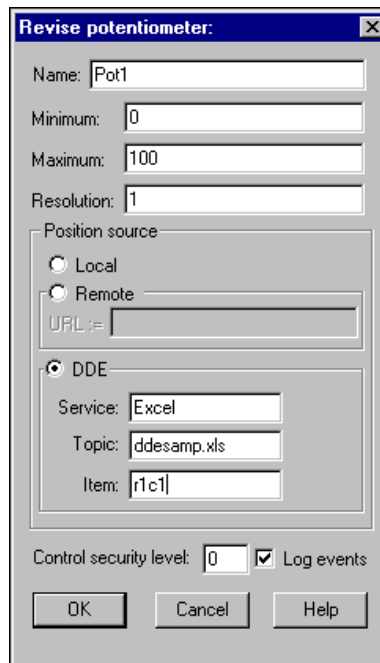
2. Change **Position source** from **Local** to **DDE**.
3. In **Service**, enter the name of the software package (in this case, Excel).
4. In **Topic**, enter the name of the data file.

5. In **Item**, enter the address of the cell you want to read a value from, such as r1c1 (for cell A1 in Excel).

6. Click **OK** to create or modify the definition of the object.

7. If the object is new, insert its display member into the panel so you can test your link.

To test your link, enter a value into the spreadsheet cell you specified and watch the Pot. Then adjust the Pot and watch the spreadsheet cell. You should see the values within the two applications change in unison.

✎ **Note** If you link to a Pot object, the linked value is numeric, so you enter a numeric value into the spreadsheet cell. But if you link to a Switch or Pushbutton object, the linked value is logical. Linked logical values are shown in spreadsheet cells as true or false. To change the value of a logical value in a spreadsheet cell, enter true or false, 0 or 1, or on or off.

DDE Alarms

The following section explains alarms that might appear in the Lookout Alarms Window.

Cannot establish DDE conversation with <service>, <topic>

This alarm occurs if a Lookout client is unable to connect to the server corresponding to the given service and topic. The alarm also occurs if the server terminates the conversation (for example, if the server is shut down). The alarm is deactivated when the Lookout client successfully connects to an item on the server.

Verify that the service and topic were typed correctly when you created the object that is using DDE. Verify that the server application is running. If the server is on another computer on the network, verify that the network is up. If the server is on a computer running Windows NT, verify that you are authorized to log on to that computer and that the current user logged onto the NT machine has trusted the DDE share to which you are trying to connect. Refer to the *Adding a Trusted DDE Share* section in Appendix E, *Networking with DDE*, for more information about using a DDE.

DDE client error for <service>, <topic>, <item>: (received NACK for advi4se)
DDE client error for <service>, <topic>, <item>: (received NACK for request)

Verify that the named item exists on the server and that the server supports DDE links for the item. This alarm occurs when Lookout is a client.

DDE client error for <service>, <topic>, <item>: (received NACK for poke)

This alarm occurs if you are using Lookout as both client and server, and have made a remote connection to an item that is not writable in a DataTable, Pushbutton, Pot, Switch, or TextEntry. The only Lookout objects that support writes (pokes) are DataTable, Pushbutton, Pot, Switch, and TextEntry. These support writes into their implicit data members only.

If the server is running Windows NT, it is possible that the DDE share on the computer is configured to support reads (advises) but not writes (pokes).

DDE client error for <service>, <topic>, <item>: (advise timed out)
DDE client error for <service>, <topic>, <item>: (request timed out)
DDE client error for <service>, <topic>, <item>: (poke timed out)

Verify that the server application is running. Verify that the item exists on the server. If the server is on another computer on the network, verify that the network is up.

DDE client error for <service>, <topic>, <item>: (received invalid data)
DDE server: corrupt data block poked to item <item>, topic <topic>

Either the server received a corrupt data block from the client, or the client received a corrupt data block from the server. This might be the result of network trouble. If the alarm is consistent and predictable, you might have discovered a bug. Call National Instruments technical support for further help.

DDE server: failed to post advise for item <item>, topic <topic>

Verify that the client application is running. Verify that the item still exists on the client. If the client is on another computer on the network, verify that the network is up.

12

Reports and Web Clients

This chapter explains different methods for generating reports and how to create a Web Client.

Generating Reports

If you want to generate reports from your Lookout process, you have several options. You can print control panels, generate HTML page reports, or create spreadsheet reports using other programs and utilities.

Printing Control Panels

You can manually print a control panel by clicking the icon in the upper left corner of the panel and choosing **Print panel**. You can print a control panel programmatically using the Panel object's `print` data member. When `print` changes from `FALSE` to `TRUE`, Lookout copies a picture of the control panel and sends that copy to your printer. You could use this to record the status of a particular panel at a particular time. Refer to the *Panel* topic in the *Lookout Help* for more information about using the Panel object.

✎ **Note** If you print a panel that contains the display of a windowed ActiveX control such as Internet Explorer, the display does not print. However, windowless ActiveX controls such as ComponentWorks knobs will print fine.

Lookout determines the printed panel size based on dimensions entered in the Revise control panel settings dialog box. There are several ways to access this dialog box. First, make sure you are in edit mode. Then, do one of the following:

- Right-click a panel title bar.
- Right-click a panel in the **Object Explorer** and select **Properties**.
- Select **Object»Modify**, then select a panel from the list.
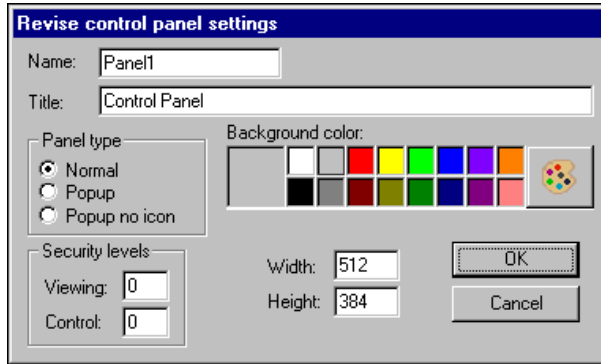
Popup and Popup no icon panel types print as they appear on the screen.

When **Panel type** is set to **Normal**, Lookout assumes that the panel is maximized and prints only the area defined by your Panel object **Width** and **Height** settings. For example, you might have a panel that looks like the following when maximized.
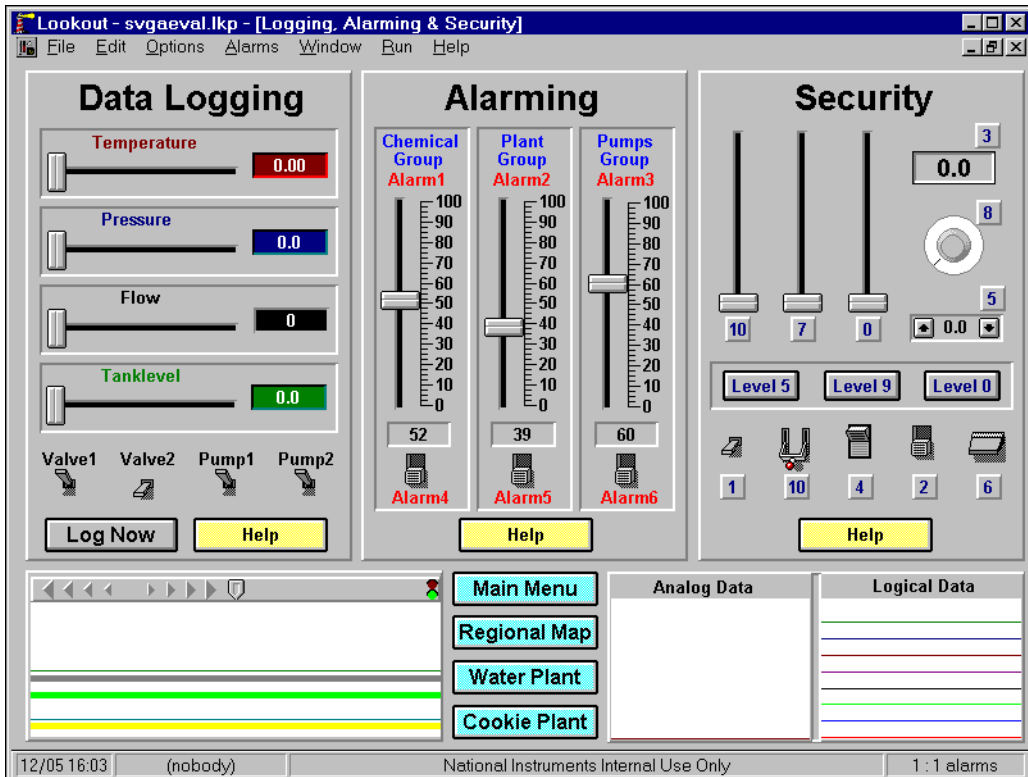
The panel in this figure has a specified **Height** of 300 pixels and a **Width** of 400 pixels. At a video resolution of 800×600, the printed panel would look something like the following figure.
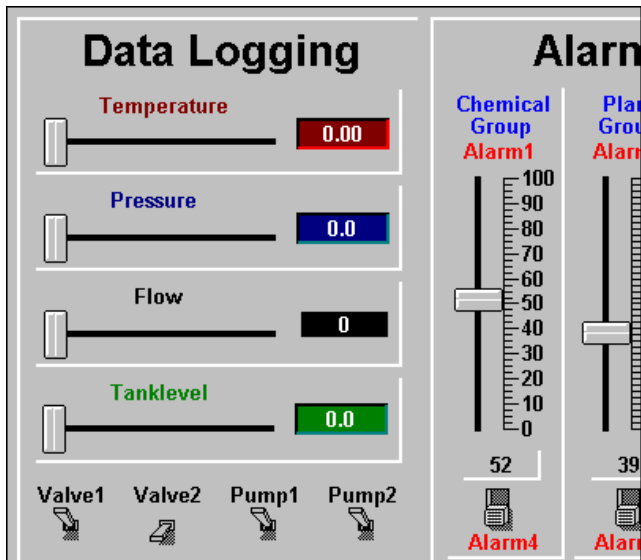
Modify the **Height** and **Width** dimensions of your Panel object and/or your screen resolution until they match. This problem applies only to **Normal** style panels.

Generating HTML Reports

You can use the Report object to generate an HTML page that you can put up on a Web server to allow people to access the reported information. The data on a report page is not live; the report page features a snapshot of the data as it existed when the page was generated. The advantage of this is that no downloading is necessary for someone to browse the HTML page this object creates, and no Lookout objects can be manipulated or changed by those browsing the page. The elements you can use on HTML report pages include data, graphics, displays, charts, control panels, text or messages, links to other HTML pages, and more. Refer to the *Report* object topic in the *Lookout Help* for more information.

You can also use the Historical Data Viewer and its HTML template to export historical data to HTML format. Select **Options»Historical Data Viewer** to launch the Historical Data Viewer, and refer to its help for more information.

Creating Third-Party Spreadsheet Reports

You can generate reports using Lookout and a third-party spreadsheet program. One way to do this would be to export data with the Spreadsheet object, run your spreadsheet program using the Lookout Run menu or Run object, and configure the spreadsheet program to run macros automatically when it opens to automatically generate a report.

Another way to create spreadsheet reports is to use Historical Data Viewer in MAX. You can select traces with start and end times, and then export that data to a text file that can be opened in a spreadsheet program. Select **Options»Historical Data Viewer** in Lookout to launch the Historical Data Viewer. Refer to its online help in MAX for more information about exporting data for reports.

Creating a Web Client Page in Lookout

Using a Web Client, you can create an HTML file that can be accessed through Internet Explorer and downloaded to a remote computer. Unlike a Report object, a Web Client allows you to display *and* control data as with any other client process. The data is live and interactive, and you can allow operators to manipulate the control objects.

To create your Lookout Web client, review the requirements, create the client process that you will export, configure the Web client export options, then export the page. These steps are described in the following sections.

Web Client Requirements and Limitations

Please note the following requirements:

- For your Web Client page to work, either the Lookout software or Lookout Player must be installed on the remote computer prior to the first download. The Lookout Player is available free of charge.

- For your Web Client to connect to a server process, you need a concurrent client license registered on the server machine. Otherwise the server will refuse the client connection.

- The Lookout Web Client is only supported on Internet Explorer.

- Standard client limitations apply when exporting to a Web Client.

- Only ActiveX controls that ship with Lookout can be reliably exported to a Web Client. However, not all ActiveX controls in the Lookout Recommended category are installed with Lookout. The NI HyperTrend and the NI AlarmEvent Browser cannot be exported to a Web Client.

Create a Client Process to Export

To export a Web client successfully, make sure you have created and connected the client properly. For more information about properly creating clients, refer to the *Lookout Client/Server Application Design* section of Chapter 2, *Developing an Application*. For more information about properly connecting clients to server processes, refer to the *Connecting Clients to a Server* section of Chapter 5, *Creating Lookout Processes*.

Web Server Export Options

Before you can export a Web client page, you must set the export options.

1. Make sure Lookout is in edit mode, and select **File»Web server export options**. The following dialog box appears.
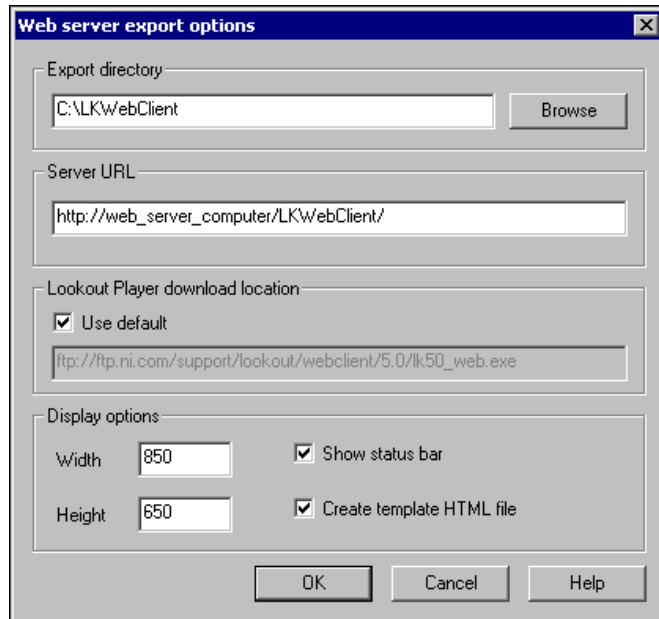
2. Enter an appropriate value for **Export directory**. When you create your HTML page, Lookout creates an entire support package necessary to run the process, except for a copy of the Lookout Web Client (Lookout Player). **Export directory** is the location to which all these necessary process files (including the HTML page) will be exported. The following are three different scenarios that you might have:

   - Your development machine is different from the Web server computer and you do not have network access from the development machine to that computer. In this case you can put an arbitrary folder on the development machine as the **Export directory,** as shown in the previous illustration. After you export the files to this directory, you can copy that directory and all its contents to the Web server root directory.

   - Your development machine is different from the Web server computer and you have network access from the development machine to that computer. In this case you can export the files to the Web server computer over the network. As shown in the following illustration, in the **Export directory** field, enter a path to the Web server root directory on the Web server computer.
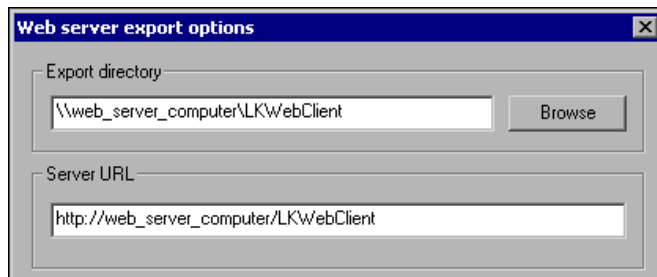
   - Your development machine and the Web server computer are the same. In this case you can directly export the files to the Web server directory by entering a path to that directory in **Export directory**.

   A *Web server root directory* is the directory that the Web server software exposes to clients. When you type `http://web_server_computer`, you are referencing the Web server root directory on the Web server computer. For example, assume the Web server root directory is `c:\Inetpub\wwwroot` and you enter `c:\Inetpub\wwwroot\LKWebClient` for the **Export directory**. The `client.htm` could then be accessed from the client machine with the following URL:
   `http://web_server_computer/LKWebClient/client.htm`.

The default Web server root directory for Microsoft Personal Web Server 4.0 or Microsoft IIS (Internet Information Services) 5.0 is `c:\Inetpub\wwwroot`. The default Web server root directory for Apache Web Server 1.3 is `C:\Program Files\Apache Group\ Apache\htdocs`.

You can export to a subdirectory of a Web server root, such as `LKWebClient` in the previous illustration.

3. Enter an appropriate value for **Server URL**, which is the path to your HTML page. It must include the complete Web server path to the page, not including the page itself. For example, if the path to your page is `http://web_server_computer/LKWebClient/client.htm` use `http://web_server_computer/LKWebClient/` as the **Server URL**. Do not use spaces in any of your path elements.

   If you do not have access to Web server software and only want to run the client on your LAN (Local Area Network), you can expose the client files through a shared folder. For example, if your HTML page is on a computer called `server_computer` in a folder called `LKWebClient`, then in **Server URL** you can type in `\\server_computer\LKWebClient\`. Make sure the folder `LKWebClient` is shared on the network. This creates an HTML file that does not require Web server software. You cannot access this file over the Internet, but you can access it on your LAN.

4. Select a **Lookout Player download location** from which to download the Lookout Player (Lookout Web Client software). If you already have Lookout software installed on the client computer, you do not need to install the Lookout Player. If you check **Use default**, the location will be the National Instruments FTP site. This URL will be included as a link at the top of the Web Client page, so that a user at the client machine can easily download the Player. The Lookout Player may also be included on your Lookout software CD.

   The National Instruments FTP site will always have the latest version of the Lookout Player. However, we suggest keeping a copy of the Lookout Player installation file on your network for faster download. In that case, you can provide a path to the file instead of checking **Use default**. For example, if you have the file `lk51_web.exe` in a folder named `LookoutPlayer` on a computer named `Resources`, you would enter `\\Resources\LookoutPlayer\lk51_web.exe` as the **Lookout Player download location**. Make sure that you have shared the `LookoutPlayer` folder.

5. Select your **Display options**, which set the size of your display in pixels. If you set the **Height** and **Width** to values smaller than your client, scroll bars will be available on the downloaded page. **Show**

status bar selects whether to include the Lookout status bar on your page. If you want your HTML page to look like a Lookout screen, select this option. If you want your client to appear as an ordinary Web page, disable this option, maximize the control panel of your client application in Lookout, and make sure the background color of your panel matches the background color of your HTML file.

Showing the status bar allows viewers access to all of the functions Lookout enables through the status bar, such as opening the Alarms Window by clicking the alarms section of the status bar, navigating between control panels using the control panel navigation arrows, logging in by clicking the user name section of the bar, and accessing the **About Lookout** information by clicking the organization name.

Note   If you want users to be able to log in to a Web Client with an account other than (nobody), make sure the `lookout.sec` file is copied to the Windows `System` directory on the Web Client machine.

Check the **Create template HTML file** option if you want your exported page to create an HTML file in addition to the Web Client itself. You can then add any other HTML features you want on the page yourself. If you do not change the file name or move your changes to another file, and you export again while this option is selected, Lookout overwrites any changes you might have made to the file yourself. In most cases, you should check this option the first time you export your Lookout Web client. Any time after that when you need to export a new version of the process, you should deselect this option to keep any changes or improvements to your HTML file from being overwritten.

6.   Click **OK** and proceed to the next section, *Exporting a Web Client*.

Exporting a Web Client

After you have configured your export options, export the Web Client page by selecting **File»Export Web server files**. If you have more than one process running, a dialog box prompts you to select which process to export.

When you export, Lookout creates two folders and an HTML page in the directory you set as your export directory. The `Lookout` folder contains all the support files necessary to run the client process with the following subfolders: `cbxs`, `graphics`, `processfiles`.

Lookout creates the HTML page with only the minimum coding necessary, but you can edit it in any HTML editor.

Accessing a Lookout Web Client

Any Web Client HTML page you create with Lookout can be accessed from any computer that has Internet Explorer installed. Complete the following steps to access a Lookout Web client.

1. Open Internet Explorer and make sure the security setting for the Internet or your Intranet (depending on which one you are using) is **Medium** or lower. (You can do this by going to **Tools»Internet Options** and selecting the **Security** tab.)

2. In Internet Explorer, enter the URL for your HTML page. For example, if you used the **Web server export options** shown in this chapter and your HTML page is called `client.htm`, then the URL that you would enter would be `http://web_server_computer/LKWebClient/client.htm`.

    If your Web browser security settings are not correct, you might get a warning that the page may not display correctly. Click **OK**. Verify your Internet Explorer security settings again.

    If you have the correct security settings, a dialog box appears asking if you want to install and run the Lookout Web Client. Click **Yes**.

3. If your client computer does not have Lookout software installed, you are prompted to download the Lookout Player. Click **OK** and then click **Click here to download** to get the Player. This is the link provided in the **Lookout Player download location** section of the **Web server export options**.

    After your download is finished, run the file to install Lookout Player on the client computer. You might be prompted to reboot. After installing Lookout Player, launch the HTML page again.

4. Lookout Web Control determines whether any process-related files need to be downloaded to the client computer. If so, a dialog box appears asking whether to download the files. The dialog box shows you how much storage space is needed for the download. Click **OK.** The files are downloaded to the folder where the Lookout software is installed. After the files have been downloaded, the Lookout Web Client appears, embedded in your Web browser.

Note If you are downloading to a Windows 2000/NT/XP computer, you must be logged in with administrator privileges.

Setting Up Your Own Web Server

Lookout is not itself a Web server program. If you want to have your Web page available on the Internet, you must put your page and the support files on a computer running Web server software such as the Apache Web server or Microsoft's Personal Web Server.

You can get Web server software as a free download from either Microsoft or the Apache Software Foundation, as well as from other sources. There are also commercial Web server software packages available. Consult your system administrator for help in selecting, installing, and configuring Web server software, if necessary.

13

Networking and Deploying an Application

This chapter describes how to network and deploy Lookout applications.

Deploying Lookout Application Files

The development cycle creates at least four unique files for every application: a process file (`.l4p`), a source code file (`.lks`), a state file (`.l4t`), and a security file (`.lka`).

If you develop a Lookout process on one computer and want to run it on another computer, you must copy the process (`.l4p`) and security (`.lka`) files to the new computer for your process to run properly. You can also copy the source and state files, but Lookout can recreate these files in the new location if necessary.

Depending on how you have developed your process, you may also need to copy one or more of the following files when you deploy your application: `lookout.sec`, aggregate process files, `lookout.ini` or *object*`.ini` files, custom graphic files, and custom documentation. Refer to the following sections for more information about these files.

Process File

A process file is a compiled file that contains the Lookout process you want to run. It is updated every time you select the **File»Save** command. Process files are binary files with a `.l4p` (Lookout Process) file name extension. You do not directly edit or modify process files with a text editor. Rather, Lookout compiles the `.l4p` file on an object-by-object basis as you create each object. This is the main file to deploy for your process.

Aggregate Definition File

An aggregate definition file that you have created by selecting **File»Save As Aggregate** has a `.l4p` extension. When you deploy an application that includes an Aggregate object, you must copy its relevant aggregate

definition file into the directory with your other process-related files. Refer to the *Aggregate* topic in the *Lookout Help* for more information about creating aggregate definition files and using Aggregate objects.

Source Code File

The source code file (*processname*.lks) is automatically updated when you select **File»Save**. Lookout compiles this file to create a .l4p file, so it includes object definitions, names, I/O configuration, communications, control logic, control panel layout and other object parameters.

Source code files are standard ASCII text files that you can print or view with any word processor or text editor. You can use this information for debugging and documentation purposes.

You should always keep your .lks source files backed up. To open a process in a version of Lookout newer than the version used to create it, you must use the .lks file. A .lks file is automatically compiled when you open it in Lookout. To compile a lookout source file manually, refer to the instructions in Appendix C, *CBL Compiler*.

State File

The state file (*processname*.l4t) contains the actual values of object data members. These values include setpoints and other important data held within and used by objects. You might think of the state file as permanent memory for Lookout setpoints and real-time trends. Lookout uses the state file to store the position of switches, potentiometers, and trend lines. When Lookout first loads an application, it reads the appropriate state file to determine what state the controls and trends should be in upon startup, unless a control has a remote source connection, in which case it takes its initial value from the remote source.

By default, the state file updates any time you save, close, or exit the Lookout process file. It can also update on a periodic basis as defined by the process properties or the Loader object. You need to deploy this file only if it contains critical state information that must be used every time your process is launched. Otherwise, it is automatically generated and updated by your Lookout process. Refer to the *Opening a Process in a Known State* section later in this chapter for more information about preventing the state file from being updated.

Security Files

Every Lookout process has an attached security file (*processname*.lka). Network security information for a Lookout process is kept in the .lka file for that process. You must keep the .lka file in the same directory as the .l4p file for your network security settings to work. If you misplace the .lka file, all users will be able to read/write over Logos to any data member in the process. To reconstruct the file, you would have to reset all your security permissions.

The lookout.sec file contains user and group account information relevant to both local/user interface and network security. If you want basic authentication to work between different computers running Lookout on your network, you must have an identical lookout.sec file installed on each computer. Copy your lookout.sec file to the Windows System directory of every computer on which you intend to run Lookout.

Lookout INI File

The lookout.ini file is used to hold various Lookout system settings. The settings in this file are best made through the Lookout dialog boxes. Changing these settings manually is an advanced activity we recommend only experienced users attempt, and you should not copy this file when you deploy your application. For more information on the lookout.ini file, refer to Appendix D, *Lookout INI File Settings*.

Object INI Files

Occasionally Lookout must create a .ini file for a driver object, especially if you alter certain advanced configuration settings. Check the Lookout directory on your development computer for any .ini files that may need to be included when you deploy your process files.

Graphics and Documentation

If you have used any Image Navigator graphics or imported your own custom graphics, those files must be copied into the Lookout\Graphics directory when you deploy your process. If you have created any custom online help, remember to copy that to the appropriate location, too.

ActiveX Files

If you use ActiveX controls in the process you are deploying, you must install and register these controls on the computer(s) that will be running your application.

Registering Network Computers

If your Lookout processes will be networked across several computers, the networked computers must be registered with each other so that they can recognize/find each other. You can do this manually from each computer, or you can use the List server and autodetection features to speed up the process, as explained in the following sections.

✎ **Note** If you are logged on to a Windows 2000/NT/XP computer as Guest or Restricted User, certain development tasks will not work correctly, because they require writing to the registry or WINNT folder. These tasks include creating a user account, adding a client license, and registering a Logos computer.

Manually Registering Computers

Follow these steps to manually register the computers using Lookout on your network.

1. Open a process.

2. Select **Object»Connection Browser** or **Object»Object Explorer**.

3. Right-click Network, and select **Register Network Computer**. The following dialog box appears.
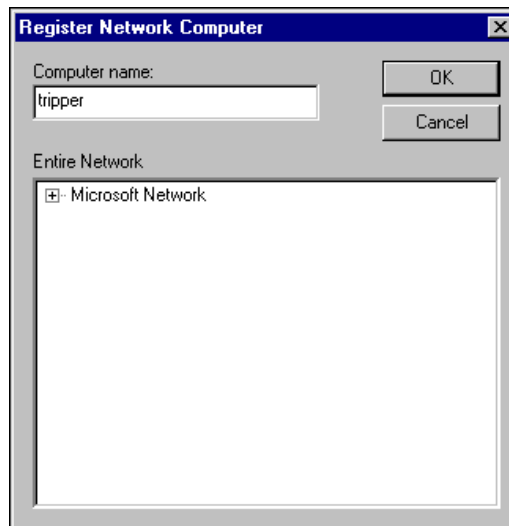
Type the name of the computer in the **Computer name** field. If you are unsure of the spelling, you can browse the network for computers by clicking the network node in the **Entire Network** field.

To remove a computer from your list of manually registered computers, select **Object»Connection Browser** or **Object»Object Explorer** and double-click `Network`. Highlight the name of the computer you want to remove, and select **Unregister Network Computer**.

4. Click **OK**.

List Servers and Auto-Detection

Lookout has both auto-detection and a list server enhancement for networking. These features help you access networked computers that are running Lookout or other National Instruments products that use the proprietary National Instruments Logos networking (including the LabVIEW Datalogging and Supervisory Control Module and certain FieldPoint modules).

Using the list server feature, you can register one or more computers on another computer that then acts as a list server. You can also set your computer to auto-detect other computers on your subnet that are using Logos networking.

Note   Software or hardware using early versions of Logos may not be accessible this way unless you manually register them. Additionally, not all National Instruments products using Logos networking permit you to enable or disable these options. Refer to the documentation for the specific product to see whether or not these networking features are configurable for that product.

Registering with a List Server

You create a list server by registering your computer on it.

1. Select **Options»Network**. The following dialog box appears.
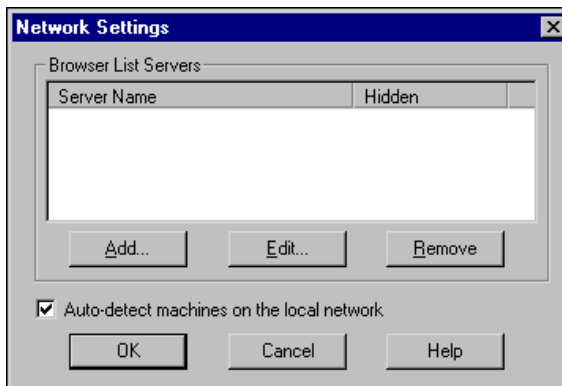
2. To register on the list server, click the **Add** button. The following dialog box appears.
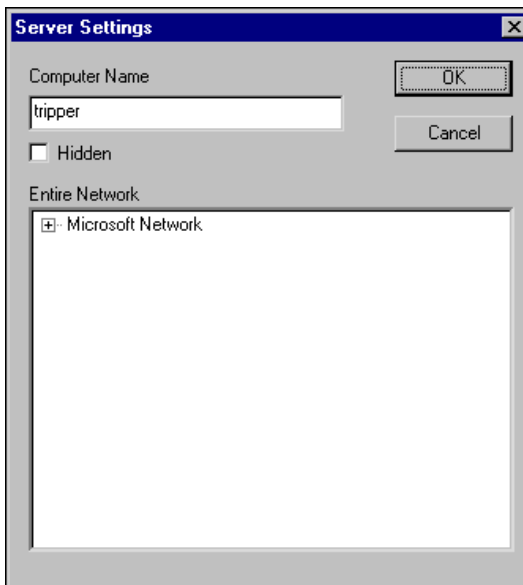
3. Enter the name of the computer you want to use as a list server in the **Computer Name** field, or browse for it in the **Entire Network** field.

4. Repeat this process for every computer you want to make visible through the selected list server computer.

Hiding Your Computer from List Server Clients

You can hide your computer from other computers using the same list server.

1. Select **Options»Network**.

2. Select a server and click the **Edit** button.

3. Select the **Hidden** Option in the **Server Settings** dialog box.

Selecting this option means that your computer can see all other computers using the list server you have registered, but those client computers cannot see you using the list server.

List Server Setting Interactions

Whether computers using National Instruments Logos networking are visible to each other via the list server depends on factors such as the following:

• interaction of the various list server settings

• whether the computers are all on one small local network or on a wide area network incorporating various subnets

Figure 13-1 shows the basic interaction of computers on a network, with the auto-detection option disabled.
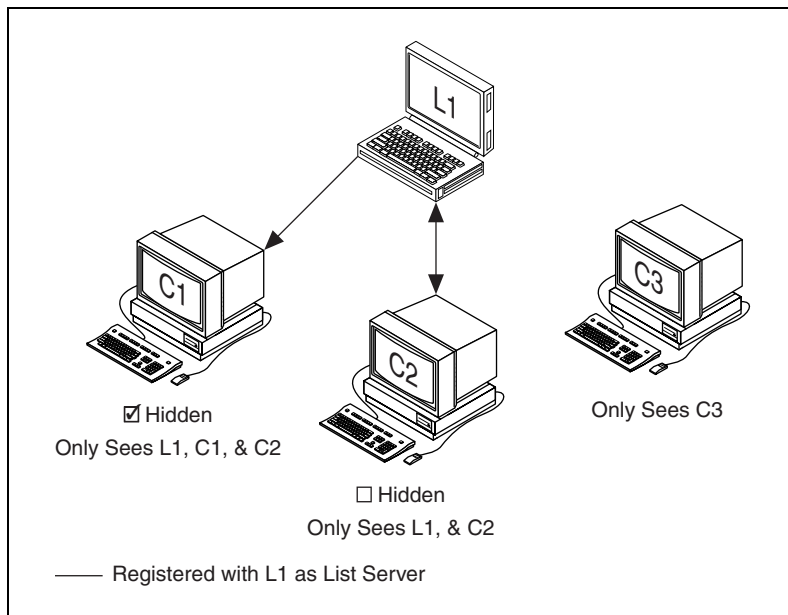
Figure 13-1. Effects of Registering with a List Server

Two computers, C1 and C2, have registered computer L1 as a list server. A third computer, C3, has not registered L1 as a list server. Computer C1 registered L1 with the **Hidden** option selected, but computer C2 did not.

With this configuration, the C1 computer can see the L1 list server, itself, and computer C2. The C2 computer can see the L1 list server and itself. The C3 computer can see only itself.

This example ignores manual registration of specific computers. Any computer can always manually register, and see, any other computer on the network. And manual registration does not automatically expose a computer to list server clients.

Auto-Detecting Computers on Your Subnet

You can set your computer to automatically detect any computer running the Classifieds service (`lkads.exe`) on your computer's subnet.

1. Select **Options»Network**.

2. Enable the **Auto-detect machines on the local network** option. With this option checked, you will be able to detect other computers on your subnet running the National Instruments Logos networking services and connect to data points they may be generating or accessing.

Auto-detection does not work across a subnetwork; you cannot auto-detect, or be detected by, computers running on different subnets, as shown in the following figure.
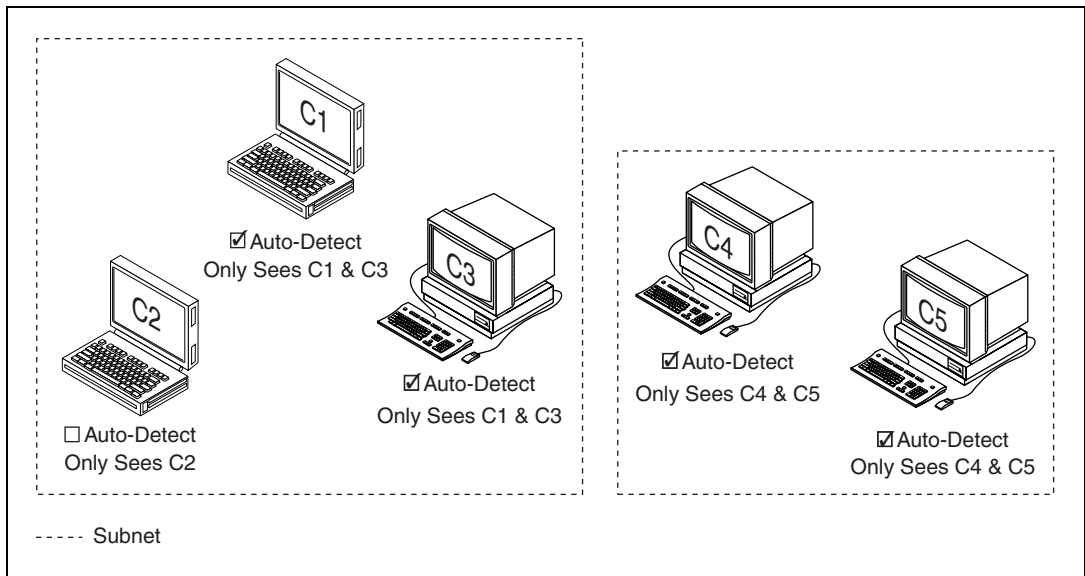
Figure 13-2. Effects of Auto Detection

In Lookout, computers with auto-detection enabled can see each other via the list server. Disabling auto-detection prevents your computer from being seen by other list server clients, even if they have auto-detection enabled. Those other computers could gain access to your computer by manually registering your computer.

⚠ **Caution**   Do *not* use the list server as a security feature. Any computer using National Instruments Logos networking can browse to another computer on the network, register it manually, and connect to data. You must implement security with user permissions.

Interaction of Auto-Detection and List Server Registration

When you have computers on different subnets using both a list server and the auto-detection feature, the interactions can become somewhat complicated. Basically, the following rules govern these interactions:

• A computer that has registered itself with a list server can see all computers the list server can see, except for those with auto-detection disabled.

- Manual registration does not automatically make a computer visible to a list server or to list server clients.

- A computer registered with a list server that has the hidden option enabled cannot be seen by the list server or list server clients.

- Computers with auto-detection enabled can see each other via the list server, when they are on the same subnet. Computers with auto-detection disabled cannot be seen via the list server.

In Figure 13-3, the registration and hidden options are similar to the list server illustration, but subnet boundaries are indicated, the auto-detection feature is enabled on some of the computers, and several extra computers are indicated.
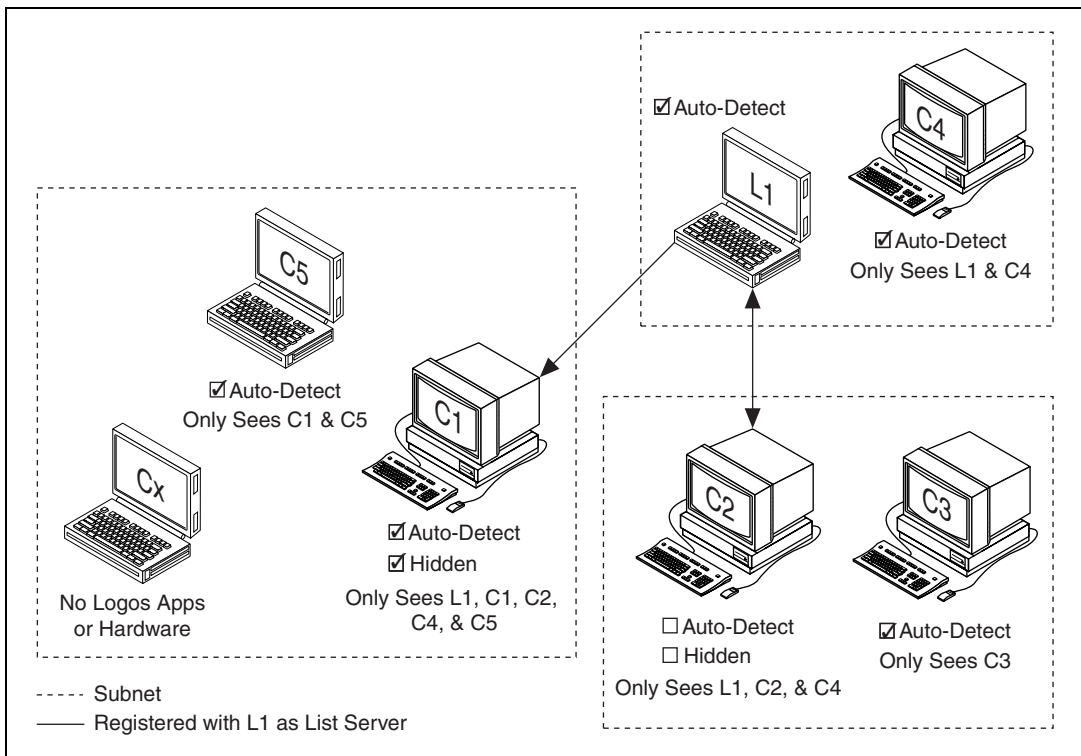
Figure 13-3. Interaction of Auto-Detect and List Server Registration

The list server L1 is on a subnet with another computer called C4, which is running Lookout, has not registered L1 as a list server, but does have auto-detection enabled. That computer will see the L1 computer and itself, but no other computers.

The C1 computer is located in a subnet with other computers (one running National Instruments products and others not), has the hidden option enabled, and has auto-detection enabled as well. Its companion computer, C5, has auto-detection enabled but is not registered with the list server. Computer C1 can see computer L1, computer C4, computer C2, computer C5 and itself. It cannot see computer Cx, which is not running any of the Logos services, nor can it see computer C3, which in on a different subnet from the L1 list server computer and is not visible to the L1 list server computer.

Computer C5 can see itself and computer C1 only.

As before, the C2 computer has L1 registered as a list server, with the hidden option disabled, and with auto-detection disabled. This computer can see computer L1, computer C4, and itself.

The C3 computer is on the same subnet as the C2 computer. It has not registered computer L1 as a list server, but does have the auto-detection option enabled. Computer C3 can see itself but none of the other computers on the rest of the network.

Computer C4, with auto-detection enabled, can see itself and computer L1. Not being registered with the list server, it cannot see other computers on the other subnets.

The list server computer (which can be any computer running Lookout) can see itself, computer C2, and computer C4. It cannot see computer C1, because C1 has the hidden option enabled, and so computer C5 is also hidden from L1.

This example ignores manual registration of specific computers. Any computer can always manually register, and see, any other computer on the network. Manual registration does not automatically expose a computer to list server clients.

Synchronizing Lookout Computers

To keep your data properly time stamped, you must make sure the times on your computers are properly synchronized. The Lookout *time synchronization* service is installed as a service in Windows 2000/NT/XP that runs every time you run your computer. Time synchronization runs as a background process in Windows Me/98.

If your primary server is off-line for some reason, a computer scheduled to synchronize automatically seeks out the second computer on the synchronization server list. At the time of the next synchronization, the computer first looks for the primary server before seeking a secondary synchronization server. If no computer is set as a primary time server, your computer runs on its own clock.

Complete the following steps to configure time synchronization.

1. Select **Options»Time Synchronization**. The following dialog box appears.
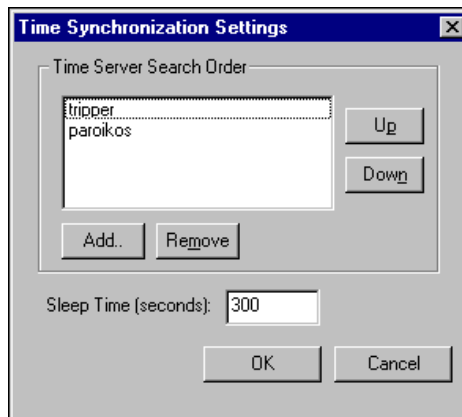
2. Determine the time synchronization order that you want to use. If you have some computers running Windows Me/98 and other computers running Windows 2000/NT/XP in your network, you should list your Windows 2000/NT/XP computers first in the server search list.

3. Set the order that you want in the **Time Server Search Order**. Any computer that is running the time synchronization service can serve as a time server or a time client. The primary time server is the first computer listed in the **Time Server Search Order** field. Do not include a computer in its own list of time synchronization services.

   • To add a computer to the **Time Server Search Order** field, click the **Add** button. If you know the name of the computer you want to add, you can type it into the **Computer name** field. If you do not know the exact name of the computer, you can browse for it in the network tree contained in the second field.

   • To remove a computer from the **Time Server Search Order** field, highlight the computer name and click the **Remove** button.

- To change the order in which your computers search for a time synchronization server, select the computer name and click the **Up** or **Down** buttons.

- Use **Sleep Time (seconds)** to set how long each computer waits between each synchronization. You should set the primary time synchronization server sleep time to 60 seconds.

Suppose you have four computers you need to have synchronized. If one fails, the others look for the next in line to synchronize to as time servers. For computers A, B, C and D, you would use the following time server search order in each computer.

Table 13-1. Time Synchronization Order

Computer A	Computer B	Computer C	Computer D
None listed	A	A	A
—	—	B	B
—	—	—	C

As the primary time server, Computer A would have no other servers listed. As long as Computer A is running, it in effect synchronizes to itself. Computer B should synchronize to Computer A as long as A is running. If A is not running, B should synchronize to itself. Computer C should synchronize to Computer A if it is running, Computer B if A is not running, and to itself if neither A not B is running. This pattern should be used for all the computers you want in one synchronized set.

4. Click **OK**.

5. Implement this same search order for all of the computers on your network running Lookout, similar to what is shown in Table 13-1.

Checking the Network Connection between Two Computers

Lookout networking, like the `ping` program, is based on TCP/IP and assumes that computer network addresses can be resolved by name. Use the `ping` program to see if two computers are properly networked before trying to run Lookout.

Run `ping` by opening a command prompt and entering

```
ping server
```

from the client and

```
ping client
```

from the server.

If you cannot successfully run ping from both computers, Lookout networking will not work.

When ping runs successfully, it produces output similar to the following:

```
>ping plato
Pinging plato.ni.com [123.45.67.89] with 32 bytes of
data:
Reply from 123.45.67.89: bytes=32 time<10ms TTL=128
Reply from 123.45.67.89: bytes=32 time<10ms TTL=128
Reply from 123.45.67.89: bytes=32 time<10ms TTL=128
Reply from 123.45.67.89: bytes=32 time<10ms TTL=128
```

If `ping` succeeds, but only after a long time, Lookout will also take a long time to make connections.

One possible cause of a slow response is an incorrect domain suffix search order. Check this in your **DNS configuration** dialog box, and correct the search order if necessary. To access DNS configuration in Windows NT 4.0, select **Start»Settings»Control Panel»Network» Protocols**. Click **TCP/IP Protocol** to open the TCP/IP properties dialog box. Select the **DNS** tab to check the search order.

To access DNS configuration in Windows Me/98, select **Start» Settings»Control Panel»Network**. Select the **Configuration** tab, and click **TCP/IP** in the list of installed components. The **TCP/IP properties** box appears. Select the **DNS Configuration** tab to check DNS search order.

Under Windows NT version 4 and Windows 2000/XP, open a command prompt and enter the command

```
nslookup computername
```

to see if DNS is working and to find out how long it takes to do the name lookup.

If `ping` does not succeed, try accessing the Network Troubleshooter by selecting **Start»Help»Troubleshooting**. This Windows utility can sometimes diagnose a problem. In some cases, it might be necessary to use the machine's fully qualified name, such as `plato.dy.ni.com`.

If the Network Troubleshooter does not resolve your problems, you might need to ask for help from a network administrator.

Monitoring Windows Services

Lookout requires three background services that run in Windows outside of the Lookout application itself to be running on your computer while it is running: Lookout CitadelService, Lookout Classifieds, and Lookout TimeService. In your Windows NT task manager, these services appear as `lkcitdl.exe`, `lkads.exe`, and `lktsrv.exe`. Under Windows NT systems, these services run automatically as NT services.

Lookout installs a National Instruments services manager, denoted by an icon located in the system tray of the Windows taskbar, near the computer clock. The columns of circles from left to right represent Classifieds, TimeService, and CitadelService. A green light indicates the service is running. A red light indicates the service is stopped. When you right-click this icon, a shortcut menu appears from which you can start or stop any of the Windows services.

Running Lookout Processes

You can run many processes at one time. You cannot, however, run two processes with the same process name on a single computer.

You may increase performance by running multiple instances of Lookout on one computer, or distributing Lookout processes on networked computers, instead of running one large process in one instance of Lookout.

Selecting Startup Process Files

If your computer runs Lookout 24 hours a day, you may want to ensure that, if the computer temporarily loses power, it will automatically reboot and begin executing your processes when power returns.

Complete the following steps to select startup processes.

1. Select **Options»Startup**, and the following dialog box appears.
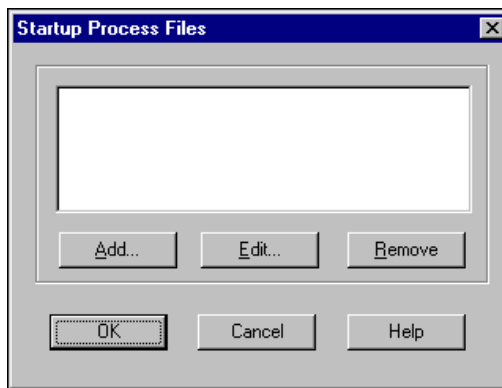
2. To add a file to your list of startup processes, click the **Add** button. A dialog box you can use to browse for a file appears.

3. Select the file you want to run when Lookout opens and click **Open**.

    You can add as many process files as you want. The files will open in the order in which they are entered in the Startup Process Files dialog box. However, the number of processes that you can load is limited to what you can list using about 2,600 characters total, which includes process names and paths. To go beyond that limitation, use a Loader object to load additional processes at startup.

    To edit a path to a file, highlight the file name and click the **Edit** button.

4. Click **OK** when you are done.

5. To make sure Lookout loads and runs when your computer boots or reboots, consult your operating system documentation instructions on how to set a default startup application.

Launching Lookout Automatically when Windows 2000/NT/XP Boots Up (NT Service)

You can configure an automatic log on in any version of Windows. With Windows NT version 4 or Windows 2000/XP, bypassing a login requires editing the registry. Alternatively, you can configure Lookout to run as an NT service so that it starts automatically anytime your Windows 2000/NT/XP computer boots up and continues to run even if you log off of Windows. In this way, Lookout is running in the background while Windows 2000/NT/XP waits for a user to log on. Follow these steps to configure Lookout to run as an NT service.

✎ **Note** NetDDE communication does not work if you are logged out of Windows NT, whether Lookout is running as an NT service or not.

Complete the following steps to enable Lookout as an NT service.

1. Select **Options»NT Service**. The NT Service dialog box appears, as shown in the following figure.

2. Select the **Start Automatically as NT Service** checkbox to enable Lookout as an NT service.

3. To configure Lookout to log on under a specific NT user account, select the **Log on as a specified user** checkbox and enter the **Account** name and/or the domain name, and a **Password**. You must enter a **Domain** if the specified user account is a domain account. You can leave **Domain** blank if the specified user account is a local account on that computer.

   If you do not enable the **Log on as specified user** option, the Lookout NT service will run under the NT LocalSystem account. This is acceptable in most cases, unless your Lookout process needs access to a resource that is available only to a certain user.

4. Configure a printer if you want anything printed while running Lookout as an NT service.

   a. Make sure you are logged on to Windows as a user who has a printer set up (configured in Windows through **Start»Control Panel»Printers**).

    b.   Click **Printer** in the **NT Service** dialog box. You cannot configure the printer from this dialog box unless you are logged into Windows with an account that has a printer set up.

    c.   Configure the printer settings. Your configuration will be maintained by Windows in the registry for use when printing from Lookout as an NT service.

5.   Click **OK**.

6.   If you want Lookout to automatically run a particular process when it starts up, select **Options»Startup** in Lookout to configure a start-up process.

Lookout will now run as an NT service anytime the computer boots up or a user logs off of Windows. To gain access to the Lookout interface, a user must log on to Windows. If a Lookout process with additional security settings is running at that time, the user may then be required to log in to Lookout in order to access that process.

Opening a Process in a Known State

Information stored in a process state file (`.l4t`) is read and used when the process is opened and updated when the process is closed. You can force a process to open in a known state every time by essentially locking the state file, as follows:

1.   Put your process into the state in which you want it always to open.

2.   In edit mode, select **File»Modify Process**.

3.   Uncheck the **Save state file(s) every** checkbox, and click **OK**.

4.   Save your process.

5.   Change the *processname*`.l4t` file attributes to be read-only. In Windows, right-click the file and choose **Properties**.

Troubleshooting and Maintenance

Fixing Broken Connections

If you rename a process or computer, rearrange folders in your processes, or move objects in a folder hierarchy, you can break Logos connections between different instances of Lookout. Besides renaming an object or changing an alias, or network malfunctions, most broken connections are path problems. You can solve most connection problems by checking the paths in your connections.

For more information about ensuring the quality of your data and connections, refer to *Data Quality Problems* section of Chapter 9, *Alarms and Events*, and the *Quality Functions* section of Chapter 7, *Expressions*.

Troubleshooting Expressions

If you have trouble getting expressions formed correctly, check the following items.

Make sure that your expressions conform to the proper Lookout syntax. Refer to Chapter 7, *Expressions*, for more information about proper syntax for expressions.

Make sure you use double quotes (") properly to set off strings.

Make sure that the path modes used in your expressions are correct and that your paths are as complete as necessary. For easier path maintenance, use Symbolic Link connections. For more information about Symbolic Links, refer to the *Symbolic Link Connections* section of Chapter 5, *Creating Lookout Processes*.

Troubleshooting Communication with Control Hardware

If you have trouble communicating with your control hardware, check the following.

- Make sure you are using the proper interface and cables. For serial communication, you can use either straight or null-modem cables. Try changing the cable type from one to the other too see if the problem is resolved. Refer to the control hardware documentation for setup information.

- Make sure that the control hardware is able to communicate with its proprietary software. For example, if you have a DirectLogic PLC, verify that it can communicate with DirectSoft software. If not, then it will not be able to communicate with Lookout. Contact the control hardware's manufacturer for technical support in troubleshooting communication with its proprietary software.

- Make sure that you are using the proper configuration settings such as baud rate and parity.

- Try increasing the driver objects's **Receive Timeout** or the serial port's **Receive Gap** setting to allow the control hardware more time before a communication alarm is produced.

- If your device requires hardware handshaking, select **Options»Serial Ports** and make sure that the COM port connected to your control hardware is set to **Radio (RTS/CTS)**. If you are connecting to remote hardware using a dial-up modem, make sure the COM port is set to **Dial-up**.

- Generate a serial diagnostic file to help you troubleshoot your communication problems, as described in the *Diagnosing Serial COM Port Problems* section of Chapter 4, *Serial COM Port Communication*.

- You can generate a diagnostic file for Modus Ethernet objects. Edit the `modbus.ini` file in the `Lookout` directory by adding a `DiagnosticPath` key to the `[ethernet]` section. Set the key equal to the path where you want to create the diagnostic file. Then restart Lookout and run your process to generate the file.

Keyboard Problems

If you use a touch screen or some other nonstandard keyboard, and you have keyboard difficulties under Windows NT, uninstall the Lookout keyboard driver as follows.

1. Select **Start»Settings»Control Panel**.

2. Open **Add/Remove Programs**.

3. Select **NI Lookout** and click **Change**.

4. Click **Add/Remove** and uninstall the **NT Keyboard Driver**.

14

Setting Up Redundancy

This chapter explains how to set up redundancy, to protect your Lookout application from computer failures.

With Lookout, you can configure two process control computers for redundancy, providing automatic transfer of control if the primary computer should fail. Redundancy only applies to systems that implement networking of multiple Lookout nodes. It does not apply to stand-alone systems.

✏️ **Note** In versions of Lookout earlier than Lookout 4, redundancy was handled through NetDDE. National Instruments recommends you use Lookout's TCP/IP-based networking when possible, but you can still use your old NetDDE-based processes with the newer version of redundancy. However, you must equip your backup computer with a monitor and load process like the one described in the *Setting up the Standby Process and Computer* section. You must also modify the process properties of both primary and standby processes to handle state file and data logging paths appropriately, as described in this chapter.

No other alterations should be necessary to upgrade your redundancy processes, though you should test the new arrangement thoroughly. The need remains for following NetDDE naming conventions as well as the other instructions given in Appendix E, *Networking with DDE*. If your client process uses NetDDE in shifting access from one computer to another, it will continue to work in the current version of Lookout.

You set up computer redundancy using the Lookout Monitor, Loader, and Symbolic Link objects, and by setting up a network configuration in which one computer is designated the *primary* computer and another is designated the *standby* computer. The primary computer normally monitors and controls the process while the standby computer monitors the primary computer.

If the standby computer stops receiving information from the primary computer, the standby computer automatically takes over and assumes the role of the primary computer. This is referred to as a *failover*. From that point on, the standby computer assumes the role of the primary computer, and all other computers running Lookout processes automatically recognize this fact. At the same time, client processes accessing a server

running on the primary computer switch from accessing the primary to accessing the standby. When the primary computer comes back up on the network, the standby computer and any client processes affected return to their previous operation.

Standby System Configuration

Figure 14-1 depicts a typical Lookout network structure with the standby option configured. The dotted lines show how the hardware can be controlled and monitored by *either* machine in a redundant system.
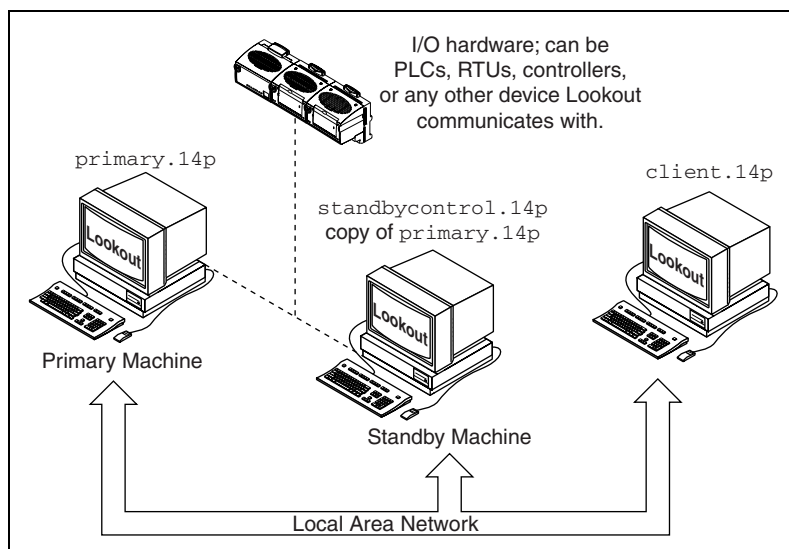
Figure 14-1. Standby Network Configuration

There are several basic principles assumed in Figure 14-1.

- The computers used to implement redundancy must be properly set up on your network and able to communicate with each other. Refer to Chapter 13, *Networking and Deploying an Application*, for more information about ensuring communication between networked computers.

- Both the primary *and* standby computers must have direct access to your control hardware. Your standby computer must have the same direct access to all I/O as the primary computer, because it may assume the responsibility of the primary computer. There are several ways to accomplish this, depending on your hardware, network, and communications topology.

- You can designate any computer on the network as the standby computer. It does not need to be the closest physical computer on the network, though this is typically the case since the standby computer must be connected to the control hardware.

- The standby computer must have a copy of the primary computer process file in its `Lookout` directory, `primary.14p` in this example. As soon as the standby computer fails to receive a signal from the primary computer, it automatically loads its local copy of the `primary.14p` file.

- In a standby system configuration, the primary computer not only updates its own state file, but also the state file of the standby computer.

Failover Scenarios

There are four reasons for the standby computer to take control of the process.

- The primary computer is down (that is, it is not turned on, Windows is not running, or an application fault has locked the computer).

- The primary computer is running, but it is not running Lookout.

- The primary computer is running Lookout, but it is not running the correct process file.

- The primary computer is running the correct process file within Lookout, but the two computers cannot communicate over the network.

Enabling File Sharing for State File Transfer

One part of proper redundancy is making sure that Lookout has the most recent state file for a process available when it loads that process. For this to occur, your computers must have file sharing permission with each other.

1. Configure both your primary and standby computers to share files with each other. Refer to your Windows documentation for information about setting up file sharing.
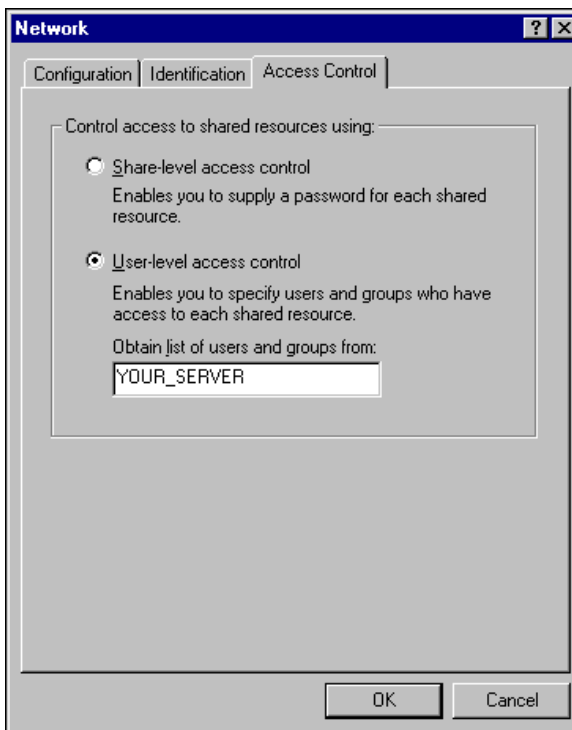
2. On the standby computer, configure the directory containing the state file (default is the Lookout folder) to be shared. In Windows, right-click the folder and choose **Sharing** to access the folder properties dialog box.

3. Click **Add** and add the primary computer. Make sure full access privileges are granted.

4. Click **OK**.

Setting up the Primary Process and Computer

Complete the following steps to set up your primary computer and server process for redundancy.

1. Ensure that your standby process will be able to open with the most current Lookout state file (.l4t) available. To do this, configure the **State Information** options of your process as follows.

    a. If you are setting up redundancy on an existing process, right-click the process name in Object Explorer and select **Properties**. If you are creating a new process, you will see the same dialog box.

b. Select **Save State File with Process File** to save the state file in the location from which the process file was opened. This is the local state file for your process.

c. Select the **Save Standby State File** checkbox to save one (or more) copies of the state file for your standby process. Enter a complete path, including state file name, to each location where you want to save a state file on the standby computer. If you are saving the state file to more than one standby or back-up computer, separate the paths with the vertical bar (|) operator symbol. The state file the standby computer uses to initialize its version of your process file must be in the directory containing the standby copy of the process file.

d. For this to work, you must have given permission for your primary computer to write to the Lookout directory of the standby computer, as described in the previous section, *Enabling File Sharing for State File Transfer*.

e. Set how often Lookout saves the state file in the **Save state file(s) every** *[xxxx]* **minutes** field. The Lookout default is 60 minutes. Your state file save rate should be the same in your primary and standby processes.

f. Click **OK**.

When your process opens, Lookout checks the dates of the state files you have selected and uses the most recent file as the state file to initialize values in your process.

2. Make sure that when your primary computer comes back online after being down that Lookout loads as soon as the computer boots, and that Lookout then loads the proper process. To make sure your computer launches Lookout when it boots, refer to your operating system documentation. To ensure that the proper process is loaded automatically, refer to the *Selecting Startup Process Files* section of Chapter 13, *Networking and Deploying an Application*.

3. Repeat these steps for each process running on your primary computer that you need to run on your standby computer.

Setting up the Standby Process and Computer

Complete the following steps to set up your primary computer and server process for redundancy.

1. Ensure that your standby computer is connected to the network, is connected to the control hardware, and has Lookout installed.

2. Copy all of the necessary process files to the standby computer. The Lookout directory on the standby computer should include the same contents and structure as the Lookout directory on the primary computer.

3. The standby Lookout must run a separate process to monitor the process being run on the primary computer. This process loads the standby computer's copy of the primary process when data ceases to be available from the primary computer, and unloads the it when the primary computer comes back online. Create this standby monitoring process using the following steps.

    a. Create a new process on the standby computer.

    b. Create a Monitor object. The following dialog box appears.
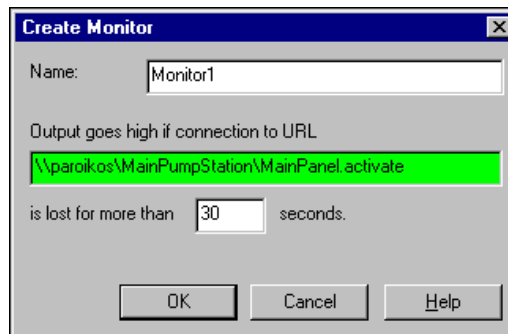
Enter the URL to some data member in the process on the primary computer. The value of this data member does not matter, nor does it matter if that data member is used by any other Lookout object. If the Monitor object does not detect the presence of that data member for the amount of time you enter in the **seconds** field, the Monitor output goes TRUE. Then the Monitor object signals the other elements of the Lookout redundancy system into action. When the primary computer and process reappear on the network, the Monitor object signals the standby process and clients to return to their previous mode of operation.

c. Create a Loader object. The following dialog box appears.

When using the Loader for redundancy, set the **Process Name** the same as the name of the primary process, and set the **Process File** the same as the primary process file name. This Loader object loads and unloads the standby process on command from the Monitor object.

Because you earlier set the state file to be located in the Lookout directory of your standby computer, you should select the **Save State File in Lookout Folder** option. If you have reason to save copies of the state file in other locations as well, you can select the **Save Standby State File** option.

Note Settings in the Loader object override any settings you might have made when you created or set the properties for your process.

Set how often you want the state file saved in the **minutes** field.

Set the location for your Citadel historical data logging to take place, just as you would when creating a new process. Refer to the next section for more information about ensuring data continuity.

Enter `Monitor=TRUE` in the **Load** field and `Monitor=FALSE` in the **Unload** field, where `Monitor` is the name of the Monitor object.

4. Click **OK**.

5. Save the process.

Client processes are redirected from interaction with the primary process to interaction with the secondary process, when signaled by a Monitor object. When the Monitor object signals that the primary process is back online, the clients switch back to interacting with the primary process.

Ensuring Historical Data Continuity

If you depend on only the primary computer to log data, a failover means that data is no longer being logged. To ensure data continuity, make sure your second computer will log data when it takes over for the failed computer. When your primary computer is restored, you can archive the data logged from the standby computer and merge it with the data logged from the primary computer. If you want the traces to show the same name throughout the failover scenario, you can rename them from the standby computer before you archive them.

For full data continuity, use the CitadelControl object to archive your data regularly so that it is properly backed up to another path in case your primary logging database is damaged.

Configuring Clients for Standby Operation

You may have any number of Lookout client processes running on other computers in your Lookout network. These clients must respond to a change from primary to standby computers.

Lookout clients created for use with a redundancy-equipped system should use the Symbolic Link to switch your client process access from one server process to another. The Symbolic Link can shift from primary to standby process when necessary and your client continues to function, having made its connections through the Symbolic Link.

Complete the following steps to set up your clients to work in a failover scenario.

1. Create a Monitor object, just as you did with the standby control process. Use the same settings for the Monitor object as you did in the standby monitoring process on your standby computer, and click **OK**.

2. Create a Symbolic Link object. The following dialog box appears. Use the **Dynamic** field as the source, and enter an IF expression function that points to the primary computer and standby computer.
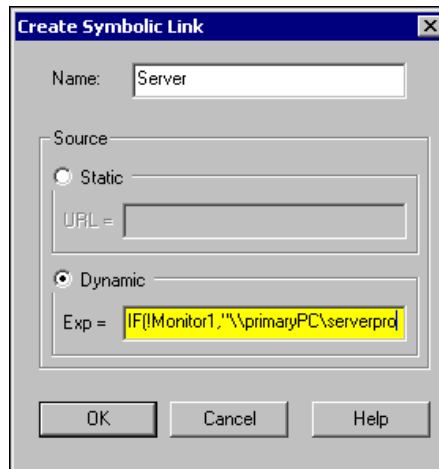
In this illustration, the following expression is used:

```
IF(!Monitor1,"\\primaryPC\serverprocess","\\standbyPC\serverprocess")
```

This expression states that if the (implicit) value of the Monitor object is *not* TRUE (indicated by the exclamation point), the client is connected to the server process running on the primary computer. If the Monitor object goes TRUE, your client process will connect to objects in the server process running on the standby computer.

After you have made this preparation, develop your client process as you ordinarily would, except that you make your connections to the server process through your Symbolic Link instead of the process itself. Use your Symbolic Link to insert expressions, create remote source controls, and display historical data in a HyperTrend.

Configuring Time Synchronization for a Redundant System

If timing and time-stamped data are crucial to your backup system, consider synchronizing both your primary and standby computers to a third computer. In any case, you should ensure that your primary, standby, and client computers are all appropriately synchronized. Refer to the *Synchronizing Lookout Computers* section of Chapter 13, *Networking and Deploying an Application*, for more information about time synchronization.

A

Numeric, Time, and Date Formats

This appendix describes the format for numeric displays, including time and date values.

Numeric Display Formats

If you display a numeric signal on a control panel, Lookout provides several numeric formats to choose from. The different possible formats are listed in the following tables.

Table A-1. General Numeric Format Examples

Numeric Format	Value	Display
(General)	123.78	Displayed as 123.78
(General)	123.789	Displayed as 123.789
Note: General format displays the value in the most compact form possible.		

Table A-2. Leading Zeroes Numeric Format Examples

Numeric Format	Value	Display
000000000	123.789	Displayed as 000000124
00000000	123.789	Displayed as 00000124
0000000	123.789	Displayed as 0000124
000000	123.789	Displayed as 000124
00000	123.789	Displayed as 00124
0000	123.789	Displayed as 0124
000	123.789	Displayed as 124
00	123.789	Displayed as 124
0	123.789	Displayed as 124

Table A-3. Fractional Numbers with Trailing Zeroes Format Examples

Numeric Format	Value	Display
0.0	123.789	Displayed as 123.8
0.00	123.789	Displayed as 123.79
0.000	123.789	Displayed as 123.789
0.0000	123.789	Displayed as 123.7890
0.00000	123.789	Displayed as 123.78900
0.000000	123.789	Displayed as 123.789000
0.0000000	123.789	Displayed as 123.7890000
0.00000000	123.789	Displayed as 123.78900000

Table A-4. Exponential/Scientific Notation Format Examples

Numeric Format	Value	Display
0E0	123.789	Displayed as 1E+2
0.0E+0	123.789	Displayed as 1.2E+2
0.00E+0	123.789	Displayed as 1.24E+2
0.000E+0	123.789	Displayed as 1.238E+2
0.0000E+0	123.789	Displayed as 1.2379E+2
0.00000E+0	123.789	Displayed as 1.23789E+2
0.000000E+0	123.789	Displayed as 1.237890E+2
0.0000000E+0	123.789	Displayed as 1.2378900E+2
0.00000000E+0	123.789	Displayed as 1.23789000E+2

Table A-5. Hexadecimal Format Examples

Numeric Format	Value	Display
0x0	123.789	Displayed as 0x7B
0x00	123.789	Displayed as 0x7B
0x000	123.789	Displayed as 0x07B
0x0000	123.789	Displayed as 0x007B
0x00000	123.789	Displayed as 0x0007B
0x000000	123.789	Displayed as 0x00007B
0x0000000	123.789	Displayed as 0x000007B
0x00000000	123.789	Displayed as 0x0000007B

Time and Date Values

The following items are true for time values in Lookout:

- Time values are stored as numeric values with the following format. If there are no colons in the entry, the time period is assumed to be given in days.

  days:hours:minutes:seconds

- Because dates and times are represented by numeric values, you can add, subtract, and include dates and times in expressions, just as you would any other numeric signals.

- When you choose a format for a time value, lowercase formats (hh:mm:ss) indicate a specific point in time, and uppercase formats (HH:MM:SS) indicate a time span.

- For points in time, Lookout uses the 1900 date system in which the number 1 corresponds to midnight, January 1, 1900. The number 2 corresponds to midnight, January 2, 1900 and so on. For example, the number 34491.5 represents noon, June 6, 1994.

- When your operating system switches in or out of daylight savings time, Lookout corrects for the change relative to universal time so that there is no data discontinuity or loss in historical data.

- You may enter one hour as 1:00:00, but Lookout stores the number as 0.04167 (or 1/24 of a day). Days are represented by the integer portion of the number. You may also find it helpful to know that one second = 0.000011574 and one minute = 0.000694444.

The following table shows time constant examples.

Table A-6. Examples of Lookout Time Constants

Time Period Constant	Lookout Interpretation
0:23	23 seconds, or 0.0002662 days
75:00	75 minutes, 0 seconds, or 0.05208 days
12:00:05:01	12 days, 0 hours, 5 minutes, 1 second, or 12.003484 days
199::	199 hours, 0 minutes, 0 seconds, or 8.2917 days
0:10.023	10.023 seconds, or 0.0001160 days
12.75	12.75 days
17:64:22.5	invalid number: because hours are specified, minutes must be ≤59

Time spans indicate a period of time as hours, minutes, seconds, and fractions of a second. Time spans are specified with uppercase letters, as shown in the following table.

Table A-7. Time Span Display Examples

Time Format	Value	Display
H	0.4789	Displayed as 11 (hours)
H.H	0.4789	Displayed as 11.5 (hours)
H.HH	0.4789	Displayed as 11.49 (hours)
M	0.4789	Displayed as 690 (minutes)
M.M	0.4789	Displayed as 689.6 (minutes)
M.MM	0.4789	Displayed as 689.62 (minutes)
S	0.4789	Displayed as 41377 (seconds)
S.S	0.4789	Displayed as 41377.0 (seconds)
S.SS	0.4789	Displayed as 41376.96 (seconds)
HH:MM	0.4789	Displayed as 11:29 (11 hours, 29 minutes)

Table A-7. Time Span Display Examples (Continued)

Time Format	Value	Display
HH:MM:SS	0.4789	Displayed as 11:29:36 (11 hours, 29 min, 36 seconds)
HH:MM:SS.S	0.4789	Displayed as 11:29:36.9
HH:MM:SS.SS	0.4789	Displayed as 11:29:36.96
HH:MM:SS.SSS	0.4789	Displayed as 11:29:36.960
MM:SS	0.4789	Displayed as 689:36 (689 minutes, 36 seconds)
MM:SS.S	0.4789	Displayed as 689:36.9
MM:SS.SS	0.4789	Displayed as 689:36.96
MM:SS.SSS	0.4789	Displayed as 689:36.960

Absolute times and dates indicate a specific moment in time. Absolute times are specified with lowercase letters, as shown in the following table.

Table A-8. Absolute Date and Time Display Examples

Time Format	Value	Display
hh:mm	34668.7889	Displayed as 18:56 (6:56 p.m.)
hh:mm:ss	34668.7889	Displayed as 18:56:02
mm/dd hh:mm	34668.7889	Displayed as 11/30 18:56
mm/dd hh:mm:ss	34668.7889	Displayed as 11/30 18:56:02
mm/dd/yy	34668.7889	Displayed as 11/30/94
mm/dd/yy hh:mm	34668.7889	Displayed as 11/30/94 18:56
mm/dd/yy hh:mm:ss	34668.7889	Displayed as 11/30/94 18:56:02
dd/mm hh:mm	34668.7889	Displayed as 30/11 18:56
dd/mm hh:mm:ss	34668.7889	Displayed as 30/11 18:56:02
dd/mm/yy	34668.7889	Displayed as 30/11/94

Table A-8. Absolute Date and Time Display Examples (Continued)

Time Format	Value	Display
dd/mm/yy hh:mm	34668.7889	Displayed as 30/11/94 18:56
dd/mm/yy hh:mm:ss	34668.7889	Displayed as 30/11/94 18:56:02

B

Structured Query Language

This appendix describes Structured Query Language (SQL), Open Database Connectivity (ODBC), and accessing Citadel data using both SQL and ODBC.

Introduction

The Citadel historical database includes an Open Database Connectivity (ODBC) driver, which enables other applications to directly retrieve data from Citadel using Structured Query Language (SQL) queries.

What is ODBC?

ODBC is a standard developed by Microsoft. It defines the mechanisms for accessing data residing in database management systems (DBMSs). Nearly all Windows-based applications that can retrieve data from a database support ODBC.

Note Some applications are not completely ODBC compliant. If you plan to use Microsoft Query, Microsoft Access, or Microsoft Visual Basic, ensure that the maximum column name length does not exceed 62 characters. These applications cannot handle longer names. Applications that are completely ODBC-compliant can handle names up to 126 characters long. All traces whose names exceed the maximum column name length are excluded from queries.

Because Lookout objects may include network path information in their names, and because you can arrange Lookout objects in hierarchies inside your processes, you may find it easy to exceed the 62 and 126 character limitations of ODBC. Consideration given to naming and organizing objects can minimize the risk of encountering this difficulty.

What is SQL?

Structured Query Language (SQL) is an industry-standard language used for retrieving, updating, and managing data. In Lookout, you can use SQL to build queries to extract data from Citadel. The Citadel ODBC driver also includes many built-in data transforms to simplify statistical analysis of retrieved data.

✎   **Note**   SQL is a complex subject. If you are going to perform elaborate data searches using SQL, refer to an SQL reference manual.

Creating a Citadel ODBC Data Source

Complete the following steps to create a Citadel ODBC data source for use with Lookout.

1. Click the Windows **Start** button and select **Settings»Control Panel**.

2. Run the ODBC applet. It might be called **ODBC** or **ODBC Data Sources** or something similar, depending on your operating system version number.

✎   **Note**   Shut down all ODBC applications, such as databases, spreadsheets, word processors, and Microsoft Query, before you run the ODBC applet.
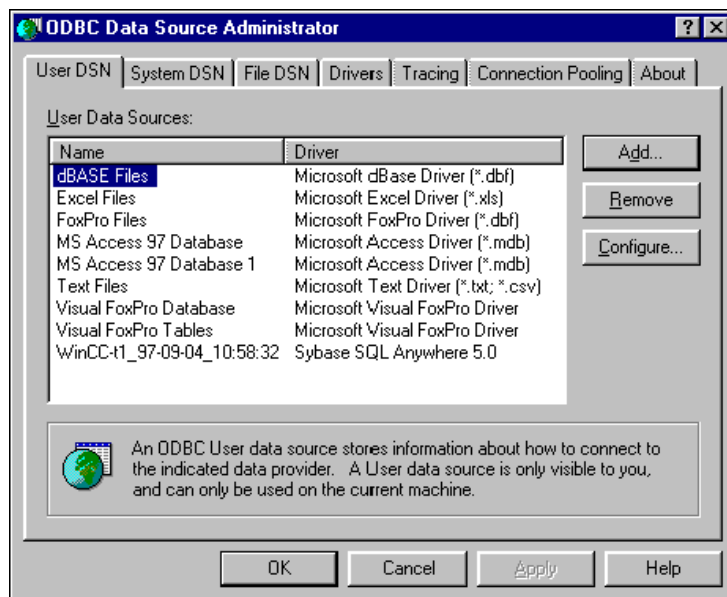
3. Select the **User DSN** tab or the **System DSN** tab, depending on which type of data source you want to create. User DSNs are only visible to the user who created them on the current machine. System DSNs are available to all users on the current machine.

4.   Click the **Add** button. The following dialog box appears.
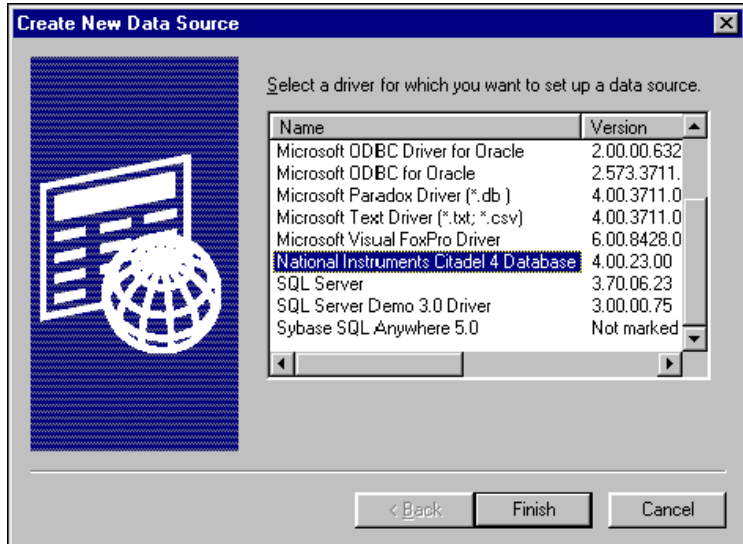
5.   Select **National Instruments Citadel 4 Database**, then click **Finish**.

6.   In the **National Instruments Citadel ODBC Setup** dialog box, fill in the **Data Source Name**, **Description**, and **Database Path** fields.

   a.   The **Data Source Name** is the name that ODBC applications use to select your data source.

   b.   **Description** is a free-form text string you can enter to describe the data source.

   c.   **Database Path** should match the location of the Citadel database you intend to access. Use a fully qualified path to a remote computer if you are accessing the Citadel database on a remote computer, such as `\\Tripper\c:\lookout\database`.
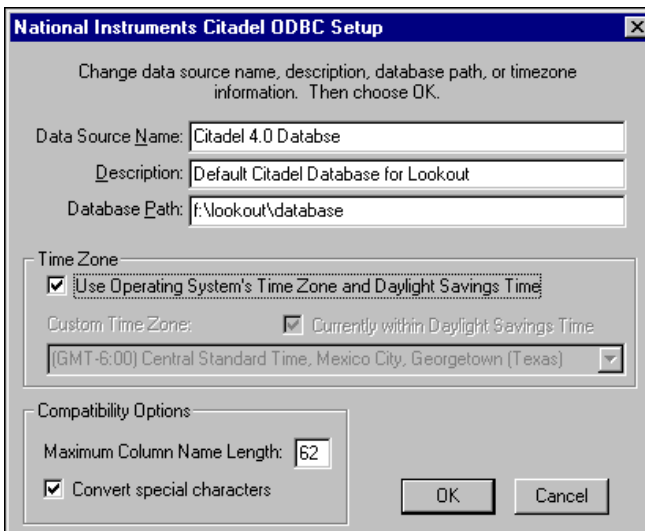
The **Data Source Name** must be different from any other ODBC data source name. The **Description** is arbitrary. The **Database Path** gives the location of the folder where the data for this source is stored. In Lookout 4 and later, it is possible to configure multiple Citadel databases. If you do so, you probably want to configure one ODBC data source for each. If you used the **Options»System** dialog box in Lookout to configure a default Citadel database on the local computer, you should create a data source for the location specified in the **Default Path** field in the Lookout System Options dialog box. If you used **File»Modify Process** in Lookout to configure a different database for a process, you probably want to create a separate data source for that process using the location specified in the **Citadel Database Folder** field of the Lookout Modify Process dialog box.

7. Click **OK** in the **Setup** dialog box, then click **OK** in the **ODBC Data Source Administrator** dialog box.

✏️ **Note**   Some applications are not completely ODBC compliant. If you plan to use Microsoft Query, Microsoft Access, or Visual Basic, make sure **Maximum Column Name Length** does not exceed 62 characters. These applications cannot handle longer names. Applications that are completely ODBC compliant can handle names up to 126 characters long. All traces whose names exceed the **Maximum Column Name Length** are excluded from queries. Because Lookout objects may include network path information in their names, and because you can arrange Lookout objects in hierarchies inside your processes, you may find it easy to exceed the 62 and 126 character limitations of ODBC. Consideration given to naming and organizing objects can minimize the risk of encountering this difficulty.

If you plan to use Microsoft Access or Visual Basic, select **Convert special characters** to force Lookout names into an accepted format by replacing characters within the names with the characters in Table B-1.

The special characters changed in Lookout 4. If you are converting Lookout processes from a version earlier than Lookout 4, you might have to rewrite any SQL queries you set up in your earlier processes.

Table B-1. Special Access SQL Characters

Special Character	Converted Character
period (.)	at sign (@)

Accessing Citadel Data with SQL

Traces Table

The ODBC driver presents Citadel data to other applications as a traces table. The table contains a field or column for each data member logged to the Citadel database and three fields you can use to specify query criteria and to time stamp retrieved data: **Interval**, **LocalTime**, and **UTCTime**.

Interval specifies the query value sample rate. **Interval** can range from 10 ms to several years. **Interval** defaults to 1 (one day). **WEEK** is a standard seven days, but **MONTH** and **YEAR** account for different month lengths and leap years.

Because Citadel is event-driven, it only logs a value when the value changes. Using **Interval**, you can query Citadel for values evenly spaced over a period of time.

LocalTime and **UTCTime** are time-stamps that indicate when values are logged. Citadel stores the time in **UTCTime** format and derives **LocalTime** from the stored time.

The following WHERE clause query uses **Interval** and **LocalTime** to select data over a specified time at one-minute intervals. Notice that time and date formats are the same as those used in Lookout.

```
SELECT * FROM Traces
WHERE LocalTime>"12/1 10:00"
   AND LocalTime<"12/2 13:00"
   AND Interval="1:00"
```

Points Table

The Points table is used to retrieve the actual values logged by Lookout for a data member and the times they were logged. As Lookout logs values for data members asynchronously, there is no correlation between the timestamps for one data member and another. For this reason, when querying the Points table, you may only query one data member at a time.

The `WHERE` clause using **LocalTime** and **UTCTime** is supported for the Points table; however, **Interval** is not relevant to the Points table. The data transforms are also not relevant to the Points table and are not supported.

An example of a query using the Points table could be

```
SELECT LocalTime, "\\computername\process\Pot1" FROM
Points
WHERE LocalTime > "12/1 10:00" AND
LocalTime < "12/2 10:00"
```

Data Transforms

Your queries can include special commands that perform data transforms to manipulate and analyze historical data. Table B-2 lists data transform commands.

Table B-2. Data Transform Commands

Command	Transformation
Min{Datapoint}	Returns the minimum for *Datapoint* across the interval.
Max{Datapoint}	Returns the maximum for *Datapoint* across the interval.
Avg{Datapoint}	Returns the average for *Datapoint* across the interval.
Stdev{Datapoint}	Returns the standard deviation for *Datapoint* across the interval.
Starts{Datapoint}	Returns the number of starts (that is, the number of transitions from OFF to ON) for *Datapoint* across the interval. For numeric points, 0.0 is interpreted as OFF, and all other numbers are treated as ON.
Stops{Datapoint}	Returns the number of stops (that is, the number of transitions from ON to OFF) for *Datapoint* across the interval.

Table B-2. Data Transform Commands (Continued)

Command	Transformation
ETM{Datapoint}	Returns the amount of time *Datapoint* was in the ON state across the interval.
Qual{Datapoint}	There might be gaps in the historical data traces in Citadel because of machine shutdown, Lookout shutdown, or a similar occurrences. Qual returns the ratio of time for which valid data exists for *Datapoint* across the interval to the length of the interval itself. If valid data exists for only one-half of the interval, Qual returns 0.5.

Using these data transforms, you can directly calculate and retrieve complex information from the database such as averages and standard deviations, so you do not need to extract raw data and then manipulate it in another application.

For example, you need to know how many times a compressor motor started in December. You also need to know its total runtime for the month. Use the following query to get your answers:

```
SELECT
"Starts{\\computername\processname\PLC.MotorRun}",
        "ETM{\\computername\processname\PLC.MotorRun}"
FROM Traces
WHERE LocalTime>="12/1/95"
   AND LocalTime<"1/1/96"
   AND Interval="31"
```

SQL Examples

The following examples are typical query statements; however, your queries might be much more involved, depending on your system requirements.

```
SELECT *
FROM Traces
```

Retrieves the current value of every data member logged to Citadel. Because your query does not occur at the same moment in time as a PLC poll, signals scanned from PLCs are not included in the
`retrieved data.`

```
SELECT *
FROM Traces
WHERE Interval="0:01"
```

Retrieves the value of every data member logged today in one-second increments. Notice that the interval value is enclosed in quotation marks.

```
SELECT LocalTime, "\\computername\processname\Pot1"
FROM Traces
WHERE LocalTime>"12/2/2000 16:50"
AND Interval="0:01"
```

Retrieves and time stamps the value of Pot1 in one-second increments from 4:50 p.m. to now. Names are enclosed by quotes.

```
SELECT LocalTime, "\\computername\processname\AB1.I:3",
"Max{\\computername\processname\AB1.I:3}"
FROM Traces
WHERE LocalTime>"10/1/95"
AND LocalTime<"11/1/95"
AND Interval="1:00"
```

Retrieves and time stamps an Allen-Bradley PLC input in one-minute intervals for the month of October. This query also indicates the highest occurring input value of each minute.

```
SELECT LocalTime,
"\\computername\processname\OVEN1_SP",
"\\computername\processname\PLC.OVEN1_PV",
"Max{\\computername\processname\PLC.OVEN1_PV}",
"Min{\\computername\processname\PLC.OVEN1_PV}",
"Avg{\\computername\processname\PLC.OVEN1_PV}"
FROM Traces
WHERE LocalTime>"12/2/2000 14:00"
    AND LocalTime<="12/2/2000 17:00"
    AND Interval="1:00:00"
```

Retrieves an oven temperature at 3:00 p.m., 4:00 p.m., and 5:00 p.m. and shows the highest, lowest, and average temperatures for every hour between 2:00 p.m. and 3:00 p.m., 3:00 p.m. and 4:00 p.m., and 4:00 p.m. and 5:00 p.m.

Accessing Citadel Data with Microsoft Query

1. Launch Microsoft Query.

2. Choose **File»New**. If the Citadel data source is not visible, you must create it. If it is visible, skip to step 3.

   a. Select the <New Data Source> entry in the **Databases** tab. Make sure the **Use the Query Wizard to create/edit queries** option is *not* selected.
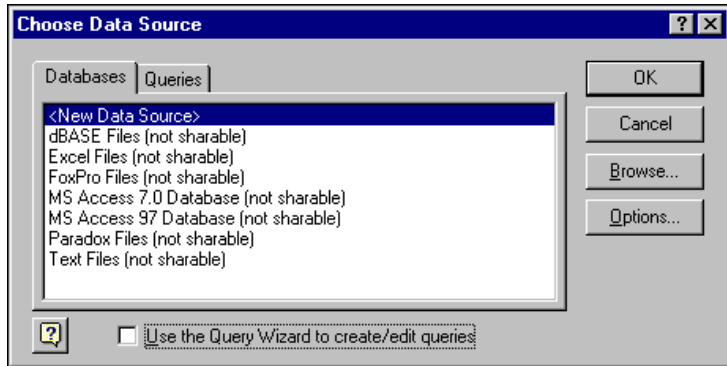
   b. The Create New Data Source dialog box appears. Fill in the fields as shown in the following illustration.
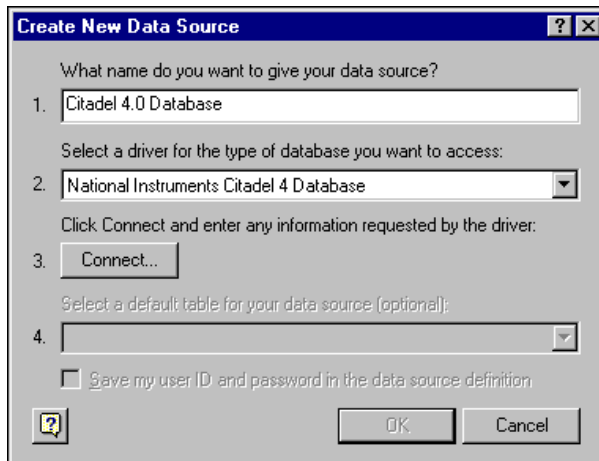
   c. Click the **Connect** button. The National Instruments Citadel ODBC Setup dialog box appears. Fill in the fields as shown in the following illustration. Enter the location of your database in the **Database Path** field.
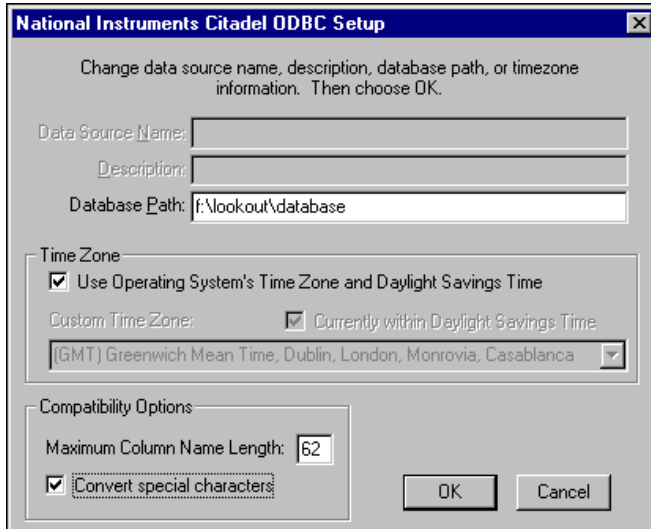
d. Click **OK**. The Create New Data Source dialog box reappears, and should look like the following illustration.
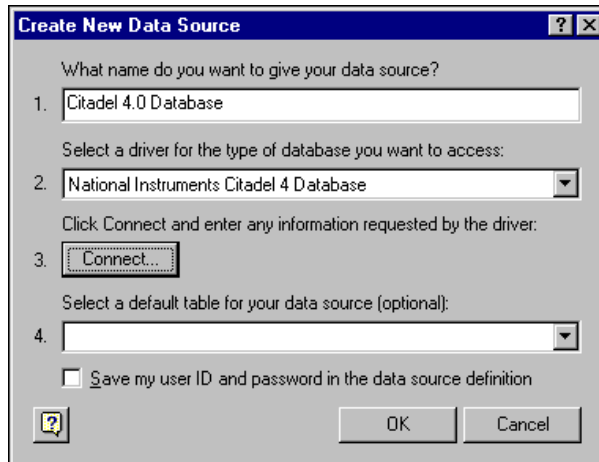
e. Do *not* select a default table or save your ID and password. Click **OK**. The Choose Data Source dialog box appears, this time with the Citadel database as one of your data source choices. Choose Citadel as your data source and click **OK**.
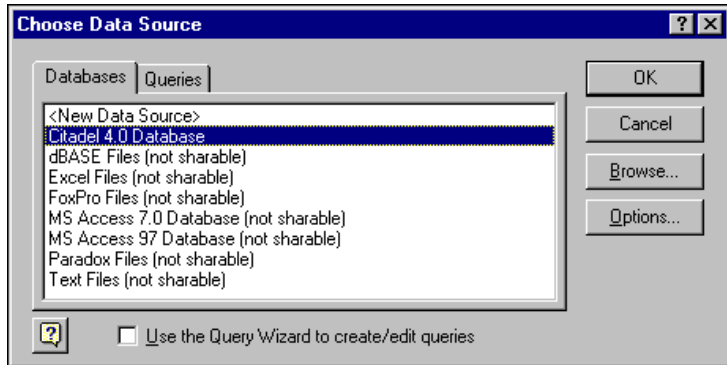
3. Select the Citadel data source. Make sure the **Use the Query Wizard to create/edit queries** option is *not* selected.

✎ **Note** If Microsoft Query is unable to connect to Citadel, make sure you have logged data to Citadel and entered the correct database path in the ODBC Setup dialog box.

✎ **Note** If Citadel is not listed in the Data Source dialog box, make sure you have created Citadel as a data source. Refer to the *Creating a Citadel ODBC Data Source* section for more information about creating Citadel as a data source.

4. In the **Add Tables** dialog box, double-click Traces or Points, depending on whether you want to view raw data (Points) or resampled data (Traces). In the following figure, Traces are used.
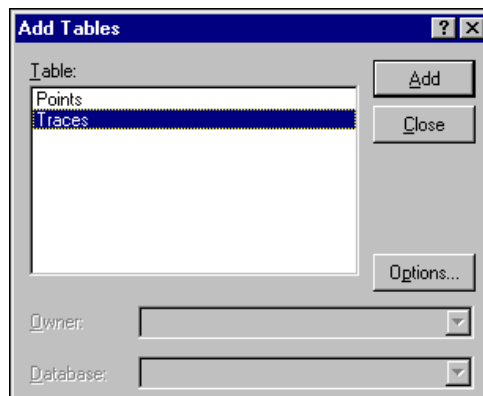
5. Close the dialog box.

Microsoft Query presents the full Query Window with the `Traces` and `Points` tables as shown in Figure B-1. The names in these tables are a comprehensive list of all name values that have been logged to Citadel.
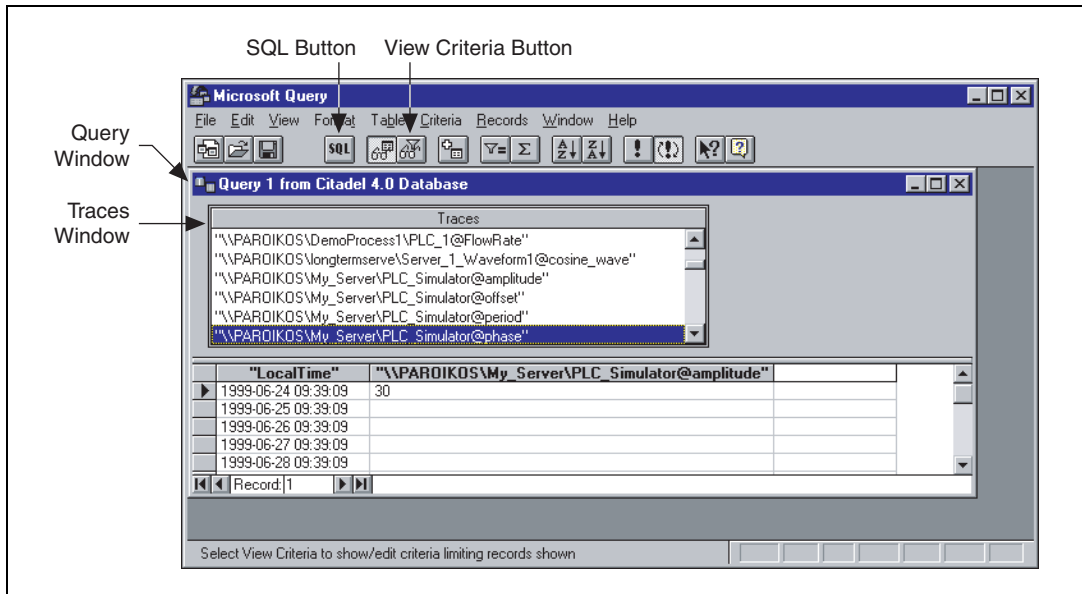
Figure B-1. The Query Window with Traces Table

6. To view a trace, double-click or drag the field you want to view to a blank column in the data pane. In Figure B-1, `LocalTime` and `"\\PAROIKOS\My_Server\PLC_Simulator@amplitude"` have been dragged down.

7. To view a data transform value, enter the function directly into a blank column. For example, to view the minimum value of `PLC_Simulator.amplitude`, you would enter `"min{\\PAROIKOS\My_Server\PLC_Simulator@amplitude}"`. You must include the quotation marks and braces.

   The data set in Figure B-1 was retrieved using no specific criteria, so the ODBC driver used the default. Although there are several ways to specify criteria, this example uses the criteria pane.

8. Click the **View Criteria** button. The pane appears in the **Query** window.

9. Add a field to the criteria pane by double-clicking the field, or by dragging it to the blank column in the criteria pane. In Figure B-2, an Interval criteria of one minute was chosen.
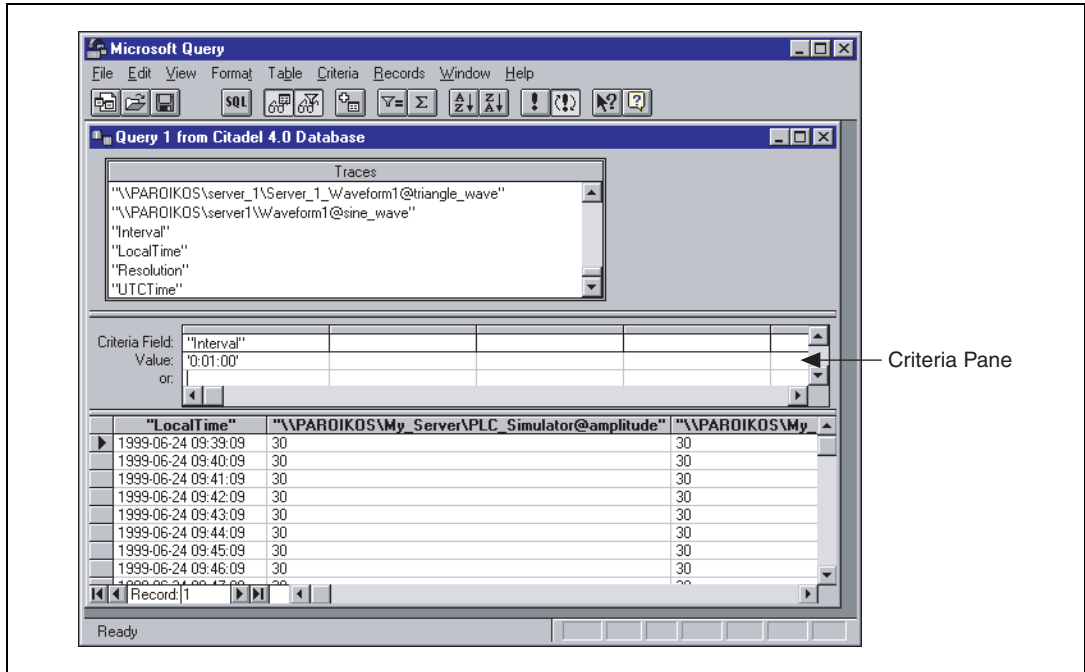
Figure B-2. The Query Window with Interval Criteria

When you enter qualifying criteria values, use the syntax demonstrated in *SQL Examples* earlier in this chapter. Refer to the following figure for more information.

As soon as you specify criteria, Microsoft Query retrieves the specified data. You can save your query at any stage of development, and as you build your query, the application builds an SQL statement.

10. Click the **SQL** button to view or edit the query statement.
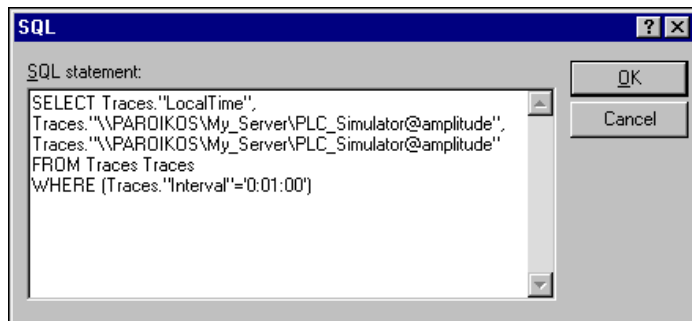
Accessing Citadel Data from Other Software

Refer to NI Developer Zone at `ni.com/zone` for information about using Citadel with other software such as Microsoft Excel, Microsoft Access, or Visual Basic. Use a search phrase such as `Citadel SQL`.

C

CBL Compiler

This appendix explains how to use the CBL command-line compiler to compile a `.lks` file into a `.l4p` file.

Lookout automatically compiles source files when you open the `.lks` file. If an automatic compile fails, you can also use Lookout's CBL compiler to manually compile a Lookout source file from a command prompt. You can also generate a `.lks` file by hand or modify an existing `.lks` file and use the compiler to produce a `.l4p` file.

If the source file has been corrupted, you can view the error file for compilation errors and either repair the source file or delete corrupted sections and rebuild your process from an intact foundation.

CBL is a command-line compiler that takes three arguments: sourcefile, targetfile, and error log file. To use it, type the following command at the command prompt, where *file* is the name of your file:

```
cbl file.lks file.l4p file.err
```

This command compiles the `.lks` file into the `.l4p` file and writes any error messages into the error file. If `file.err` is not included in the command line, errors will appear at the command prompt. If the compiler generates error messages, you need to debug your object or your `.lks` file.

You can directly edit your `.lks` file with any text editor. In general, if there is no simple repair to be made as indicated by an error message, you can often repair a file by deleting the problem object and recreating it in Lookout. National Instruments does not recommend editing the `.lks` file. If you must edit the `.lks` file, refer to the object topics in the *Lookout Help* for information helpful for editing object parameters.

CBL Compiler Error Messages

You can use the `.err` file to track down compilation errors. The error file records errors, one error per line, with a number at the beginning of each error message. This number is the line number of your `.lks` file where this error was encountered. Go to these line numbers in your `.lks` file to correct the errors before recompiling your `.lks` file.

The following is a partial list of the error messages that you might encounter, what might be causing them, and how they can be corrected.

Class not found: *class_name*

The `.lks` file referred to an object for which no corresponding `.cbx` file exists. The object name might be spelled incorrectly, or the `lookout.dat` file might need to be removed so that Lookout loads the `.cbx` file. For example, the line

```
Foo1 = new Foo ();
```

will produce the error

```
Class not found: Foo
```

Not a member of *object name***:** *member_name*

The name used is not recognized as a valid member name for the object. For example, the line

```
Pot1.fred = 3;
```

produces the error

```
error: Not a member of 'Pot1': 'fred'
```

Member is not writable for *object name***:** *member_name*

The data member is read-only. For example, the line

```
Modbus1.update = 3;
```

produces the error

```
error: Member is not writable for class
'Modbus':'update'
```

Name is not valid

The name is not valid. For example, the line

```
Foo@ = new Pot (0, 100, 1, yes, 0);
```

produces the error

```
error: Tag is not valid: Foo@
```

Member is not readable for *object*: *member_name*

The data member is write-only. For example, the line

```
Modbus1.1 = Pot1.increment;
```

produces the error

```
error: Member is not readable for class
'Pot':'increment'
```

D

Lookout INI File Settings

This appendix describes the lookout configurations contained in the `lookout.ini` file, located in your main `Lookout` directory.

While it is generally best to configure Lookout through dialog box options, you can edit this file yourself with any standard text editor. The `.ini` file sections and entries are listed in this appendix.

The syntax for a `.ini` file setting is *Key=setting*, where *Key* represents the particular feature or function of Lookout you want configured, and *setting* represents the configuration value or choice.

Table D-1. Lookout.ini File Sections, Keys, and Settings

Section	Key	Setting
[System]	NoReloadOfCBXes	=0 CBXs are not reloaded automatically (default)
		=1 CBXs are reloaded automatically
	CategorizeClasses	=0 Objects are not categorized in the Object Create dialog box
		=1 Objects are categorized in the Object Create dialog box (default)
	VBuffer	=0 off-screen bitmap buffer inactive
		=1 off-screen bitmap buffer active (default)
	VBufferSize	=(*width*height*) of off-screen bitmap buffer (default is screen size, which sets the video buffer to buffer the entire screen)
	SmoothMoves	=0 SmoothMoves off (default)
		=1 SmoothMoves on

Table D-1. Lookout.ini File Sections, Keys, and Settings (Continued)

Section	Key	Setting
`[System]` (continued)	`AskBeforeShutdown`	=0 Skip **OK to abort current process?** message window on process exit =1 View **OK to abort current process?** message window on process exit (default)
	`SuppressRedundant Writes`	This setting pertains to remoted Pots. It essentially halts a write (a sending out of a datagram from the Pot) if the last value that the Pot received is the same value it was about to write out. =0 Do not suppress redundant writes from Pots (default) =1 Suppress redundant writes from Pots
	`MaxWavesQueued`	This integer value specifies how many wave files the PlayWave object can queue up. (default = 10)
	`Startup`	Name of the startup process.
	`SaveSource`	You can tell Lookout to save or not save the `.lks` (source) file. =0 Do not save `.lks` file =1 Save `.lks` file (default)

	`AlwaysMaxed`
	
	`NoTaskSwitch`
	
	`NoMenuBar`
	
	`NoCaptionBar`
	
	`LimitPopups`
	
	`AckAlarms`
		Security levels that limit certain abilities. They correspond to what you see in the System Options dialog box. (default = 0)

Table D-1. Lookout.ini File Sections, Keys, and Settings (Continued)

Section	Key	Setting
[System] (continued)	PopupLimit	How many popups can be visible at one time. The security level associated with this feature is LimitPopups. (default = 4)
	NoExit	Security level associated with exiting Lookout. (default = 8)
	LVirtualKeyboard	Determines whether virtual keyboard appears on a left and/or right mouse click. (default = 0)
	RVirtualKeyboard	Determines whether virtual keyboard appears on a left and/or right mouse click. (default = 0)
[Alarms]	Header	=0 Do not show column headers in Alarms Window =1 Show column headers in Alarms Window (default)
	Group	=0 Do not show group info in Alarms Window =1 Show group info in Alarms Window (default)
	ShowPriority	=0 Do not show priority in Alarms Window =1 Show priority in Alarms Window (default)
	ShowTag	=0 Do not show object name in Alarms Window =1 Show object name in Alarms Window (default)
	MostRecent	=0 Display all instances of multiple alarms (default) =1 Display only one instance of multiple alarms

Table D-1. Lookout.ini File Sections, Keys, and Settings (Continued)

Section	Key	Setting
[Alarms] (continued)	AllGroups	=0 Show alarms for specified groups only (default)
		=1 Show alarms for all groups
	Audible	=0 Give no audible notification of alarms
		=1 Give audible notification (beep) of alarms (default)
	Priority	Filter out alarms displayed in the Alarms Window that are below this priority. (default = 1)
	OldAlarmsLimit	How many old occurrences of the same alarm to display in the Alarms Window. (default = 0)
	Style	Display style for the Alarms Window. (default = 1622)
		=1620 top
		=1621 bottom
		=1622 floating
		=1623 minimized
	Lines	Number of lines that can be displayed in the Alarms Window. (default = 3)
	fHeight	Font height of font in the Alarms Window. (no default)
	fWeight	Font weight of font in the Alarms Window. (no default)
	fItalic (no default)	=0 text in Alarms Window NOT italics
		=1 text in Alarms Window italics

Table D-1. Lookout.ini File Sections, Keys, and Settings (Continued)

Section	Key	Setting
[Alarms] (continued)	LogDevice	Which device to log alarms to. (default = "(none)") "(none)" "LPT1:" "LPT2:" "LPT3:" "LPT4:" "COM1:" "COM2:" "COM3:" "COM4:"
	fFace	Font name of font in the Alarms Window. (no default)
[Citadel]	DatabasePath	Path to Citadel.
	DatabaseMachine	Name of the computer you set for Citadel.
[COM1, COM2, COM3, …]	LineType	COM port configuration. =0 wired (default) =1 dial-up =2 RTS-CTS
	CTSTimeOut	How long (in ms) Lookout waits after asserting RTS for a CTS before sending the frame. (default = 100) (minimum = 0) (maximum = 1000)
	RTSDelayOff	How long (in ms) after Lookout has finished sending the frame Lookout continues to assert RTS. (minimum = 2) (maximum = 2000)

Table D-1. Lookout.ini File Sections, Keys, and Settings (Continued)

Section	Key	Setting
[COM1, COM2, COM3, …] (continued)	ReceiveGap	How many empty bytes Lookout must receive before recognizing the end of a frame. (default = 20) (minimum = 0) (maximum = 1000) Lookout 3.8 and later default = 20 Lookout 3.7 and earlier default = 5
	DTR_HangUp	Determines how Lookout hangs up a modem. =0 Use +++ATEH to hang up modem =1 Use DTR to hang up modem (default)
	DialSecs	Length of time Lookout waits to receive a connect signal back from the modem it is calling. The time period begins when Lookout first sends the local modem the dialing prefix command. (default = 60) (minimum = 20) (maximum = 1200)
	Retries	The number of times Lookout dials the specified phone number and attempts to connect to the modem at the other end of the line. (default = 3) (minimum = 1) (maximum = 10)

Table D-1. Lookout.ini File Sections, Keys, and Settings (Continued)

Section	Key	Setting
[COM1, COM2, COM3, …] (continued)	AlarmPriority	Alarms associated with this COM port will have this alarm priority. (default = 1) (minimum = 1) (maximum = 10)
	HangUpSecs	Length of time Lookout waits after hanging up before it sends the local modem the next dialing prefix signal. (default = 2) (minimum = 1) (maximum = 10)
	DiagnosticFile	Full path to a location where Lookout creates a log file recording all serial transactions on this COM port. If a file already exists at this location, it will be appended to. This file is ASCII text, and is primarily used to diagnose serial port communications problems, including modem responses. (no default) After editing the lookout.ini file, reload your Lookout process file to force Lookout to reread the .ini file.
	DiagnosticAllHex	When set to 1, all contents of the diagnostic file are presented as the hexadecimal value for each byte. If set to 0 or not set, Lookout writes ASCII characters, if possible. (default = 0)
	DiagnosticTimeStamp	When set to 1, enables timestamps for diagnostic file. (default = 0)

Table D-1. Lookout.ini File Sections, Keys, and Settings (Continued)

Section	Key	Setting
`[COM1, COM2, COM3, … ]` (continued)	`DiagnosticTimeStampDateFormat`	Sets diagnostic file timestamp format. (default = 66) 66 = mm/dd/yyyy hh:mm:ss 73 = hh:mm:ss.s
	`DialPrefix`	Lookout sends these Hayes AT commands before every use of the modem. (default = "ATX4MVEDT")
`[Database]`	`AlarmPriority`	Priority of database alarms. (default = 9) (minimum = 1) (maximum = 10)
`[DDE]`	`FullSendSeconds`	How often (in seconds) to update all DDE tables. (minimum = 30)
	`DefService`	Default service and topic parameters for all DDE connections. (no defaults)
	`DefTopic`	Default service and topic parameters for all DDE connections. (no defaults)
`[Events]`	`fWeight`	Weight of the font used when printing events. (no default)
	`fItalic`	Italics of the font used when printing events. (no default)
	`fHeight`	Height of the font used when printing events. (no default)
	`fFace`	Name of the font used when printing events. (no default)
	`Units`	Height (in points) of the font used when printing events. (default = 10)

Table D-1. Lookout.ini File Sections, Keys, and Settings (Continued)

Section	Key	Setting
[Alarm Reports]	fWeight	Weight of the font used when printing alarms. (no default)
	fItalic	Italics of the font used when printing alarms. (no default)
	fHeight	Height of the font used when printing alarms. (no default)
	fFace	Name of the font used when printing alarms. (no default)
	Units	Height (in points) of the font used when printing alarms. (default = 10)
[Edit]	Position	Where the next created panel element appears relative to the currently selected panel element. =309 right (default) =310 left =311 top =312 bottom =313 center
	Pixels	How many pixels away in the direction selected by Position to create the next panel element. (default = 1) (minimum = 0) (maximum = 200)
	SnapToGrid	=0 off (default) =1 on
	ShowGrid	This only means showing the dots, not doing the actual snapping. =0 off =1 on (default)

Table D-1. Lookout.ini File Sections, Keys, and Settings (Continued)

Section	Key	Setting
[Edit] (continued)	GridSize	How far apart (in pixels) are the snap to grid hotspots. (default = 20) (minimum = 2) (maximum = 200)
	GridColor	Color of the snap to grid dots. =5570569 invert the panel color (default) =66 black =16711778 white
[Run commands]	Show00	This setting affects the run command identified by the two digits at the end. This controls how the command is run. =0 no command defined (default) =1 iconic =2 normal =3 maximized
	Desc00	This setting affects the run command identified by the two digits at the end. It defines the description of the command that will be displayed in the Run menu.
	Cmnd00	This setting affects the run command identified by the two digits at the end. This is the DOS-style command itself.
	Levl00	This setting affects the run command identified by the two digits at the end. This is the security level required to run the command. (default = 10)

Table D-1. Lookout.ini File Sections, Keys, and Settings (Continued)

Section	Key	Setting
`[Fieldbus]`	`AlarmPriority`	Alarm priority for Fieldbus objects. (default = 0)
`[Pots]`	`SnapDelay`	When a remoted Pot's value is changed from a control panel, Lookout allows this amount of time (measured in seconds) to go by and if the remote value does not change to coincide with the newly selected value, the remoted Pot snaps back to the old value. (default = 10) (maximum = 65)
`[tagname of a pot]`	`SnapDelay`	Same value as described in the `[Pots] SnapDelay`, except you can manipulate this on a Pot by Pot basis. (default = [Pots]SnapDelay setting)
`[Recipe]`	`tagname of a recipe.alarmpriority`	Alarm priority used by a Recipe object. (default = 8)
`[S5-AS511]`	`Diagnostics`	Turns on diagnostic recording functions in S5-AS511 driver. When activated, this option logs diagnostic information to the `s5as511.dai` file in the Lookout directory. =0 diagnostics are off (default) =1 diagnostics are on

Table D-1. Lookout.ini File Sections, Keys, and Settings (Continued)

Section	Key	Setting
[Sixnet]	IOMAPAlarmPriority	Priority of the **Unable to load Sixnet IOMAP library: iobase32.dll** alarm in Sixnet. (default = 5)
	ProjectAlarmPriority	Priority of the **No Sixnet configuration currently loaded** alarm in Sixnet. (default = 5)
[LocalTable]	Delay	Periodically post a DDE table with this frequency (measured in ms). (default = 500)
	EchoCursor	=0 only echo a cursor value if the cell the cursor is pointing to does not have a connection to it =1 no echoes (default)
[tagname of a table]	Delay	Same value as described above, except you can manipulate this on a table-by-table basis. (default = [LocalTable]Delay setting)
	EchoCursor	=0 only echo a cursor value if the cell the cursor is pointing to does not have a connection to it =1 no echoes (default)
[Tiway]	UpdateOutputs	=0 never update writes (purely event driven) (default) =1 update all writes every 100 polls

Table D-1. Lookout.ini File Sections, Keys, and Settings (Continued)

Section	Key	Setting
[tagname of a Tiway]	UpdateOutputs	Same value as described above, except you can manipulate this on an object-by-object basis. (default = [Tiway]UpdateOutputs setting)
[Graphics]	Directory	Full path to the Lookout graphics directory.
[DataPath]	Default	Full path to the Lookout data directory.
	tagname of a spreadsheet	Full path name for an individual spreadsheet.
[Registration]	SerialNumber	Serial number of your Lookout software.
	Key	Key code issued by National Instruments.
	Organization Company	These two entries are for the same piece of information, the organization for which the key was issued.
	NIInternalCode	This is how many days Lookout has been running. It is only updated in versions of Lookout where this is important, like the integrator package.
	Name	Name for which the key was issued.
	HardwareKey	Hardware key information.
[Defaults]	Login	Default login name.

Table D-1. Lookout.ini File Sections, Keys, and Settings (Continued)

Section	Key	Setting
[network]	SymlinkCacheSeconds	Default value is 600, meaning 600 seconds or 10 minutes. This sets the time that Lookout will keep track of values from a given symbolic link focus after that symbolic link has changed focus. Refer to the Symbolic Link documentation for detailed information and the effect of this setting.
[security]	DisableWindows	Set to 0 to fix a problem with some OPC servers that can develop when you activate **User cannot switch to another program** in the Lookout **System options**. Refer to the *OPC Server Security Considerations* topic in the *Lookout Help* for more information about using this setting.

E

Networking with DDE

This appendix explains how to use DDE networking with Lookout.

While you can do most process networking with the faster and more efficient TCP/IP, you might want to use DDE to communicate with other applications, such as Microsoft Excel (as described in Chapter 11, *DDE*).

You can use Lookout DDE networking capabilities to do the following:

- View the same or different screens simultaneously on separate nodes
- Make setpoint adjustments from any node
- Acknowledge alarms from any node
- Configure specific nodes for monitoring only
- Configure a peer-to-peer architecture
- Configure a client-server architecture
- Configure network nodes to communicate through standard telephone lines
- Configure network nodes to communicate through radios

Networking multiple Lookout nodes is a powerful and advanced capability. This appendix refers to many concepts and terms explained elsewhere in the manual. Before you attempt to work with networking, make sure you have a strong understanding of the Lookout architecture and are comfortable working within Lookout on a single node.

To network using Lookout, you must meet the following requirements:

- Purchase and install on each computer a separate, licensed copy of Lookout.
- Equip each computer on the network with compatible network hardware.
- Install Microsoft Windows on each computer.

Because Lookout is a native Windows application, it supports all the networks that Microsoft Windows supports for DDE networking, including Microsoft Windows Network, Microsoft LAN Server, Novell NetWare, Banyan VINES, 3Com 3+Open, DEC Pathworks, IBM LANServer, and so on.

NetDDE Networking Considerations

Networking with Lookout involves passing data back and forth between two or more Lookout nodes. In Lookout, data is passed between and contained within objects. There are two basic methods for passing data between multiple Lookout nodes: *Multilink Networking*, which requires a NetDDE (Network Dynamic Data Exchange) link for every value being passed between the nodes, and *Table Networking*, which implements a *data concentrator* at each node where only tables (which concentrate the data) are linked between nodes. Although both methods involve connecting objects over a network using NetDDE, the implementation and total effect can be quite different.

If you plan to use one of the NetDDE methods, you have to configure automatic NetDDE activation and possibly trusted DDE shares. Refer to the *Running NETDDE.EXE Automatically* section and *Adding a Trusted DDE Share* section in this appendix for setup instructions.

There is one more method for implementing network-like capabilities between multiple Lookout nodes. Unlike the first two methods, however, this technique does not rely on data being passed between Lookout nodes. Instead, you use PLCs, RTUs, or I/O to share data between Lookout nodes. This method is called *hardware networking*.

Multilink NetDDE Networking

Multilink networking consists of passing data from an object on one computer to another object on a different computer. When linked, any change to the value of one object is instantly propagated to and reflected by the second object on the other computer. The way in which data is passed between two objects across a network depends on whether they are controllable objects.

Linking Controllable Objects

User-controlled objects have a DDE option. These objects include Pot objects, Switch objects, and Pushbutton objects.

Controllable objects each have three **Position source** options: **Local**, **Remote**, and **DDE**. When you choose DDE, you instruct the object to take its value from a remote source. This could be a cell in a spreadsheet, another DDE-aware application, or a second copy of Lookout running on the network.

Linking Controllable Objects Together Across a Network

You need at least two computers connected by a functioning network to use the following example. This example refers to the two computers as Station1 and Station2.

1. At Station1, create a Switch object called `Switch1`. Leave **Position source** at the default setting of **Local**.

2. At Station2, create a corresponding Switch object called `Switch1`. However, change the **Position source** to **DDE**. (Although you are not required to give objects matching names over a network, it makes your documentation easier to follow and your applications easier to maintain.)
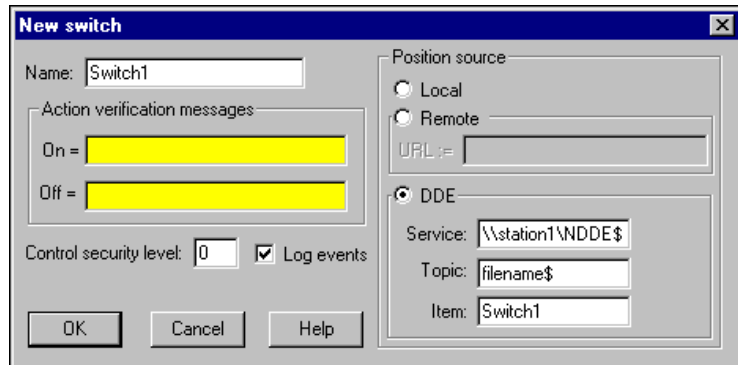
3. In **Service**, enter `\\Station1\NDDE$`, where `Station1` is the network name of the other computer. The backslashes, `NDDE`, and dollar sign are required by Microsoft Windows.

4. In **Topic**, enter `Filename$`, where `Filename` is the name of the Lookout process file on the other computer.

    Again, the dollar sign is required but the `.l4p` file extension is not.

5. In **Item**, enter the name of the object at the other computer that you want to link to (in this case, the other object is also called `Switch1`).

6. Click **OK**.

Test your network by flipping the switch. When you toggle the switch at either station, it instantly flips at the other workstation. You can add to, modify, delete, or revise your logic on either computer, completely on-line, and still maintain the network link between the two objects.

Linking Non-Controllable Objects

If a Lookout object is not a Pot, Switch, or Pushbutton, it probably does not have built-in DDE capabilities. However, all you are really trying to do is pass data between two computers. After the data is in Lookout, any object can access it, so you can create a DdeLink object to receive data from another application using DDE or NetDDE. Refer to the *DdeLink* topic in *Lookout Help* for more information about using DDE. In this example, the other application is Lookout running on a second computer.

To Access Real-Time Data at Another Computer

You need at least two computers connected by a functioning network to follow this example. This example refers to the two computers as Station1 and Station2.

1. At Station1, create an Average object. It can average whatever values you want, but give it the name `Average1`.

2. At Station2, create a DdeLink object and name it `Average1`. Although you are not required to give objects matching names over a network, it makes your applications easier to maintain.
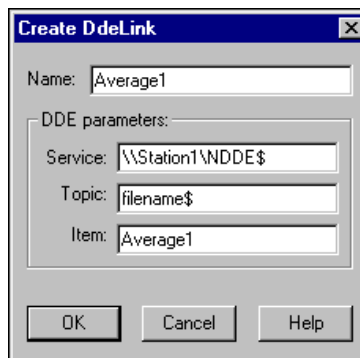
3. In **Service**, enter `\\Station1\NDDE$`, where `Station1` is the network name of the other computer. The backslashes, `NDDE`, and dollar sign are required by Microsoft Windows.

4. In **Topic**, enter `Filename$`, where `Filename` is the name of the Lookout process file on the other computer.

   Again, the dollar sign is required but the `.14p` file extension is not.

5. Enter the name of the object at the other computer that you want to link to in the **Item** field (in this case, it is also called `Average1`).

6. Click **OK**.

7. Finish the definition of the DdeLink object by inserting the DDE expression into your panel.

After you establish the link, you can connect other objects at Station2 to the `Average1` DdeLink object. Remember, just because you named the DdeLink object `Average1` does not make it a Lookout Average object. It just receives data from the Average object at Station1.

Table Networking

Table networking consists of consolidating all the data that needs to be shared over the network in a single table or array of values. You might think of these tables as "data concentrators"—although they can do much more than just serve as a repository for data. Refer to the discussion of the *DataTable* object in the *Lookout Help* for more information about the DataTable object. You can link a table through NetDDE to a corresponding table on another computer. When linked, the tables update each other on all changes within their databases.

Note   Table networking is one of the most involved and advanced concepts in Lookout. You should have a complete understanding of objects, database connections, events, and general Lookout architecture before you attempt table networking.
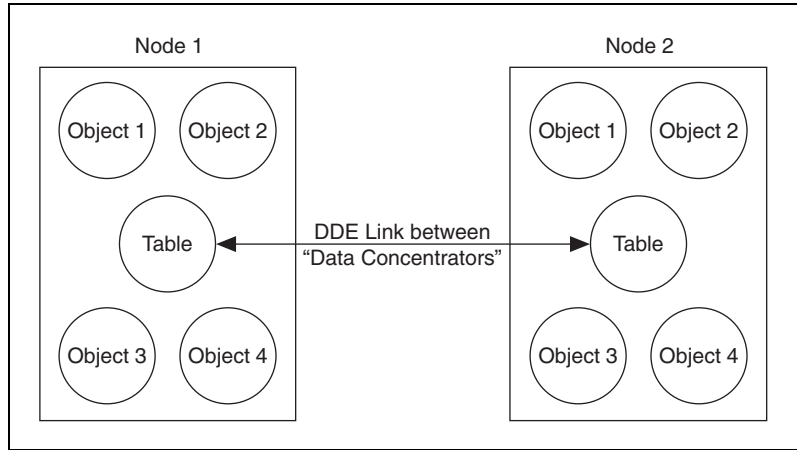
Figure E-1. Table Networking

You might have better results linking multiple DataTables for networking purposes if you plan before jumping headfirst into your application. The following example shows how table networking is done.

This example consists of three computers, each running a copy of Lookout. Set up three nodes on the network with the names Station 1, Station 2, and Station3. Station1 is the only node directly connected to a PLC, but you want to monitor and control the I/O from any of the three nodes. The following diagram depicts the network structure of this example.
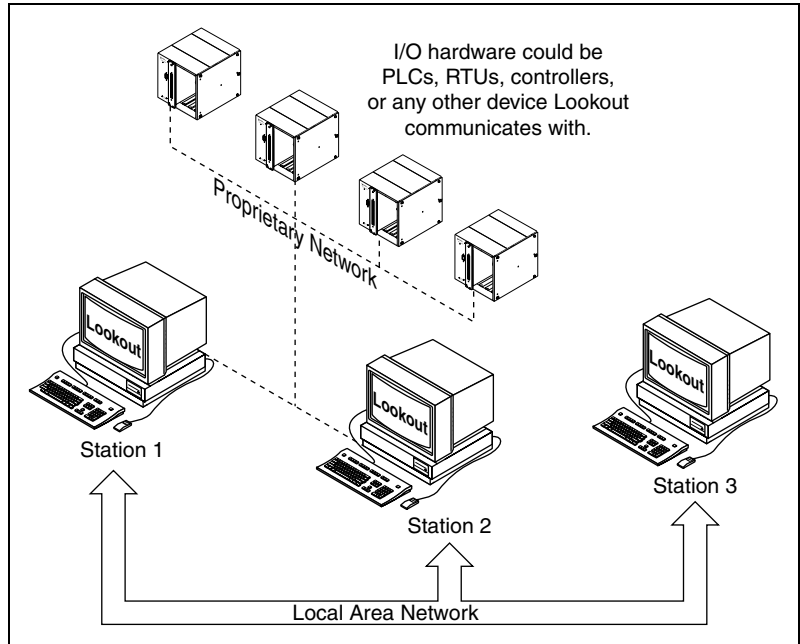
Figure E-2. Network Structure of the Example

By doing a little planning, you can configure Lookout at Station 1 so that you can reuse its process file at Station 2 and Station 3. The following diagram depicts the general design of the Lookout process file at Station 1. Notice that driver objects are connected to the DataTable object using dotted lines while all other objects are connected using solid lines. This is for illustrative purposes only.
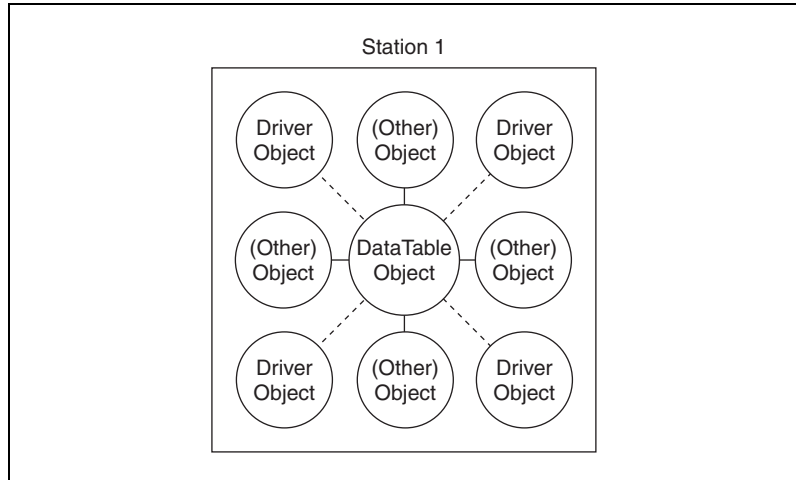
Figure E-3. Lookout Process Design at Station 1

The focus of this picture is the driver objects and their connections to the DataTable. Eventually, you will modify this process file to run at Station 2 and Station 3 by deleting all connections to driver objects, then deleting the driver objects themselves at Stations 2 and 3.

It is important to route all signals to and from driver objects through the DataTable. If you want to display an analog input from a PLC on a control panel, do not connect the PLC signal directly to the control panel. Instead, first connect it to the DataTable and then connect the appropriate DataTable cell to the control panel. If you need to connect an operator setpoint to an output on a PLC, do not connect the setpoint directly to the PLC. First connect the setpoint to the DataTable, then connect the appropriate DataTable cell to the output on your PLC. This is only necessary for signals that originate from, or are destined for, a driver object. Although this procedure adds another step to your Lookout configuration for Station 1, it greatly simplifies the development of the process files for Station 2 and Station 3.

Now that you have developed an application at Station 1, move over to Station 2. Copy the process file to the Station 2 computer and delete all the connections to and from your driver objects. Then, delete the driver objects themselves.

Remember, you do not want Station 2 attempting to communicate with the PLCs because there is no physical connection to that proprietary network. Station 2 should get its information from Station 1.

The only thing left to do is modify the DataTable object at Station 2. Change its **Table location** parameter to **DDE**. In **Service**, enter `\\Station1\NDDE$`. In **Topic**, enter `Filename$`, where `Filename` is the name of the process file at Station 1. In **Item**, enter the name of the DataTable object at Station 1. Your DataTables are now bidirectionally connected through a NetDDE link.

At this stage, you can add more operator stations to the network. Copy the process file from Station 2 to Station 3, or to any number of other Lookout nodes on the network. No changes to the files themselves are necessary. Your Lookout network topography should now look something like this.
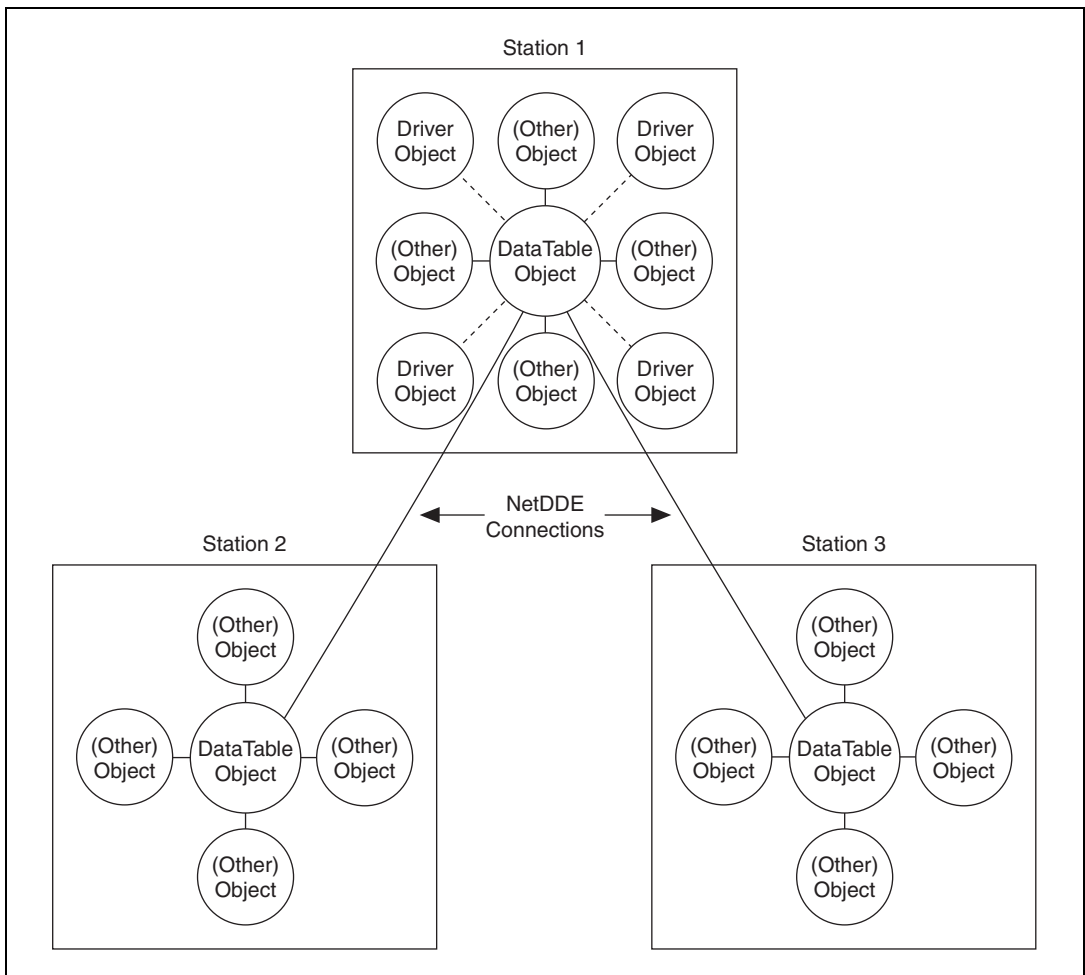
Figure E-4. Network Topology of Example

Hardware Networking

In general, you should try to avoid hardware networking as described here with Lookout 4 or greater. To implement the hardware networking method, all Lookout nodes you want networked must have direct communication access to all your hardware as displayed in the following diagram. Because of this, the hardware networking method is only applicable in certain circumstances.
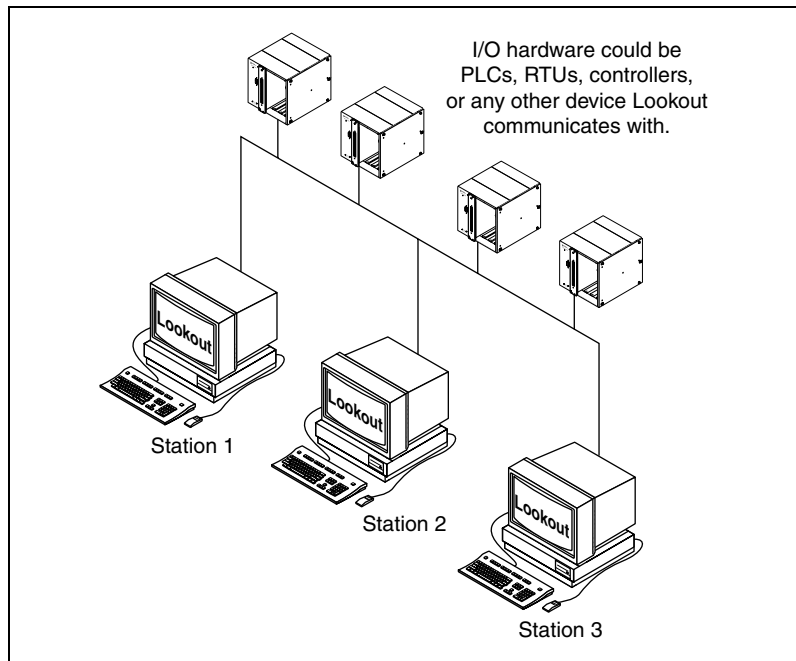
Figure E-5. Hardware Networking Example

This is the most straightforward of the three networking methods to implement. Configure a process file at one node and copy it to all other nodes that have direct communication access to your hardware. The only thing to remember when building your process file is to configure all Pots, Switches, and Pushbuttons with the **Remote** parameter setting. This ensures that they stay in sync with the point they are writing to. In other words, if a switch at one Lookout node changes the status of an output, all the respective switches at the other Lookout nodes will reflect the change and automatically flip to stay in sync. Refer to the *Pot*, *Switch*, and *Pushbutton* topics in the *Lookout Help* for more information about the **Remote** parameter.

Your hardware, network, and communications topology are the determining factors on whether you can use hardware networking. The hardware networking method is seldom available because it is rare that all your Lookout nodes can communicate directly with all your hardware.

Multilink and Table Networking Comparison

There are advantages and disadvantages to both multilink and table networking. Multilink is quick and easy for small jobs. However, every NetDDE link requires processor time, so the more links you have, the more CPU overhead is used. Remember, each value passed between computers requires a separate DdeLink object. Your computer can quickly become bogged down if hundreds of links are constantly sending values back and forth.

On the other hand, while the table method might require more initial planning and development time, there will be only one NetDDE link between computers connecting the DataTables. This method dramatically reduces the amount of DDE traffic on the network. That is why table networking is the method of choice for large systems where computers are sharing hundreds or thousands of values across the network.

So how many points are too many? There is no single answer, but Lookout has been used with the table networking method on systems having over 6,000 values, and the network update time is well under one second with only three percent CPU usage.

Setting up Networking with DDE

Lookout uses Network DDE (Dynamic Data Exchange) for client/server networking.

Two components of Microsoft Windows are required when using Network DDE: the Network DDE agent (`netdde.exe`) and the NetDDE Share Database Manager (DSDM). Exactly how these components are used depends on the operating system on which Lookout runs: Windows Me/98 or Windows 2000/NT/XP.

The program `netdde.exe` is an agent that runs in the background and transfers DDE messages between computers. This program should be running before an instance of Lookout is started if you intend for that session to act as a server. Lookout does not automatically run `netdde.exe` itself.

Running NETDDE.EXE Automatically

The following sections show how to make sure that `netdde.exe` runs every time you start Windows.

Windows Me/98

1. In Windows Me/98, Click **Start»Settings»Taskbar**.

2. In the **Taskbar Properties** dialog box, choose the **Start Menu Programs** tab.

3. Click the **Add** button.

4. In the **Command line** field of the **Create Shortcut** dialog box, enter `netdde.exe`, then click **Next**.

5. Choose the **Start Menu»Programs»StartUp** folder, then click **Next**.

6. Select a name (such as `NetDDE`) for the shortcut; select **Finish** and **OK**.

Windows 2000/NT/XP

1. In the Program Manager, double-click the **Control Panel** icon, or run `control.exe` using the Program Manager **File»Run** dialog box. If you are using NT version 4.0, click **Start**, point to **Settings**, and choose the **Control Panel**.

2. In the **Control Panel** window, double-click the **Services** icon.

3. From the list of services, select `Network DDE`.

4. Click the **Startup** button.

5. In the **Startup Type** box, select `Automatic`.

6. Select **OK** and **Close** to exit the **Control Panel**.

Adding a Trusted DDE Share

To allow someone else to connect to a DDE share (for example, `process$`) on your computer when you are logged in, you must *trust* the DDE share. When other users connect to the share remotely, the application they connect to is running in your security context because you are the logged-on user. You must give permission for the other users to access the share. Even another person who is an administrator cannot trust a share for your account.

In Windows Me/98, trusted shares are automatically created by Lookout.

In Windows NT, you must use the program `ddeshare.exe` to add a trusted share. Exactly how you set this up depends on the level of security you require. The following sequence of steps describes how to set up a share that gives everyone on the network full DDE access to Lookout.

1. From the Program Manager, select **File»Run**. If you are using Windows 2000/NT/XP, click **Start** and select **Run**.

2. Enter `ddeshare.exe`.

3. Select **Shares»DDE Shares**.

4. In the **DDE Shares** dialog box, select **Add a Share**.

5. In the **DDE Share Properties** dialog box, do the following:

   a. For the **Share Name**, enter the name of the Lookout process file that you want to share, omitting the `.l4p` extension, and adding a '`$`' at the end. For example, if your process file was `process.l4p`, you would enter

      `PROCESS$`

      on this line.

   b. For the **Static Application Name**, enter `Lookout`.

   c. For the **Static Topic Name**, enter the process file without extension, such as `process`.

   d. Leave the **Allow start application** and **is service** boxes unchecked.

   e. In the **Item Security** box, select **Grant access to all items**.

   f. Click the **Permissions** button.

   g. In the **DDE Share Name Permissions** dialog box, do the following:

      • In the **Names** list, select `Everyone`. If there is no entry for `Everyone`, use the **Add** button and associated dialog box to create it.

      • In the **Type of Access** box, select `Full Control`.

      • Select **OK**.

   h. Select **OK**.

6. You should now be back in the DDE Shares dialog box. Select the share you just created and click the **Trust Share** button.

7. In the **Trusted Share Properties** dialog box, select the **Initiate to Application Enable** button, then click **OK**.

8. In the **DDE Shares** dialog box, click **OK**.

9. Exit the DDE Shares program.

Note   For the DDE share to be available, it must be trusted by the user currently logged in to the computer (using steps 6 through 9). Alternatively, the share can be provided as a service by selecting the **Is service** box in the **DDE Share Properties** dialog box. In this case, the share is always available to anyone listed in the DDE Share Name Permissions dialog box.

Note   When a Lookout client tries to connect to a DDE server on a computer running Windows NT, it might query the user for a user name and password that are valid on the server computer. In this case, the user password should not be NULL, or the attempt to connect to the server will fail.

F

Technical Support and Professional Services

Visit the following sections of the National Instruments Web site at `ni.com` for technical support and professional services:

- **Support**—Online technical support resources include the following:

  - **Self-Help Resources**—For immediate answers and solutions, visit our extensive library of technical support resources available in English, Japanese, and Spanish at `ni.com/support`. These resources are available for most products at no cost to registered users and include software drivers and updates, a KnowledgeBase, product manuals, step-by-step troubleshooting wizards, conformity documentation, example code, tutorials and application notes, instrument drivers, discussion forums, a measurement glossary, and so on.

  - **Assisted Support Options**—Contact NI engineers and other measurement and automation professionals by visiting `ni.com/support`. Our online system helps you define your question and connects you to the experts by phone, discussion forum, or email.

- **Training**—Visit `ni.com/custed` for self-paced tutorials, videos, and interactive CDs. You also can register for instructor-led, hands-on courses at locations around the world.

- **System Integration**—If you have time constraints, limited in-house technical resources, or other project challenges, NI Alliance Program members can help. To learn more, call your local NI office or visit `ni.com/alliance`.

If you searched `ni.com` and could not find the answers you need, contact your local office or NI corporate headquarters. Phone numbers for our worldwide offices are listed at the front of this manual. You also can visit the Worldwide Offices section of `ni.com/niglobal` to access the branch office Web sites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

Glossary

A

ACK	Acknowledge (an alarm or event).
ActiveX	An ActiveX control is a type of software object that can be used in developing applications or run inside of other applications. ActiveX controls, the third version of OLE controls, use COM technologies to provide interoperability with other types of COM components and services.
Aggregate	A collection of Lookout objects, connected and configured, that is saved as a process that can be used within other processes.
alarm	Software notification of an abnormal condition in a process.
alias	Name given to a data member. A data member can have more than one alias, each with different associated configurations.
ASCII	American Standard Code for Information Interchange.

B

baud rate	Measurement of data transmission speed, formally defined as the number of electronic state changes per second. *See* bps.
.bmp files	Graphic files in bitmap format. *See* .wmf files.
bps	Bits per second—measure of the rate of transfer of data.

C

CBL compiler	Lookout uses the CBL (Control Block Language) compiler to compile a Lookout source file (`.lks`) into a binary file (`.l4p`).
.cbx file	A Lookout file containing a Lookout object. A `.cbx` (Control Block Extension) file can have one or more objects in it.
Citadel	The Lookout historical database that stores your data for access later.

client	A Lookout process that monitors a Lookout server process. Lookout clients should be computer independent so that they can be run from any computer on your network.
COM port	Serial communications port.
.csv files	Comma Separated Value file, a format widely accepted by spreadsheet and other data handling programs.
CTS	Clear to Send. Part of a handshaking protocol for certain devices that connect the serial port of a computer.

D

data member	Data source or sink associated with a Lookout object. A data member can be both readable and writable. It contains information about the current state of the object, such as the value, whether the control is visible or hidden, and so on. Driver object data members represent the I/O values of your control hardware. *See also* alias.
data type	Kind of value (numeric, logical, or text) that a parameter or data member can hold.
database	Collection of data stored for later retrieval, display, or analysis.
datagram	Message sent between objects in Lookout. A datagram contains a route and a value.
DCOM/COM	Distributed Component Object Model, a Microsoft standard in which client program objects request services from server program objects.
	The Component Object Model (COM) is a set of interfaces, clients, and servers used to communicate within the same computer.
DDE	Dynamic Data Exchange, currently used in Lookout to exchange data with other programs (such as Microsoft Excel) running on your network.
deadband	An amount of change that must be exceeded for an alarm to sound or a change in state to be recorded. A deadband prevents small oscillations in a value from triggering an alarm and then canceling it too rapidly.
deviation	An amount of change that must be exceeded before a value is logged to a database. A deviation setting can filter out insignificant changes in a value when logging data.

dialing prefix	Part of the Hayes AT command set for use with modems.
DLL	Dynamic Link Library, which is a collection of small, special-purpose programs which can be called by a larger program running on the computer.
DNS	Domain Name System. A static, hierarchical name service for TCP/IP hosts.
driver objects	Lookout objects used to communicate with PLCs, RTUs, and other I/O devices.
DTR	Data Terminal Ready.

E

edit mode	Lookout mode in which you can alter and create objects within a process. Switch in and out of edit mode by pressing <Ctrl-Space> or by selecting **Edit»Edit Mode**.
engineering unit	In Lookout, used to refer to scaled or converted data. *See also* raw unit.
Ethernet	A widely used, standardized local area networking technology, specified in the IEEE 802.3 standard.
event	An event can be anything that happens within the Lookout environment. In Lookout, events include such things as adjusting a control value, entering or leaving edit mode, opening or closing a control panel, and logging in or logging out of the system.
expression functions	Mathematical, logical, and other functions used by Lookout expressions.
expressions	Lookout expressions can function like variables, capable of performing flexible, real-time math operations, condition testing, and other complex operations functions. In Lookout, expressions can be independent (expression) objects, strings entered into yellow object parameter fields, connected to object data members, or inserted on control panels.

F

frame	In a communication, sequence of bytes sent from a computer to a device or vice versa.

H

Hi and HiHi   Alarm settings. Both warn that a value has gone above some setpoint. Generally a Hi alarm is used to alert an operator of a need for intervention. A HiHi alarm is usually used to alert an operator that the value has been exceeded by an even greater margin than a Hi alarm indicates, and is usually used to indicate an urgent need for action.

historical logging   The process of storing data in a database for use at another time, or from another location.

HMI   Human-Machine Interface.

HOA   Hand-Off-Auto control, used to set whether a value must be changed manually, is completely turned off, or functions automatically.

I

I/O   Input/Output.

I/O point   Every read-only, write-only, or read-write connection Lookout makes to external hardware is counted as an I/O point. Lookout is licensed for use with a set number of I/O points. If you exceed the number you are licensed to use with your copy of Lookout, a warning message appears on your computer screen warning you to shut down one of your processes within a specified time before Lookout cuts back on I/O usage.

(implicit) data member   The (implicit) member represents what Lookout considers to be the most commonly used data member of that object type.

IP   Internet Protocol.

L

.l4p files   File extension for Lookout process files. These are the compiled files Lookout runs when it runs a process.

.l4t files   File extension for a Lookout state file, which stores the values for Lookout controls and other objects with state information.

.lka files   File extension for Lookout security files.

.lks files	File extension for a Lookout source file, which Lookout compiles to make a Lookout process file that Lookout can run.
logging	The process of storing data in a computer database file.
logical data member	A Lookout data member that represents a binary or on/off state.
Logos	National Instruments proprietary networking.

M

MAX	National Instruments Measurement and Automation Explorer.
multiplex	A method of working with more than one data stream using only one communications channel. There are a number of different methods of multiplexing, depending on the hardware and software being used. A number of Lookout driver objects support multiplexing hardware.

N

NetDDE	A way of networking using DDE (dynamic data exchange), retained in Lookout 4 and later for compatibility with earlier versions of Lookout.
numeric data member	A Lookout data member that represents a floating point number, an analog value.

O

object	Fundamental building block of a Lookout application. An object is a self-contained software representation of some commonly-used item that includes the item's state, behavior, and identity. For example, an object can represent a specific switch, timer, or PLC. It can also be a more abstract tool used in control strategies or networking.
object connections	Software links between objects used to transmit data and commands from one object to another.

ODBC	Open DataBase Connectivity, a standard application programming interface (API) for accessing a database. You can use ODBC statements to access files in a number of different databases, including Access, dBase, DB2, and Excel.
	ODBC is compatible with the Structured Query Language (SQL) Call-Level Interface. ODBC handles SQL requests by converting them into requests an ODBC database can use.
OPC	OLE for Process Control, an industry standard interface providing interoperability between disparate field devices, automation/control systems, and business systems. Based on ActiveX, OLE, Component Object Model (COM), and Distributed COM (DCOM) technologies.

P

parameter	Defines a Lookout object's characteristics and functionality limits. Parameters appear in the dialog box when you create an object or modify its properties.
ping	A utility program in Windows and DOS that checks to see if a computer can be reached across a network. Also used to indicate the running of that program.
PLC	Programmable Logic Controller.
poll	A software event in which a computer checks some value in a device or register. In Lookout, a logical command that forces a device poll to check data member values.
poll rate	How often a device is polled.
polymorphism	In Lookout, data polymorphism means that one data type is interpreted appropriately when connected to an input of another type.
pop-up panel	One variety of Lookout control panel that can only be displayed at the size set by the process developer, and which cannot be maximized. When open, a pop-up panel remains on top of other panels until minimized.
process	In Lookout, process refers to a Lookout "program," used for industrial automation, control, monitoring, or reporting.

process file	The Lookout binary file Lookout executes when running a process. Carries the `.l4p` extension.
proxy user	You can specify a proxy user account as a way to handle unrecognized access requests to your data.

R

raw unit	Data as it arrives in your process, such as voltage or amperage. Thermocouple data, for instance, arrives in volts as the raw unit, and must be converted to degrees. *See also* engineering unit.
receive gap	A serial communications setting that determines the number of empty bytes (or amount of time) a driver receives before recognizing the end of a message frame and requesting another message.
redundancy	A system for making sure that a computer can come online and run a Lookout process if the computer currently running that process fails for some reason.
RTS	Request to Send, part of a handshaking protocol for certain devices that connect the serial port of a computer.
RTU	Remote Terminal Unit, a device similar to a PLC for use at a remote location, communicating with a host system through radio or telephonic connections.
run mode	Lookout mode in which processes run but no editing changes can be made. Switch in and out of run mode by pressing <Ctrl-Space> or selecting **Edit»Edit Mode**.

S

server	A process that provides data (services) to client processes. In Lookout, server processes are intended to have direct connections to field hardware. Client processes interact with field hardware through server processes.
source file	Lookout file that can be compiled to produce a binary Lookout process file that runs a process. Uses a `.lks` file extension.
SQL	Structured Query Language, used to get information from and update information in a database.

standby	A computer standing by to take over running a process if the primary computer fails or falls offline.
startup file	A Lookout process file (.l4p) you designate to open and run any time Lookout is opened.
state file	The Lookout file that stores the value of all Lookout control parameters and object data members in use in a process. Uses the file extension .l4t.

T

TCP/IP	Transmission Control Protocol, a method (protocol) for sending data between computers. Used with IP, the Internet Protocol. TCP/IP sends data as packets, with IP handling the delivery of data and TCP keeping track of the individual packets.
text data member	Lookout data member used for text data.
trace	A term for data from a single source over some period of time, stored in an ODBC-compliant database.
traces table	ODBC databases present data in the form of traces tables. A traces table contains a field or column of data for each data member being logged, along with a field you can use to query the database.
trend	Historical data showing the change in a value over time. Often used in connection with graphing the data for display.

U

URL	Uniform Resource Locator. In Lookout, a URL is used in any green dialog box field, to indicate an a fully-specified path to a data member, object, process folder, or process.

W

.wmf files	File extension given to Windows Metafile graphic files.

Index

A

ABS function (table), 7-12

Access (Microsoft), compliance with ODBC (note), B-4

accounts, managing. *See* user accounts

acknowledging and clearing alarms, 9-13
    programmatically, 9-13

ACOS function (table), 7-20

action verification, implementing, 10-16

ActiveX files, deploying, 13-3

ActiveX objects
    overview, 1-17
    using in processes, 5-9

Administrator account, 10-4

aggregate definition file, 13-1

aggregate objects, 5-9

Alarm condition option for numeric data members, 5-14

alarms
    acknowledging and clearing
        preventing users from acknowledging, 10-10
        programmatically, 9-13
    configuring for server process, 3-7
    data quality problems, 9-14
    DDE, 11-5
    defining alarms conditions
        automatic generation by objects, 9-2
        configuring alarm areas, 9-3
        data member values for triggering, 9-1
        setting priorities, 9-4
    printing
        as alarms and events happen, 9-12
        manually, 9-10
    selecting processes for monitoring, 9-4
    showing networked alarms and events, 3-25
    silencing audible alarms, 9-14

    viewing
        filtering alarms, 9-9
        in Alarms Window, 9-6
        setting display options, 9-8

Alarms Window
    display options, setting, 9-8
    filtering alarms, 9-9
    overview, 1-11
    viewing alarms, 9-6

Alias for numeric data members, 5-12

AND function (table), 7-9

Apache Web server, 12-10

application development
    See also deploying application files
    design of applications, 2-1
    design recommendations
        backing up data, 2-4
        choosing operating system, 2-4
        client license considerations, 2-7
        Connection Browser, 2-7
        connections, 2-7
        control strategy, 2-7
        enabling logging, 2-7
        folder hierarchy, 2-6
        Lookout Help, 2-5
        multiplexing panels in complex processes, 2-8
        Object Explorer, 2-7
        power failure, protecting against, 2-4
        redundancy, 2-4
        server/client design implementation, 2-5
        using powerful objects, 2-5
    development process, 2-3
    example application, 2-1

architecture for Lookout. *See* Lookout

archiving historical data files, 8-5

H

I

K

L

X